UNIG2U-BENCH: DO UNIFIED MODELS ADVANCE MULTIMODAL UNDERSTANDING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified multimodal models have recently demonstrated strong generative capabilities, yet whether and when generation improves understanding remains unclear. Existing benchmarks lack a systematic exploration of the specific tasks where generation facilitates understanding. To this end, we introduce UniG2U-Bench, a comprehensive benchmark categorizing generation-to-understanding (G2U) evaluation into 7 categories and 30 subtasks, requiring varying degrees of implicit or explicit visual transformations. Extensive evaluation of over 30 models reveals three core findings: 1) Unified models generally underperform their base Vision-Language Models (VLMs), and Generate-then-Answer (GtA) inference typically degrades performance relative to direct inference. 2) Consistent enhancements emerge in spatial intelligence, visual illusions, or multi-round reasoning subtasks, where enhanced spatial and shape perception, as well as multi-step intermediate image states, prove beneficial. 3) Tasks with similar reasoning structures exhibit correlated behaviors, and unified models built on the same base VLM behave similarly even after controlling for the shared baseline, suggesting that generation-understanding coupling induces class-consistent inductive biases shaped largely by the backbone. These findings highlight the necessity for more diverse training data and novel paradigms to fully unlock the potential of unified multimodal modeling.

1 INTRODUCTION

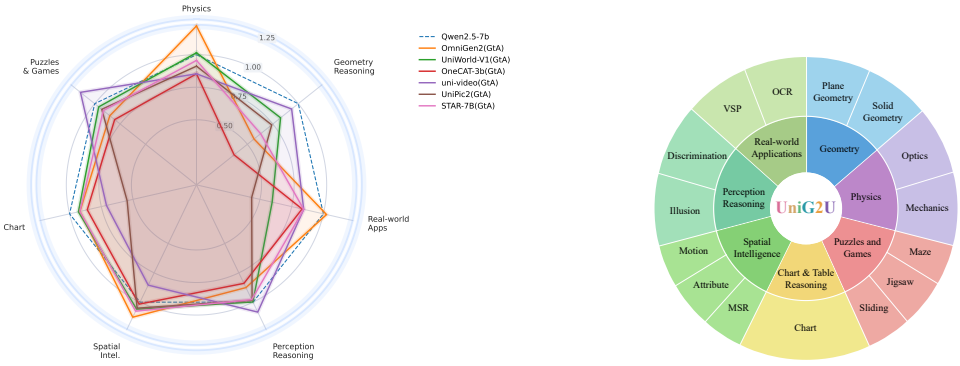
Recent advances in multimodal foundation models have sparked growing interest in unifying understanding and generation within a single architecture Yang et al. (2025b). Unified models have achieved impressive results in image synthesis, editing, and cross-modal generation, suggesting that strong perceptual and semantic understanding is essential for high-quality generation Deng et al. (2025); Chen et al. (2025); Xie et al. (2025a); Zou et al. (2025); Xie et al. (2025b); Li et al. (2025b).

However, existing studies mainly investigate this relationship in one direction: understanding improves generation. Whether generation can in turn enhance understanding remains largely unexplored. In many real-world problems—such as geometric construction, spatial layout reconstruction, or puzzle state tracking—reasoning involves visual transformations that cannot be fully captured through linguistic abstraction. In these scenarios, generation may serve as an externalized reasoning medium that exposes visual structures otherwise inaccessible to text-only reasoning.

Existing benchmarks such as UniMMU Zou et al. (2025) and MME-Unify Xie et al. (2025b) mainly evaluate understanding and generation separately, asking whether models can answer questions or generate images independently. They do not directly test whether generative capability actively improves understanding. Moreover, evaluating unified models is challenging because performance differences may stem from scaling effects or backbone changes rather than the integration of generative reasoning.

To address this gap, we introduce **UniG2U**, a large-scale diagnostic benchmark designed to evaluate the Generation-to-Understanding (G2U) effect. UniG2U contains **3,000 curated instances** spanning **seven reasoning categories and 30 subtasks** (Fig. 1b) and evaluates **over 30 models**. By pairing unified models with exact or closest available base VLMs under matched inference protocols, we provide a controlled diagnostic comparison of performance differences associated with multimodal

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.



(a) Model performance radar.

(b) Task selection and coverage in UniG2U.

Figure 1: **Overview of UniG2U.** (a) Relative performance radar of representative models. (b) Task taxonomy and coverage of the benchmark.

unification. Because paired models may still differ in pretraining data, instruction tuning, and implementation, this comparison does not fully isolate a causal effect of generation. Overall performance trends across task categories are illustrated in Fig. 1a.

Each unified model is evaluated under two inference modes: **Direct** and **Generate-then-Answer (GtA)**. Our experiments reveal three key findings. First, unified models often underperform their paired base VLMs on standard understanding tasks, a gap we call the *alignment tax*. Second, despite this overall decline, improvements consistently emerge in spatial reasoning, visual illusion, and multi-step reasoning tasks, where intermediate visual structures provide useful cues. Third, G2U gains exhibit task- and model-level correlations: tasks with similar reasoning structures move together, and models built on the same base VLM behave similarly even after controlling for the shared baseline.

Our main contributions are summarized as follows:

1. **A Novel Testbed.** We introduce UniG2U, a large-scale diagnostic benchmark for studying generation-to-understanding effects in unified multimodal models, containing 3,000 curated instances across seven reasoning categories and 30 subtasks.
2. **Comprehensive Evaluation.** We conduct a systematic empirical study of unified models, benchmarking over 30 models and estimating G2U gains through matched comparisons with paired base VLMs.
3. **Analysis.** A likelihood analysis of unified training and generate-then-answer inference identifies the quantities that govern when generation can help or harm understanding, and we relate the empirical trends to these quantities.

2 Related Works

Unified Multimodal Benchmarks. Prior benchmarks cover broad multimodal capabilities and mixed-modality outputs (e.g., MME-Unify, UEval, GGBench, and OmniPersona), or study bidirectional visual-language reasoning (e.g., RealUnify, Uni-MMMU, ROVER, and UniGen) Xie et al. (2025b); Li et al. (2026); Wei et al. (2025); Zhong et al. (2026); Shi et al. (2025); Zou et al. (2025); Liang et al. (2025); Tian et al. (2025). However, they do not directly test whether intermediate image generation improves understanding. UniG2U addresses this gap by pairing unified models with their base VLMs and evaluating 3,000 instances across seven reasoning categories and 30 subtasks. Table 1 summarizes the distinction.

Unified Multimodal Models. Unified models combine understanding and generation in one architecture, using discrete visual tokens with autoregressive modeling (Chameleon Team et al., 2024; Wang et al., 2024; Chen et al., 2025), hybrid autoregressive and diffusion objectives (Xie et al., 2024; 2025a; Deng et al., 2025), or understanding backbones coupled with diffusion decoders (Lin et al., 2025; Xiao et al., 2024; Wang et al., 2025b). Their evaluation focuses on generation quality

and on understanding benchmarks inherited from VLMs (Yue et al., 2023; Lu et al., 2023). We ask instead whether the generative capability improves understanding, and relate the answer to the likelihood that unified training optimizes.

Reasoning with Visual Intermediates. Visual Sketchpad (Hu et al., 2024) and visual scratchpads (Hsu et al., 2023) let models draw auxiliary constructions before answering, and agentic systems invoke image tools during reasoning. These works show that intermediate images can help on selected tasks. UniG2U measures this effect systematically across task categories and model families, and our analysis characterizes when committing to a generated intermediate helps or hurts.

Table 1: Comparison of unified multimodal benchmarks.

Benchmark	# Understanding Samples	# Subtasks	Synergy
MME-Unify	1964	3	✗
Uni-MMMU	524	4	✓
ROVER	404	6	✓
RealUnify	400	4	✓
UniG2U (ours)	**3000**	**30**	✓

3 A Likelihood View of Generation-to-Understanding

Setting and G2U gains. Each instance consists of an input $x = (V, T, Q)$ with visual context V, textual context T, and query Q, and an answer $y \in \mathcal{Y}$. A unified model M with parameters θ defines a generation distribution $p_\theta(G \mid x)$ over intermediate visual artifacts G, an answering distribution $p_\theta(y \mid x, G)$, and a direct answering distribution $q_\theta(y \mid x)$. With $\mathrm{Perf}(M) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{A}(M(x), y)]$ for a task metric $\mathcal{A}$, we measure

$$\Delta_{\mathrm{Direct}} = \mathrm{Perf}_{\mathrm{Direct}}(M) - \mathrm{Perf}(\mathcal{B}(M)), \qquad \Delta_{\mathrm{GtA}} = \mathrm{Perf}_{\mathrm{GtA}}(M) - \mathrm{Perf}_{\mathrm{Direct}}(M), \tag{1}$$

where $\mathcal{B}(M)$ is the paired base VLM (Appendix F gives the full formalization).

Unified training and Δ_{Direct}. Understanding alone maximizes $\log p_\theta(y \mid x)$, whereas a unified model is also trained to generate visual content. Since $\log p_\theta(y \mid x) = \log p_\theta(x, y) - \log p_\theta(x)$, the joint objective decomposes as

$$\mathrm{KL}\big(p(x, y) \,\|\, p_\theta(x, y)\big) = \underbrace{\mathrm{KL}\big(p(x) \,\|\, p_\theta(x)\big)}_{\text{generation}} + \underbrace{\mathbb{E}_x\, \mathrm{KL}\big(p(y \mid x) \,\|\, p_\theta(y \mid x)\big)}_{\text{understanding}}. \tag{2}$$

Here $p_\theta(x)$ stands for the generative objective of the unified model, which is an exact likelihood for autoregressive models and a proxy for diffusion and flow-matching losses. How the two terms trade off during training depends on how the losses are weighted and normalized. The generation loss is accumulated over many visual tokens, so its gradient can be large relative to that of a short answer, but its actual influence is determined by the relative size and direction of the two gradients.

Proposition 3.1 (Gradient alignment). *Let g_U and g_G be the gradients of the understanding loss $\mathcal{L}_U$ and the generation loss $\mathcal{L}_G$. For one step $\theta^+ = \theta - \eta(g_U + \lambda g_G)$ of unified training and one step $\theta^U = \theta - \eta g_U$ of understanding-only training, if $\mathcal{L}_U$ is smooth,*

$$\mathcal{L}_U(\theta^+) - \mathcal{L}_U(\theta^U) = -\eta\lambda\, \|g_U\|\, \|g_G\| \cos(g_U, g_G) + O(\eta^2). \tag{3}$$

To first order, a step of unified training therefore improves understanding more than understanding-only training when the two gradients are aligned and less when they conflict, with an effect proportional to $\lambda\|g_G\|$. The proposition does not determine the sign of the alignment for a given task. A natural hypothesis is that alignment is more likely when answers depend on spatial layout, shape, or geometric state, which a generative model must represent, than when they depend on text, numbers, or knowledge. We use this hypothesis to interpret the results but do not measure the gradients directly.

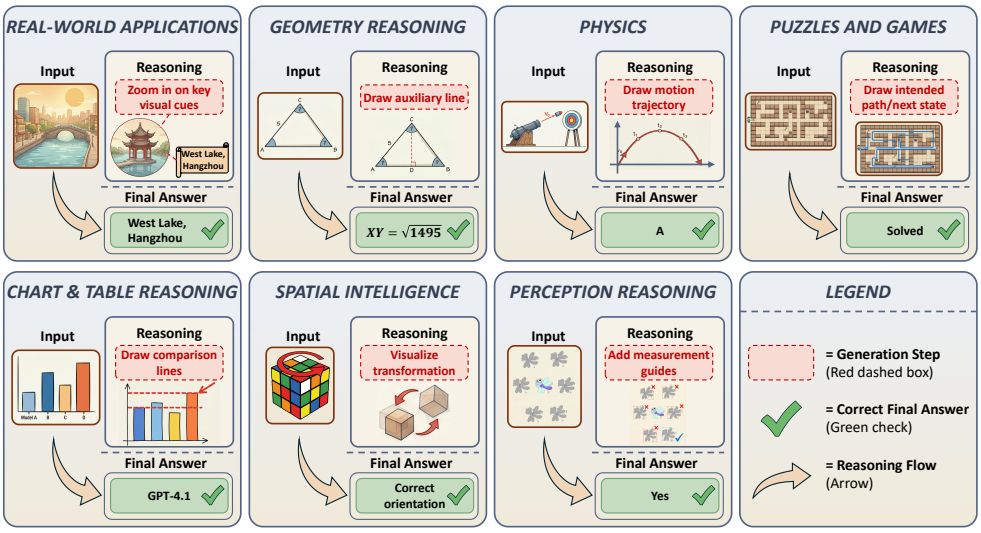


Figure 2: Task taxonomy overview of UniG2U.

Generate-then-answer and Δ_{GtA}. Under GtA, the model samples one intermediate $G \sim p_\theta(G \mid x)$ and answers with $p_\theta(y \mid x, G)$. Let $\bar{p}_\theta(y \mid x) = \sum_G p_\theta(G \mid x)\, p_\theta(y \mid x, G)$ be the implied marginal and $p_\theta(G \mid x, y)$ the corresponding posterior.

Proposition 3.2 (Commitment cost)**.** *For every* (x, y),

$$\mathbb{E}_G\big[\log p_\theta(y \mid x, G)\big] - \log q_\theta(y \mid x) = \underbrace{\log \frac{\bar{p}_\theta(y \mid x)}{q_\theta(y \mid x)}}_{\textit{externalization}} - \underbrace{\mathrm{KL}\big(p_\theta(G \mid x) \,\|\, p_\theta(G \mid x, y)\big)}_{\textit{commitment} \,\geq\, 0}. \quad (4)$$

The identity holds for log-likelihoods of the correct answer and does not directly determine changes in accuracy. It separates two effects. The commitment cost is zero exactly when $p_\theta(y \mid x, G)$ is the same for all intermediates the model may generate, and it grows with how much the probability of the correct answer varies across sampled intermediates, so that a single sample may lead away from the correct answer; it does not measure whether an intermediate is correct. The externalization gain is positive only if answering from explicit intermediates is easier than answering directly. We expect the variation, and hence the cost, to be small when the model's intermediates are consistent across samples, as in transformation-intensive tasks such as mazes and sliding puzzles. The complete analysis and the proofs are given in Appendix G.

4 The Setup of UniG2U

4.1 Dataset Construction and Task Taxonomy

UniG2U is a diagnostic benchmark for studying when and why visual generation benefits multimodal understanding. It contains **3,000 samples** organized into **7 reasoning categories** and **30 fine-grained subtasks** (Figure 2). The taxonomy follows *reasoning structures* rather than dataset sources and covers both generation-friendly tasks, such as spatial reasoning and puzzles, and perception-heavy tasks, so that we can analyze when intermediate generation is beneficial, neutral, or harmful.

We select tasks where intermediate visual generation may plausibly assist reasoning: spatial reasoning, where models can externalize transformations or layouts; chart and table reasoning, where visual re-expression reduces cognitive load; puzzles and games, where intermediate states support state tracking; and physics and geometry, where auxiliary constructions support the solution. Candidates are drawn from IllusionBench, MMSI-Bench, VisualPuzzles, Geometry3K, Uni-MMMU,

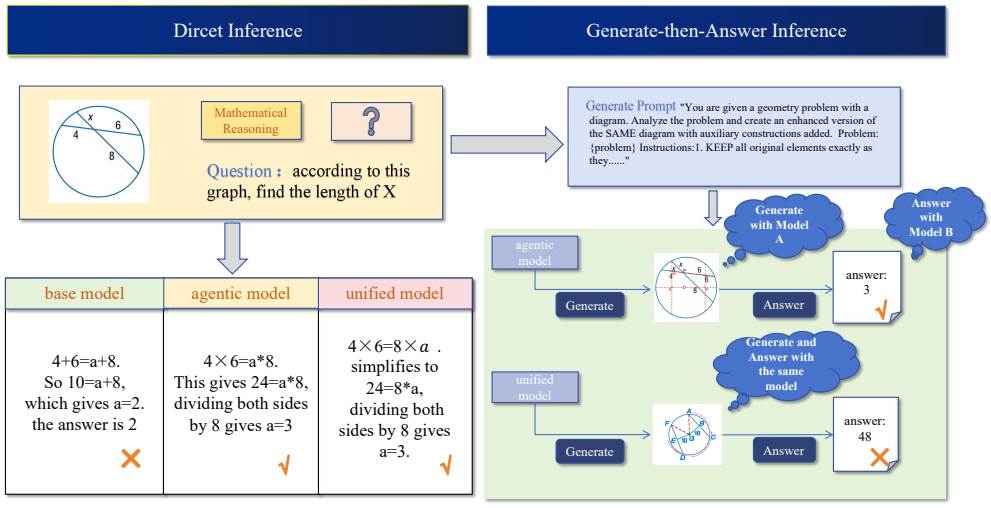


Figure 3: Overview of our two evaluation prompt formats: Direct and Generate-then-Answer (GtA).

ChartQA, RealUnify, BabyVision, PhyX, and VSP and pass a strict filtering pipeline (Appendix H). The benchmark contains 200 samples each for Real-world Applications, Geometry, and Physics, 537 for Puzzles and Games, 100 for Chart & Table, 500 for Spatial Intelligence, and 1,263 for Perception Reasoning. Although the number of samples per subtask may appear modest, it reflects the scarcity of tasks where intermediate visual generation genuinely supports reasoning, and UniG2U remains the largest benchmark designed for generation-to-understanding evaluation (Table 1). Each instance is a question-answering problem with image inputs, annotated with

- **category and subtask labels** for taxonomy-level analysis;
- **answer types** (multiple-choice, short text, or numeric) that determine the evaluation rules;
- **generation requirement tags** indicating whether intermediate visual reasoning is expected, such as a geometric construction or a state transition.

UniG2U does not prescribe a canonical intermediate image; the utility of generation is measured by its effect on the final answer.

4.2 Evaluation Protocol

Dual-protocol prompting. We evaluate all unified models under two paired protocols (Figure 3). Under **Direct** inference, the model predicts the answer from the input. Under **Generate-then-Answer (GtA)**, it first generates an intermediate visual artifact G and then answers conditioned on the input and G. For multi-step tasks such as mazes and sliding puzzles, an iterative GtA variant alternates between generating intermediate states and decisions until completion. Models without GtA support are treated as described in Appendix B.

Environment and budgets. Evaluations use the `lmms-eval` framework whenever possible, and the remaining models are run with their native stacks under identical prompts, budgets, and decoding settings. Maximum token lengths are capped, images are processed at default resolution, text decoding is greedy ($T = 0$, top-$p = 1.0$), and image generation follows each model's default configuration.

4.3 Models and Research Questions

We evaluate base VLMs, native unified models, and agentic systems that combine a planner with an external image generator. Each unified model is paired with its base VLM, or with the closest

available VLM in architecture and training data (Table 13). SenseNova-U1.5-8B-MoT, an encoder- and VAE-free Mixture-of-Transformers model with pixel-space flow matching for images and autoregressive text modeling (Diao et al., 2026), has no matched base VLM, so we report its absolute results without a G2U gain.

We address three research questions. **RQ1:** Do unified models outperform their paired base VLMs under Direct inference (Δ_{Direct}, overall and per category)? **RQ2:** Does explicit intermediate generation improve over Direct inference with fixed weights (Δ_{GtA})? **RQ3:** Are G2U gains correlated across tasks, and across models that share a base VLM or a generative architecture?

5 Results Analysis

5.1 Main Results (RQ1)

Table 2 presents the complete benchmark results across all evaluated models and the seven reasoning categories of UniG2U. Unified models are grouped with their paired base VLMs (marked with *), and Δ denotes the absolute G2U gain relative to the base model.[1] For SenseNova-U1.5, we report absolute scores under Direct and Visual CoT inference, where Visual CoT denotes its native generate-then-answer mode.

Takeaway 1: Integrating generation capabilities often degrades pure understanding performance.

Under Direct inference, 12 of the 16 paired unified models score below their base VLMs (one-sided sign test, $p = 0.038$). The direction holds both for models built on the language backbone or checkpoint family of their paired base (7 of 10) and for models paired with the closest available VLM (5 of 6). Because no intermediate image is generated in this mode, the gap is not caused by imperfect generated images. Paired models can still differ in pretraining data, instruction tuning, and implementation, so these comparisons establish an association with unification rather than the causal effect of generative training. One mechanism consistent with the gap is described by Proposition 3.1: wherever the generative gradient conflicts with the understanding gradient, joint training reduces the discriminative capability of the shared parameters. We refer to the observed gap as an *alignment tax*.

Takeaway 2: Despite overall degradation, unified models show consistent gains in spatial and illusion-related reasoning.

The impact of unification is not uniform across tasks. Table 3 illustrates this with the LLaVA-OneVision family, where unified models such as BAGEL and Show-o2 improve on subtasks involving geometric transformations, spatial tracking, and visual ambiguity.

For all four unified models of this family, the average over these five subtasks exceeds that of the base under every protocol they support, although individual subtasks vary. These tasks depend on spatial layout, shape, and geometric state, which a generative model must represent to synthesize the image, which is consistent with the alignment hypothesis of Section 3.

5.2 Comparison Between GtA and Direct Inference (RQ2)

To investigate whether explicit intermediate visual generation provides additional benefits over direct unified inference, we evaluate each unified model under both the Direct and GtA protocols with matched inference budgets.

Takeaway 3: The explicitly generation-augmented paradigm (GtA) degrades performance across most logic-intensive tasks, as forced visual externalization often propagates errors.

At the category level (Fig. 4a), most points fall below the $y = x$ line, indicating that GtA does not consistently outperform Direct across tasks. This pattern is also reflected in the aggregated results (Fig. 4b), where GtA accuracy is often lower than Direct for models supporting both protocols, with

[1] **Evaluation constraints:** To ensure fair comparisons, we exclude GtA results for models lacking image-editing training (OOD for visual modification). Direct results are omitted for models with frozen backbones, whose deviations stem from projection layers rather than parameter-level multimodal coupling. Detailed analyses are provided in Appendix B and A.

Table 2: **Main results on UniG2U across categories.** All values are reported as percentages ($\times 100$). The best result within each model group is **bolded**. Baseline models (marked with $*$) are highlighted with gray background. Δ denotes the performance gap to the respective baseline; it is not computed for unpaired SenseNova-U1.5.

Model	Real-world Apps	Geometry Reasoning	Physics	Puzzles & Games	Chart	Spatial Intel.	Perception Reasoning	Overall (G2U GainΔ)
qwen2.5-vl-7b*	40.25	**23.00**	34.00	25.73	**86.00**	23.40	39.42	**34.45** (+0.00)
OmniGen2(direct)	**41.25**	14.00	41.00	20.91	86.00	23.80	35.62	31.99 (-2.46)
OmniGen2(GtA)	41.25	13.00	**41.50**	21.91	79.00	**26.40**	34.52	31.87 (-2.58)
UniWorld-V1(GtA)	24.00	19.00	34.50	24.72	80.00	24.80	39.35	32.96 (-1.49)
OneCAT-3b(direct)	39.50	17.00	32.00	22.05	76.68	25.20	34.55	31.15 (-3.30)
OneCAT-3b(GtA)	33.50	8.50	29.00	20.69	74.00	23.80	33.10	28.80 (-5.65)
JavisGPT(direct)	32.00	21.00	26.50	23.72	78.60	21.60	35.52	30.72 (-3.73)
uni-video(GtA)	34.00	21.50	29.00	**29.33**	61.00	20.00	**42.76**	34.25 (-0.20)
UniPic2(GtA)	17.50	17.00	31.00	23.95	47.00	24.60	38.80	30.65 (-3.80)
STAR-7B(GtA)	34.00	14.50	32.50	23.56	79.00	25.20	38.56	32.68 (-1.77)
Qwen2.5-VL-3B*	41.00	15.50	**40.50**	20.42	87.00	**26.40**	35.55	**32.39** (+0.00)
UAE(direct)	36.75	10.50	30.50	19.06	85.00	25.20	**36.37**	30.94 (-1.45)
Ovis-U1(direct)	**45.50**	**23.50**	30.50	**27.28**	**90.00**	17.60	34.92	32.15 (-0.24)
Ovis-U1(GtA)	21.00	16.50	17.00	18.00	32.00	20.40	30.56	24.19 (-8.20)
yi-6B-vl*	24.00	0.00	33.50	13.22	19.00	22.60	31.20	23.73 (+0.00)
mio(direct)	18.00	0.00	31.00	12.85	15.00	**23.40**	25.49	20.70 (-3.03)
mio(GtA)	15.00	0.00	28.00	8.94	13.00	21.60	24.94	19.00 (-4.73)
deepseek-v2*	11.00	4.50	22.00	18.02	**74.00**	17.40	35.07	25.86 (+0.00)
Janus-Pro(direct)	**27.50**	7.50	19.00	22.49	29.00	24.20	35.08	27.39 (+1.53)
AIA(direct)	13.50	**8.50**	**31.00**	**23.95**	33.00	**26.60**	**36.34**	**28.65** (+2.79)
LLaDA-Instruct*	27.25	1.50	21.00	**20.83**	9.00	23.40	13.78	17.05 (+0.00)
MMaDA(direct)	**28.75**	**5.00**	**26.50**	16.01	**14.00**	**25.00**	**22.74**	**21.09** (+4.04)
Janus-1.3B*	13.50	**2.50**	**29.00**	**12.10**	**50.00**	21.80	**23.52**	**20.36** (+0.00)
FUDOK(direct)	**17.50**	1.00	21.00	9.31	22.00	**29.00**	15.52	16.40 (-3.96)
Qwen3-VL-8B*	**40.00**	**39.00**	**37.00**	**24.24**	89.00	**26.00**	**43.66**	**37.75** (+0.00)
MammothModa2(direct)	33.50	0.00	36.50	12.10	**91.00**	21.80	39.11	29.97 (-7.78)
llava-hf*	**23.00**	0.00	**29.50**	**18.96**	28.00	**24.60**	**33.57**	**26.06** (+0.00)
Emu3(direct)	16.50	**1.00**	20.50	13.37	**67.00**	23.20	24.78	21.46 (-4.60)
llava-onevision*	**43.00**	10.50	**44.00**	24.48	78.00	25.40	37.13	33.35 (+0.00)
X-Omni(direct)	33.00	9.50	29.50	26.24	83.00	27.00	35.31	31.63 (-1.72)
bagel(direct)	38.00	29.00	37.50	23.77	**92.00**	28.40	**39.96**	35.84 (+2.49)
bagel(GtA)	32.75	**30.00**	40.50	**31.58**	91.00	**29.00**	37.29	**36.10** (+2.75)
show-o2(direct)	35.50	11.00	40.00	26.01	45.00	25.80	36.50	31.59 (-1.76)
show-o2(GtA)	33.00	9.00	40.00	26.39	35.00	20.40	28.11	26.59 (-6.76)
ILLUME+(direct)	24.50	7.50	35.00	21.06	80.00	23.20	35.08	29.54 (-3.81)
ILLUME+(GtA)	20.50	2.00	35.50	19.53	58.00	21.00	34.05	27.13 (-6.22)
Unpaired native unified model (8B MoT; no matched base VLM)								
SenseNova-U1.5-8B-MoT (Direct)	**42.50**	44.50	32.50	22.72	**85.00**	29.20	38.82	36.08 (N/A)
SenseNova-U1.5-8B-MoT (Visual CoT)	39.50	**45.50**	**34.00**	**24.83**	82.00	**30.00**	**40.42**	**37.13** (N/A)
gpt4o	**45.25**	27.00	34.50	**30.73**	58.00	**36.00**	**43.73**	**38.96**
ovis2.5	44.00	**32.50**	**43.00**	23.13	**89.00**	27.80	42.28	37.51
GPT-4o + GPT-image	32.75	27.00	30.50	32.72	60.00	27.00	40.54	35.44
gemini pro +nano banana pro	**71.50**	**85.00**	**91.00**	**55.31**	68.00	**38.40**	**61.12**	**60.80**
Qwen2.5-7b+Qwen-edit	33.75	17.00	31.50	27.79	**80.00**	21.60	38.56	32.96

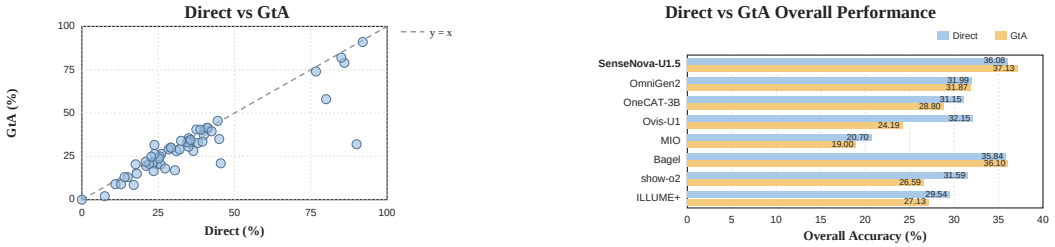
Note: * denotes the base model. Δ is computed as overall minus the base model's overall within each block; SenseNova is unpaired and has no Δ. **Bold:** Within each block (same model family), the best score in each category is highlighted in bold; ties are broken by bolding the first occurrence.

substantial degradation observed in models such as OneCAT-3B, Ovis-U1, and MIO. Overall, six of the eight models with both protocols score lower under GtA.

This trend is consistent with Proposition 3.2. Direct inference uses generation-trained representations in latent space without committing to a visual hypothesis, whereas GtA commits to a single sampled intermediate. When generation is uncertain or inconsistent with the correct answer, the commitment cost in Equation (4) can exceed the externalization gain, and the failure cases in Appendix E show errors in the intermediate propagating to the final prediction.

Table 3: **Excerpt of Absolute Results on Selected Illusion and Spatial Subtasks.** Baseline model is highlighted in gray. See Appendix C for the comprehensive table encompassing all model blocks.

Model	Illusion		Spatial Intelligence			Overall (Δ)
	icon_shape	in_shape	MSR	Attr. (Meas.)	Motion (Cam.)	
llava_onevision*	6	18	17	26	20	17.40 (+0.00)
X-Omni(direct)	5	11	20	31	24	18.20 (+0.80)
bagel(direct)	2	18	26	31	26	**20.60** (+3.20)
bagel(GtA)	2	17	25	36	24	20.80 (+3.40)
show-o2(direct)	11	22	27	24	28	22.40 (+5.00)
show-o2(GtA)	6	18	30	15	23	18.40 (+1.00)
ILLUME+(direct)	4	20	28	29	14	19.00 (+1.60)
ILLUME+(GtA)	14	16	29	28	16	20.60 (+3.20)



(a) Category-level comparison

(b) Overall accuracy comparison

Figure 4: **GtA vs. Direct inference comparison.** (a) Category-level results where each point represents a (model, category) pair comparing Direct and GtA accuracy. (b) Overall accuracy comparison for models supporting both inference protocols.

Takeaway 4: Explicit intermediate generation (GtA) yields noticeable performance gains in transformation-intensive scenarios where visual states serve as an external workspace.

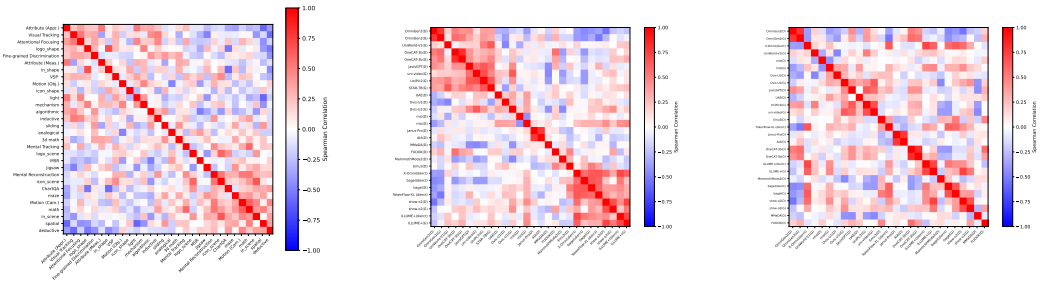
Despite the overall decline, GtA can help on transformation-intensive subtasks such as Maze navigation, Sliding puzzles, and Multi-Step Spatial Reasoning (MSR). Appendix N reports the exact per-model scores. These tasks involve long reasoning horizons where intermediate spatial states must be continuously updated.

In these tasks, each intermediate state is nearly determined by the current state and action, so the commitment cost is expected to be small, while tracking the state explicitly is easier than tracking it implicitly. For mazes and sliding puzzles, which require a sequence of moves, we report the partial success rate, that is, the number of correctly completed steps divided by the total number of steps, averaged over instances. BAGEL's partial success rate rises from 2.1% to 28.1% on mazes and from 0.9% to 19.5% on sliding puzzles (Table 14). Explicit generation then acts as a visual workspace, provided that the intermediate images remain structurally accurate.

To separate image quality from answering, we evaluate manually checked GPT-image-2 reference intermediates. Accurate intermediates remove generation errors but do not change how well the model answers from an intermediate. Consistent with this, gains are most consistent on Illusion-Shape, where answering from an explicit shape is easy, and mixed on other subtasks (Table 16).

5.3 Behavioral Correlation across Tasks and Models (RQ3)

While RQ1 and RQ2 analyze absolute performance and inference protocols, RQ3 examines whether G2U gains (Δ) exhibit structured correlation patterns across tasks and models.



(a) Task-level correlation (b) Model-level correlation (by Base VLM) (c) Model-level correlation (by Architecture)

Figure 5: **Spearman correlation heatmaps of G2U gains (Δ).** (a) Task-level correlations form two clusters—perception and reasoning—which are internally aligned but often negatively correlated. (b) Models sharing the same base VLM exhibit highly similar G2U behaviors. (c) Grouping by generative architecture yields weaker correlations, indicating that backbone representations heavily influences G2U effects.

Takeaway 5: Perception-oriented and reasoning-oriented tasks form internally correlated clusters but often exhibit negative correlations with each other. At the model level, unified models sharing the same base VLM show strong behavioral similarity, whereas models sharing only architectural paradigms display weaker correlations.

To study generation-associated differences, we analyze the relative offset Δ (the performance difference between a unified model and its paired base VLM). This paired contrast accounts for the base model's measured performance, but does not remove all confounds arising from differences in training or implementation.

Task-Level Correlation We compute pairwise Spearman correlations between subtasks using their Δ vectors across models. As shown in Fig. 5a, subtasks cluster into two groups: structural reasoning tasks (e.g., jigsaw, maze) and perception-oriented tasks. Correlations are positive within each group but frequently negative across groups, indicating a trade-off between perceptual discrimination and structural reasoning.

Model-Level Correlation We further compute correlations between models using their Δ vectors across subtasks. Models sharing the same base VLM form strong high-correlation blocks (Fig. 5b), while grouping by generative architecture produces weaker patterns (Fig. 5c). Because models paired with the same base share the term $-\mathcal{B}(M)$ in their gains, we also regress each model's subtask scores on those of its base and correlate the residuals. Models sharing a base VLM remain correlated after this control (mean Spearman $\rho = 0.31$ within groups versus 0.07 across groups, permutation $p < 0.001$), whereas grouping by generative architecture shows no such difference (Appendix D). This suggests that G2U behavior is primarily governed by backbone representations rather than generative paradigms.

6 Conclusion and Future Work

UniG2U evaluates when visual generation aids understanding across 3,000 instances, seven reasoning categories, and over 30 models. Unified models generally underperform their base VLMs, yet improve on spatial and shape-sensitive tasks, and explicit intermediate generation helps mainly on transformation-intensive tasks. Our likelihood analysis relates these findings to two quantities, the alignment between generative and understanding gradients and the commitment cost of answering from a single generated intermediate; measuring them directly is a natural next step. The analysis also suggests two directions for unified models: reweighting or restructuring the generative objective toward answer-relevant structure, and marginalizing over several intermediates instead of committing to one.

Reproducibility Statement

Section 3 defines the G2U gains and states all theoretical results, whose proofs are given in Appendix G. The construction, sources, and filtering of the benchmark are described in Appendix H, and the evaluation protocol, decoding settings, and inference budgets in Section 4.2. The complete prompts for Direct and Visual CoT inference are listed in Appendices K and L, the judge-based RA and AL evaluation pipeline with its full prompts in Appendix M, and the model roster, pairing, and configurations in Appendix N. The benchmark data and evaluation code are provided as supplementary material.

Ethics Statement

UniG2U is constructed from publicly available benchmarks and contains no personal information, and the study involves no human subjects. All models are evaluated through their publicly released weights or official APIs under their respective terms of use.

AI Use Statement

In this work, we used generative AI tools for drafting and editing parts of the paper and for assisting in the formulation and presentation of the theoretical analysis in Section 3 and its proofs. As part of the evaluation methodology, GPT-5.5 serves as an automatic judge for the RA and AL metrics (Appendix M) and GPT-image-2 generates the reference intermediates used in the control experiment (Table 16); both uses are described in the paper. We have not used generative AI tools for generating or cleaning the benchmark data, designing the experiments, or implementing the evaluation, and the remaining tasks with required disclosure are not applicable to this work. We have reviewed all AI-assisted work: the theoretical results and proofs were verified by the authors, and all numbers were checked against the evaluation records. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Chameleon Team, Xi Chen, et al. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. Charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, Fangfu Liu, Tianyu Liu, Baobao Chang, Xiaobo Hu, Kaiyuan Chen, Yixin Ren, Yang Liu, Yuan Gong, and Kuan Li. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

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

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. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Jiahao Wang, Chenjing Ding, et al. Sensenova-u1.5: Towards native unified visual intelligence. *arXiv preprint arXiv:2609.11929*, 2026.

Shasha Guo, Jia Li, Yi Wang, et al. Geovlmath: Enhancing geometry reasoning in vision-language models via cross-modal reward for auxiliary line creation. *OpenReview (ICLR 2026)*, 2026.

Kyle Hsu, Huy Pham, Inbar Guy, Ari Morcos, and Jimmy Ba. Visual scratchpads: Enabling global reasoning with distributed visual representations. In *Conference on Robot Learning (CoRL)*. PMLR, 2023.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024.

Bo Li, Yida Yin, Wenhao Chai, Xingyu Fu, and Zhuang Liu. Ueval: A benchmark for unified multimodal generation. *arXiv preprint arXiv:2601.22155*, 2026.

Ming Li, Xiaoli Wang, Qiang Zhang, et al. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for mllms. *arXiv preprint arXiv:2409.18520*, 2025a.

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

Yongyuan Liang, Wei Chow, Feng Li, Ziqiao Ma, Xiyao Wang, Jiageng Mao, Jiuhai Chen, Jiatao Gu, Yue Wang, and Furong Huang. Rover: Benchmarking reciprocal cross-modal reasoning for omnimodal generation. *arXiv preprint arXiv:2511.01163*, 2025.

Bin Lin et al. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.177.

Yang Shi, Yuhao Dong, Yue Ding, Yuran Wang, Xuanyu Zhu, Sheng Zhou, Wenting Liu, Haochen Tian, Rundong Wang, Huanqian Wang, Zuyan Liu, Bohan Zeng, Ruizhe Chen, Qixun Wang, Zhuoran Zhang, Xinlong Chen, Chengzhuo Tong, Bozhou Li, Chaoyou Fu, Qiang Liu, Haotian Wang, Wenjing Yang, Yuanxing Zhang, Pengfei Wan, Yi-Fan Zhang, and Ziwei Liu. Realunify: Do unified models truly benefit from unification? a comprehensive benchmark. *arXiv preprint arXiv:2509.24897*, 2025.

Rui Tian, Mingfei Gao, Mingze Xu, Jiaming Hu, Jiasen Lu, Zuxuan Wu, Yinfei Yang, and Afshin Dehghan. Unigen: Enhanced training and test-time strategies for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.14682*, 2025.

Fei Wang, Qiang Li, Yu Zhao, et al. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *arXiv preprint arXiv:2504.10342*, 2025a.

Guo-Hua Wang et al. Ovis-u1: Unified understanding and generation with discrete image representation. *arXiv preprint arXiv:2506.23044*, 2025b.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.15920*, 2024.

Jingxuan Wei, Caijun Jia, Xi Bai, Xinglong Xu, Siyuan Li, Linzhuang Sun, Bihui Yu, Conghui He, Lijun Wu, and Cheng Tan. Ggbench: A geometric generative reasoning benchmark for unified multimodal models. *arXiv preprint arXiv:2511.11134*, 2025.

Shitao Xiao, Guan-Zheng Wang, et al. Omnigen: Unified image generation with diffusion language models. *arXiv preprint arXiv:2409.11314*, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

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

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

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. Mmsi-bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*, 2025a.

Yan Yang, Haochen Tian, Yang Shi, Wulin Xie, Yi-Fan Zhang, Yuhao Dong, Yibo Hu, Liang Wang, Ran He, Caifeng Shan, et al. A survey of unified multimodal understanding and generation: Advances and challenges. *Authorea Preprints*, 2025b.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. *arXiv preprint arXiv:2311.16502*, 2023.

Lei Zhang, Hua Chen, Xin Liu, et al. Phyx: Does your model have the "wits" for physical reasoning? *arXiv preprint arXiv:2505.15929*, 2025a.

Yiming Zhang, Zicheng Zhang, Xinyi Wei, Xiaohong Liu, Guangtao Zhai, and Xiongkuo Min. Illusionbench: A large-scale and comprehensive benchmark for visual illusion understanding in vision-language models. *arXiv preprint arXiv:2501.00848*, 2025b.

Yu Zhong, Tianwei Lin, Ruike Zhu, Yuqian Yuan, Haoyu Zheng, Liang Liang, Wenqiao Zhang, Feifei Shao, Haoyuan Li, Wanggui He, Hao Jiang, and Yueting Zhuang. Unified personalized understanding, generating and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29400–29409, June 2026.

Kai Zou, Ziqi Huang, Yuhao Dong, Shulin Tian, Dian Zheng, Hongbo Liu, Jingwen He, Bin Liu, Yu Qiao, and Ziwei Liu. Uni-mmmu: A massive multi-discipline multimodal unified benchmark. *arXiv preprint arXiv:2510.13759*, 2025.

A Performance Analysis of Unified Models with Frozen Backbones (Non-CoT Version)

Table 4: **Direct (Non-CoT) results of frozen-backbone models on UniG2U.** All values are reported as percentages (×100). The Overall column is the average over task categories weighted by their numbers of samples.

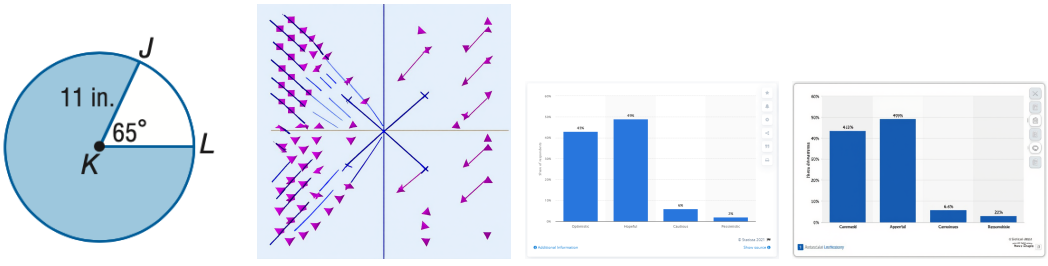
Model	Real-world Apps	Geometry Reasoning	Physics	Puzzles & Games	Chart	Spatial Intel.	Perception Reasoning	Overall
UniPic2(direct)	36.50	23.50	36.00	24.48	67.00	24.40	39.43	33.68
UniWorld-V1(direct)	35.50	22.00	33.00	25.62	65.00	27.40	36.66	32.79
STAR-7B(direct)	32.50	19.00	30.50	26.25	86.00	25.40	38.56	33.50
Uni-Video(direct)	30.00	20.50	29.50	27.16	81.00	26.00	39.67	33.93

Table 4 presents the Direct (Non-CoT) results of unified models that employ a frozen backbone. Theoretically, since the generative pathway is bypassed and the base vision-language model (VLM) backbone remains frozen during training, their performance should exactly match that of the base model (*qwen2.5-vl-7b*). However, empirical results show measurable deviations. We attribute these discrepancies to several implementation-level factors. First, subtle weight differences can emerge if a unified model utilizes Exponential Moving Average (EMA) checkpoints rather than the exact original base model weights, leading to slight performance drifts. Second, variations in inference hyperparameters—such as maximum generation length, input image resolution scaling, and decoding strategies—can significantly alter the final predictions. Third, inconsistencies in the software environment, such as different versions of PyTorch, CUDA, or the transformers library, often introduce deterministic execution variances. Furthermore, differences in visual pre-processing pipelines, particularly image encoding formats (e.g., reading raw images via PIL versus applying JPEG compression), affect pixel-level inputs. Finally, operating in different numerical precisions can cause compounding errors. For instance, computing Rotary Position Embeddings (RoPE) in bfloat16 instead of float32 can accumulate numerical errors over long sequences, eventually shifting the output logits and altering the results.

B Ablation Study for Models without Edit Training

Model	Real	Geometry	Physics	Puzzle	Chart	Spatial	Perception	Overall
UAE (Direct)	36.75	10.50	30.50	19.06	85.00	25.20	36.37	30.94
UAE (GtA)	23.00	7.50	32.50	20.33	71.00	22.80	34.68	28.61 (-2.33)
X-Omni (Direct)	33.00	9.50	29.50	26.24	83.00	27.00	35.31	31.63
X-Omni (GtA)	16.50	4.00	29.00	23.31	22.00	20.20	19.72	19.87 (-11.76)
TokenFlow-XL (Direct)	19.00	10.50	46.00	14.97	43.00	21.80	38.88	29.15
TokenFlow-XL (GtA)	14.50	9.50	26.50	10.06	38.00	21.80	34.28	24.50 (-4.65)

Table 5: Results (values ×100). Blue numbers indicate the overall drop relative to the corresponding Direct setting.



(a) Question Image (b) UAE Generated Image (c) Question Image (d) X-Omni Generated Image

Figure 6: Examples of erroneous generation from edit-untrained models.

As discussed in Section 5, not all unified models evaluated in this work are trained with explicit image-to-image (edit) capabilities. For such models, Generate-then-Answer generation that requires intermediate image editing inherently falls outside their training distribution, effectively constituting an out-of-distribution (OOD) setting. Including their Generate-then-Answer results directly in the main comparison table could therefore introduce an unfair bias.

To better understand the effect of this OOD intermediate generation, we conduct additional ablation studies on a representative subset of edit-untrained models, with the results summarized in Table 5. Specifically, we categorize their Generate-then-Answer evaluation into two alternative variants:

(1) OOD-Image Generation (UAE and X-Omni). Despite lacking edit training, the first two models are forced to generate an intermediate image under the standard Generate-then-Answer protocol (i.e., conditioned on the original input image). The generated image is then re-consumed for answering. This setting measures how severely forcing OOD intermediate generation degrades downstream reasoning performance.

(2) Prompt-Only Intermediate Generation (TokenFlow-XL). Instead of feeding the original input image into the intermediate generation stage, we restrict the final model to producing an auxiliary image conditioned strictly on the textual generation prompt (i.e., no image-to-image conditioning). The generated image is subsequently used as additional context for answering. This setting examines prompt-level generation without invoking explicit edit-style image transformations.

Findings. As demonstrated in Table 5, both Generate-then-Answer variants consistently lead to overall performance degradation compared to their Direct inference counterparts. Specifically, forcing OOD-Image generation causes overall performance drops of 2.33% and 11.76% for UAE and X-Omni, respectively, while relying on Prompt-Only generation leads to a 4.65% drop for TokenFlow-XL.

Furthermore, the qualitative examples presented in Figure 6 corroborate these quantitative declines. Visual inspection reveals that the intermediate images produced by these edit-untrained models are often completely chaotic, structurally irrelevant, or fail to encode the necessary information. Instead of aiding the reasoning process, these misaligned generations actively mislead the subsequent answering stage.

Overall, these ablation results validate our decision to exclude the Generate-then-Answer scores of edit-untrained models from the main comparison table. They strongly emphasize that Generate-to-Understand (G2U) gains rely critically on explicit and training-aligned intermediate generation capabilities.

C Additional Result Analysis: Structured Subtask Gains

In this section, we analyze the complete experimental results (Table 9) to identify subtasks where generation-to-understanding (G2U) gains are particularly evident.

While overall improvements are not universal, we observe that unified models demonstrate clearer and more repeatable gains on specific structured subtasks. In particular, **Spatial Intelligence** and **illusion-related perceptual reasoning** subtasks show more consistent positive offsets compared to base VLMs. These categories require transformation-aware reasoning, geometric consistency, or resolving visually ambiguous patterns where generation-trained representations appear to regularize internal spatial structure.

For example, in the *spatial* subtask under Perception Reasoning, multiple unified variants (e.g., UniWorld-V1, JavisGPT, uni-video, STAR-7B) achieve higher scores than the corresponding base model. Similarly, illusion-sensitive discrimination subtasks (e.g., logo-related and shape-sensitive categories) exhibit non-trivial improvements in certain unified configurations. These gains suggest that generation-induced representation coupling may enhance the model's robustness to visually ambiguous or structurally complex inputs.

In contrast, purely recognition-heavy subtasks (e.g., icon_scene, in_scene) do not show systematic improvement and sometimes experience mild degradation. This further supports our central claim: G2U benefits are not uniform but emerge in transformation-centric or structurally constrained categories.

Table 9 provides the detailed subtask-level results supporting this analysis.

D Correlation Analysis with Baseline Control

Data. We use the subtask-level results of Tables 9 and 10 for all 27 unified-model rows that have a paired base VLM and complete results on the 30 subtasks, covering both the Direct and the GtA protocol. Agentic systems, API models, and the unpaired SenseNova-U1.5 are excluded. For model m with base $b(m)$, let $\mathbf{u}_m, \mathbf{u}_{b(m)} \in \mathbb{R}^{30}$ be the subtask scores and $\mathbf{\Delta}_m = \mathbf{u}_m - \mathbf{u}_{b(m)}$ the gain vector. Within-group pairs exclude the two protocols of the same model.

Table 6: **Absolute results on selected Illusion and Spatial subtasks.** All values are reported as percentages ($\times 100$). Baseline models (marked with $*$) are highlighted with gray background. **Overall** is computed as the average over the 5 shown subtasks, and the value in parentheses denotes the gap to the corresponding baseline within each model block. The best **Overall** within each block is **bolded**.

Model	Illusion		Spatial Intelligence			Overall (△)
	icon_shape	in_shape	MSR	Attr. (Meas.)	Motion (Cam.)	
qwen2.5-vl-7b*	5	10	25	20	29	17.80 (+0.00)
OmniGen2(direct)	6	8	22	33	12	16.20 (-1.60)
OmniGen2(GtA)	6	11	29	33	18	19.40 (+1.60)
UniWorld-V1(GtA)	4	10	27	22	21	16.80 (-1.00)
OneCAT-3b(direct)	2	8	32	30	24	19.20 (+1.40)
OneCAT-3b(GtA)	1	7	39	28	3	15.60 (-2.20)
JavisGPT(direct)	12	13	16	24	19	16.80 (-1.00)
uni-video(GtA)	18	36	23	18	10	**21.00** (+3.20)
UniPic2(GtA)	8	24	21	23	23	19.80 (+2.00)
STAR-7B(GtA)	7	13	27	22	21	18.00 (+0.20)
Qwen2.5-VL-3B*	6	9	22	32	18	**17.40** (+0.00)
UAE(direct)	7	8	26	22	20	16.60 (-0.80)
Ovis-U1(direct)	7	10	12	28	18	15.00 (-2.40)
Ovis-U1(GtA)	5	15	27	27	11	17.00 (-0.40)
yi-6B-vl*	17	16	24	26	11	**18.80** (+0.00)
mio(direct)	5	13	21	24	23	17.20 (-1.60)
mio(GtA)	15	16	32	24	22	21.80 (+3.00)
deepseek-v2*	8	13	23	19	14	15.40 (+0.00)
Janus-Pro(direct)	4	21	28	37	16	21.20 (+5.80)
AIA(direct)	5	18	23	43	19	**21.60** (+6.20)
LLaDA-Instruct*	0	8	21	41	11	16.20 (+0.00)
MMaDA(direct)	1	3	32	27	20	**16.60** (+0.40)
Janus-1.3B*	6	10	15	28	18	15.40 (+0.00)
FUDOK(direct)	10	5	29	28	29	**20.20** (+4.80)
Qwen3-VL-8B*	13	22	23	35	20	**22.60** (+0.00)
MammothModa2(direct)	5	10	14	19	18	13.20 (-9.40)
llava-hf*	10	20	28	24	14	**19.20** (+0.00)
Emu3(direct)	0	2	31	23	14	14.00 (-5.20)
llava_onevision*	6	18	17	26	20	17.40 (+0.00)
X-Omni(direct)	5	11	20	31	24	18.20 (+0.80)
bagel(direct)	2	18	26	31	26	**20.60** (+3.20)
bagel(GtA)	2	17	25	36	24	20.80 (+3.40)
show-o2(direct)	11	22	27	24	28	22.40 (+5.00)
show-o2(GtA)	6	18	30	15	23	18.40 (+1.00)
ILLUME+(direct)	4	20	28	29	14	19.00 (+1.60)
ILLUME+(GtA)	14	16	29	28	16	20.60 (+3.20)

Shared-baseline artifact. For two models with the same base, $\mathbf{\Delta}_m$ and $\mathbf{\Delta}_{m'}$ share the term $-\mathbf{u}_b$, which induces correlation even when $\mathbf{u}_m$ and $\mathbf{u}_{m'}$ are unrelated. To quantify it, we take every pair of models from different groups, subtract the same base from both for each of the nine base VLMs, and compute the Spearman correlation of the resulting vectors. This shared-baseline reference has mean 0.45 (standard deviation 0.23), which exceeds the raw within-group correlation of $\mathbf{\Delta}$ (0.41). Raw Δ correlations therefore cannot distinguish shared behavior from the shared baseline.

Controlled correlation. We regress $\mathbf{u}_m$ on $[\mathbf{1}, \mathbf{u}_{b(m)}]$ by least squares and use the residual $\mathbf{r}_m$, which removes any component of the model's scores that is linear in its base scores. We compute Spearman correlations between residuals and compare the mean over within-group pairs with the mean over cross-group pairs. The p-value is the fraction of 5,000 random permutations of the group labels with a within–cross difference at least as large as the observed one.

Table 7: **Mean Spearman correlation of G2U behavior between unified models.** Raw: correlation of $\mathbf{\Delta}$; controlled: correlation of residuals after regressing on the base scores.

Grouping	Quantity	Within groups	Across groups	Permutation p
Base VLM	raw $\mathbf{\Delta}$	0.41	0.00	–
Base VLM	controlled	0.31	0.07	< 0.001
Architecture	raw $\mathbf{\Delta}$	0.07	0.06	0.26
Architecture	controlled	0.08	0.11	0.73
Shared-baseline reference for raw $\mathbf{\Delta}$: 0.45 ± 0.23				

E Failure Case Taxonomy for Generation-Augmented Understanding

In this appendix, we present representative failure cases of generation-augmented reasoning. We organize all examples under a unified three-way taxonomy based on the relationship between generated diagrams and downstream reasoning performance.

Why failure cases matter. Generation-augmented understanding is often motivated by the intuition that "drawing helps thinking": a faithful diagram can externalize latent structure, reduce working-memory burden, and provide geometric/physical constraints that guide computation. However, *visual generation is itself a structured prediction problem*. When the generated artifact is even slightly inconsistent with the underlying task, the downstream solver may inherit or amplify those errors (e.g., by trusting a distorted geometry, an incorrect scale, or a spurious path). Thus, it is important to characterize *how* generation fails, not merely *whether* an answer is wrong.

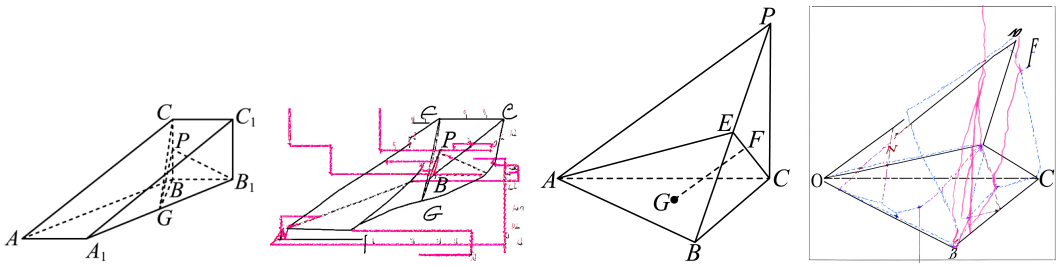
Selection protocol. We curate cases from diverse domains (geometry, physics, charts, and factual QA) to ensure the taxonomy covers (i) tasks where diagrams are essential (geometry/physics), (ii) tasks where diagrams can be helpful but must encode precise structure (charts), and (iii) tasks where diagrams are unnecessary (pure textual knowledge). For each case, we pair the original prompt with a model-generated visual output and highlight the key mismatch that explains downstream failure.

Three criteria: validity, relevance, and utility. Across domains, successful generation must satisfy three requirements: (i) **structural validity** (the diagram is internally consistent and respects physical/geometric rules), (ii) **semantic relevance** (the content corresponds to the task entities and relations), and (iii) **reasoning utility** (the diagram adds constraints or reduces ambiguity in a way that supports solving). Our taxonomy maps naturally to violations of these criteria: Category I breaks validity, Category II satisfies relevance but lacks utility, and Category III fails relevance altogether.

Category I: Capability Failure. The model fails to generate structurally correct or quantitatively valid diagrams. Rendered visuals are geometrically distorted, structurally inconsistent, or physically invalid, making downstream reasoning unreliable or impossible. This category is the most direct failure mode: even if the image appears "diagram-like", it violates invariants required by the task.

Common symptoms.

- *Geometric inconsistency:* angles/parallel lines/circle tangency implied by the prompt are not preserved.



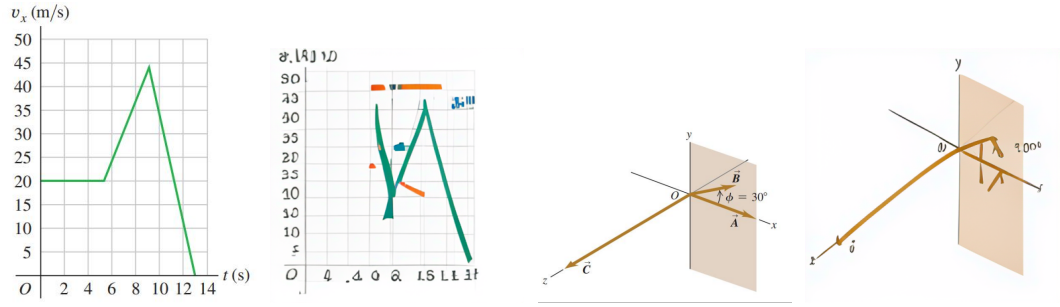
Original
Find the length of BP

Generated
Bagel

Original
When the length of PC is such that the dihedral angle $E - AC - B$ is 60ř, determine its value

Generated
Bagel

Figure 7: Capability failure. The model produces geometrically inconsistent or physically invalid diagrams, preventing correct downstream reasoning.



Original (Case 1)
How far in first 13 s? *Options:* A 320m, B 230m, C 280m, D 300m.

Generated (Case 1)
STAR-7B.

Original (Case 2)
$\vec{A}$: mag 6 along $+x$; $\vec{B}$: mag 4 in xy. *Options:* A $15\hat{k}$, B $12\hat{k}$, C $8\hat{k}$, D $18\hat{k}$.

Generated (Case 2)
STAR-7B.

Figure 8: Capability failure in physics reasoning. Generated diagrams fail to preserve quantitative/structural relations required for correct integration- or geometry-based computation.

- *Quantitative distortion:* lengths, slopes, or axis scales are visually inconsistent with stated values.
- *Physical invalidity:* forces/velocities are drawn with wrong directions or incompatible magnitudes.
- *Missing constraints:* key elements (labels, right-angle marks, reference points) are omitted or misplaced.

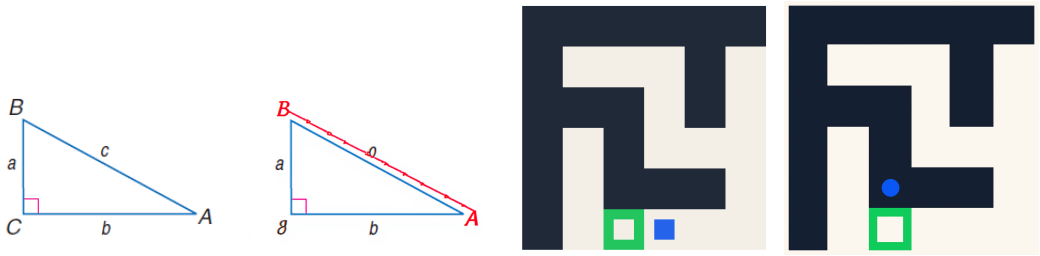
Why it breaks reasoning. Even a strong textual solver may implicitly treat the diagram as ground truth. When the generated visual is invalid, the solver can be pushed toward incorrect intermediate computations (e.g., wrong triangle similarity, wrong area under a curve, wrong cross-product direction), or may be forced to ignore the diagram entirely, losing the intended benefit of externalization.

Figure 7 shows two geometry instances where the generated diagrams violate the required geometric/physical constraints.

Figure 8 presents two physics cases where the generated visuals fail to preserve quantitative/structural relations needed for integration- or geometry-based computation.

Implication. Category I indicates a bottleneck in the generator's *diagram grammar*: models may learn surface styles (lines, arrows, labels) without learning domain invariants. A practical mitigation is to introduce lightweight validation or self-checking (e.g., verify right angles, axis monotonicity, connectivity constraints) before passing the diagram to the solver, or to route such instances back to a pure-text solution when validity is uncertain.

Category II: Surface-Relevant but Non-Utility Generation. The model generates diagrams that are semantically related to the prompt, yet they add no constraints or computational benefit for solving. In these



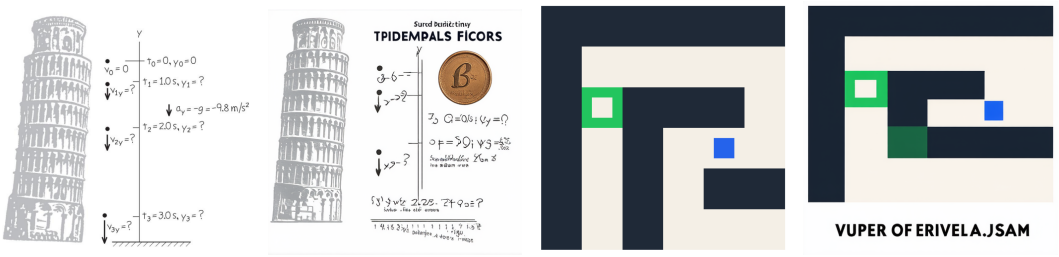
Original (Geom)

Generated (Geom)
Surface-relevant but adds no utility(Bagel).

Original (Maze)

Generated (Maze)
No valid connectivity/path(GPT-4o + GPT-image).

Figure 9: Surface-relevant but non-utility generation. Outputs are related to the prompt but provide no additional constraints or computational benefit.



Original (Case 1)
Coin dropped from rest. Position after 3.0 s?
Options: A -35m, B -44m, C -65m, D -51m.

Generated (Case 1)
UniPic2-Metaquery-9B.

Original (Case 2)

Generated (Case 2)
UniPic2-Metaquery-9B.

Figure 10: Surface-triggered generation. The model draws a "coin" due to lexical cues, while the task requires kinematic computation; the generated visuals provide little to no reasoning utility.

cases, the generation is "on-topic" but *not instrumental*: the diagram either restates what is already explicit in text, or fails to encode the specific structure needed.

Common symptoms.

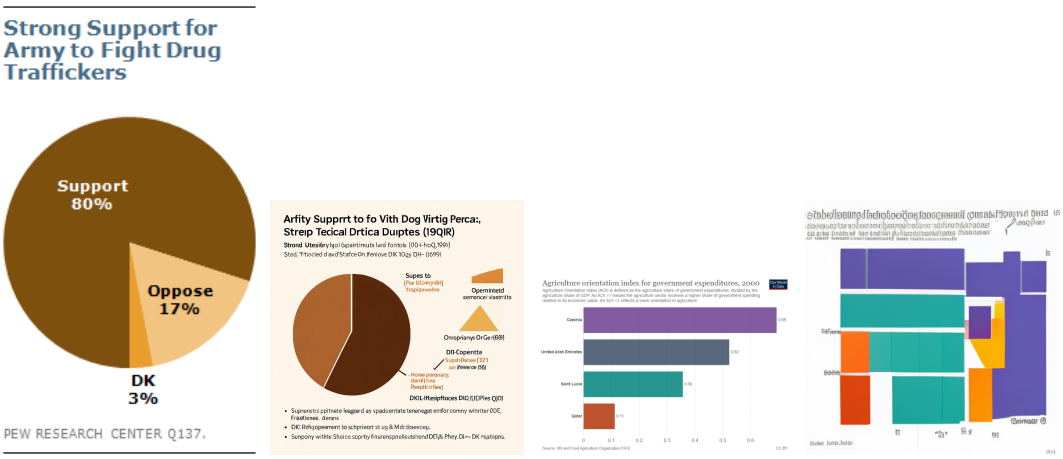
- *Correct but redundant:* the diagram matches the description but does not reduce uncertainty.
- *Missing operational structure:* key relations for computation (e.g., graph scales, maze connectivity) are absent.
- *Aesthetic completion:* the model fills the page with plausible visuals rather than task-sufficient constraints.

Why it fails to help. Utility is not guaranteed by relevance: even a perfectly relevant illustration can be useless if it does not support the specific operation needed (e.g., reading off a slope, tracing a path, or identifying a computable sub-structure). This failure mode is particularly important because it can *waste* generation budget without improving accuracy, and may even distract downstream reasoning by introducing extra, non-actionable details.

Figure 9 shows cases where the output looks on-topic but does not help: one is correct yet uninformative, and the other lacks valid connectivity/path structure.

Figure 10 illustrates surface-triggered generation in kinematics: lexical cues (e.g., "coin") induce drawing, but the task requires computation, so the visuals contribute little to reasoning.

Implication. Category II suggests that "draw something relevant" is an insufficient objective. For G2U, the generator must be *goal-conditioned*: it should produce the minimal structure that makes the answering computation easier (e.g., annotated axes, explicit path graph, marked equal angles), rather than a generic illustration.



Original (ChartQA) | **Generated (ChartQA)** | **Original (Know, C1)** | **Generated (Know, C1)**

Not a structured chart(GPT-4o + GPT-image).

What is the value of Czechia? One word.

STAR-7B.

Figure 11: Irrelevant generation. Outputs are misaligned with the task: chart generations fail to encode structured data, and chart-reading questions trigger visual content that does not encode the data.

This also motivates *budget-aware routing*: when predicted utility is low, skipping generation can improve value-per-token without sacrificing accuracy.

Category III: Irrelevant Generation. The generated image is not meaningfully related to the task, indicating complete generation–task misalignment. Here, generation is triggered when it should not be (e.g., factual QA), or the model produces a visual that does not encode the required structured content (e.g., ChartQA).

Common symptoms.

- *Mode confusion:* the model treats a purely textual query as an invitation to draw.
- *Structure loss:* the output resembles an image but lacks the schema needed (bars/axes/labels) to be a chart.
- *Hallucinated content:* visuals introduce unrelated entities, styles, or scenes not grounded in the prompt.

Why it hurts. Irrelevant generation is harmful beyond wasted compute: it can shift the solver's attention to spurious cues, reduce trust in the system, and complicate debugging (the output looks "confident" but is unrelated). In knowledge QA, the correct behavior is often to answer directly; in chart reasoning, the correct behavior is to preserve structured data rather than produce decorative images.

Figure 11 shows irrelevant outputs in chart and factual QA settings: chart generations fail to encode structured data, while factual questions unnecessarily trigger visual generation.

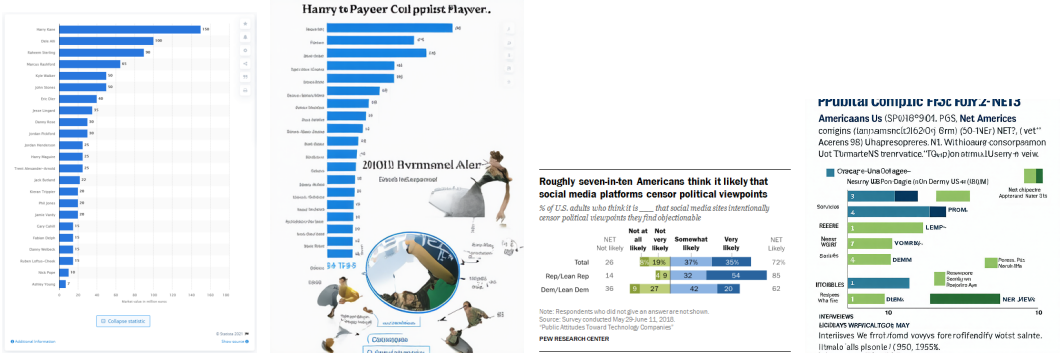
Figure 12 provides additional examples where the model produces visual content that does not help answer the question.

Practical lesson for G2U system design.

1. **Validate** when diagrams are intended to be constraints (Category I risk).
2. **Predict utility** and skip redundant illustrations (Category II risk).
3. **Route by modality** and avoid image generation for pure text tasks (Category III risk).

F Formulations and Definitions

This section formalizes the core objects studied in UniG2U. We (i) define *unified multimodal models* (UMMs) and the capability-level notion of *Generation Helps Understanding* (G2U), (ii) introduce a taxonomy of UMMs



Original (Know, C2)
What is the value of Czechia? One word.

Generated (Know, C2)
UniPic2-Metaquery-9B.

Original (Case 3)

Generated (Case 3)
Bagel

Figure 12: Irrelevant generation in chart-reading and knowledge questions. The model produces visual content that does not encode the information needed for the answer, indicating generation–task misalignment.

used throughout the paper, and (iii) distinguish **Direct** vs. **Generate-then-Answer (GtA)** inference under a unified evaluation.

F.1 Definitions: UMMs and G2U

Task setting. Each instance is formulated as a tuple $x = (V, T, Q)$, where V denotes the input visual context (e.g., images), T represents the text instructions or context, and Q is the specific query to be answered. The target answer is denoted as $y \in \mathcal{Y}$.

Pure understanding models. A *pure understanding* model maps the given input directly to an answer,

$$\hat{y} = M_U(x), \qquad M_U \in \mathcal{M}_U, \tag{5}$$

and does not expose an interface for producing intermediate *visual* artifacts intended to be re-consumed for the same instance.

Unified multimodal models (UMMs). A *unified multimodal model* (M_{UM}) supports multimodal understanding together with a generative interface that can produce an intermediate visual artifact G (e.g., image, sketch, annotation, reconstruction, edit) conditioned on the multimodal input x. Conceptually, UMMs are motivated by the hypothesis that generation can (i) shape internal multimodal representations through unified training, and/or (ii) act as a controllable reasoning resource via these intermediate artifacts.

Defining "Generation Helps Understanding" (G2U). Let $\mathcal{A}(\cdot, \cdot)$ be the task evaluation metric. We define the performance of a model M on a dataset $\mathcal{D}$ as:

$$\mathrm{Perf}(M; \mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathcal{A}(M(x), y)\big]. \tag{6}$$

To reduce confounding from architectural and pretraining differences, for a given unified system M_{UM}, let $\mathcal{B}(M_{UM}) \in \mathcal{M}_U$ denote its corresponding *purely discriminative base VLM* (i.e., the foundational understanding model upon which the unified model is built, evaluated under a matched computation budget).

We define the **G2U gain** for the unified system M_{UM} as:

$$\Delta_{\mathrm{G2U}}(M_{UM}; \mathcal{D}) = \mathrm{Perf}(M_{UM}; \mathcal{D}) - \mathrm{Perf}(\mathcal{B}(M_{UM}); \mathcal{D}). \tag{7}$$

Pairing each unified model with its exact or closest available base VLM provides a diagnostic proxy for the performance shift associated with multimodal unification, while differences in training recipes, instruction tuning, and implementation may remain. The intra-model comparison between Direct and GtA inference (detailed in Sec 5.2) holds model weights and training data fixed and estimates the inference-time effect of explicit visual externalization under our protocol; it does not identify the causal effect of generative training itself.

F.2 Taxonomy of Unified Multimodal Models

We group evaluated UMMs into three categories based on their architectural designs and training paradigms: **end-to-end unified models** (E2E), **decoupled unified systems** (Decoupled), and **agentic unified models** (UM-Ag).

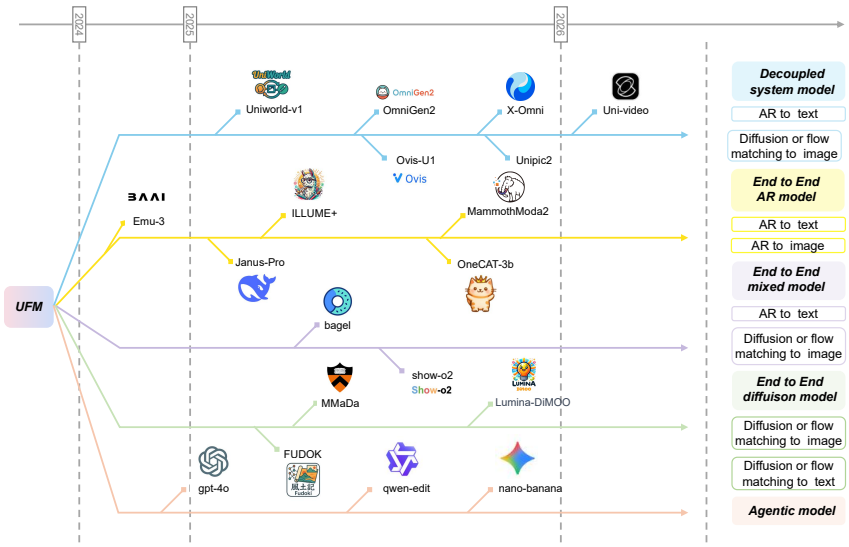


Figure 13: Taxonomy of unified multimodal models (UMMs). The evaluated models and their results are listed in Tables 2 and 13.

All categories can be conceptually expressed via a shared factorization:

$$G = \mathrm{Gen}(x), \qquad \hat{y} = \mathrm{Ans}(x, G), \tag{8}$$

where $G = \varnothing$ recovers direct answering without intermediate visual artifacts. The taxonomy fundamentally differs in how Gen and Ans are implemented and trained.

(1) End-to-end unified models (E2E). E2E models are *parameter-coupled*: Gen and Ans are realized by a single model with shared parameters θ. Crucially, the generation and understanding capabilities are **jointly trained** (e.g., via interleaved multimodal data) within a unified parameter space. A defining property is that both capabilities emerge from the end-to-end training of one unified architecture, rather than from separated components. Representative instances in our study include *Bagel* and *Show-o2*.

(2) Decoupled unified systems (Decoupled). Decoupled systems are *module-separated*. In contrast to E2E models, the generation and understanding components are **not jointly trained**. Instead, Gen and Ans are implemented by independent modules (with parameters ϕ and ψ) that are stitched together through modular orchestration. Concretely, a generation module produces an explicit intermediate artifact G, which is subsequently consumed by a frozen answering module. This modularity enables flexible component-level splicing. Typical examples include *Bagel two-stage* and *OneCAT*.

(3) Agentic unified models (UM-Ag). In agentic systems, generation and understanding are performed by **completely different models**. They achieve unification merely at the protocol level (via a control policy or bounded tool-use interaction) rather than through architectural integration or joint training. For instance, a pure language model might invoke an external, black-box image generation API. In our study, tool-driven visual exploration settings such as *GPT-4o + image tool* instantiate this category.

Conceptually, we regard UM-Ag as a generalized variant of UMMs. However, in our experimental analysis, we explicitly distinguish agentic models from standard unified models (E2E and Decoupled). This separation is necessitated by the fact that most agentic systems consist of distinct, often closed-source proprietary models that rely on external tool-use rather than undergoing any form of multimodal unified training. Consequently, they inherently lack a strictly matched base VLM, rendering them incompatible with the rigorous base-unified comparative framework defined in Section F.1 for calculating exact G2U gains.

Fig. 13 summarizes this taxonomy and highlights how the three classes differ in their degree of unification—ranging from jointly trained parameters (E2E) to stitched modules (Decoupled), and finally to tool-invoked separate models (UM-Ag).

F.3 Direct vs. Generate-then-Answer (GtA) Inference

The Generation-to-Understanding (G2U) paradigm can be instantiated through two distinct inference protocols: Direct and Generate-then-Answer (GtA). In the *Direct* setting, the system answers the query without producing

any external intermediate visual artifacts, formalized as $\hat{y} = \mathrm{Ans}(x, \varnothing)$. In the *GtA* setting, the system explicitly generates an intermediate visual rationale G and re-consumes it to derive the final answer, as defined in (8) (or via an agentic rollout with multiple G_t).

Decomposing G2U gains. Define the best GtA performance for model M as:

$$\mathrm{Perf}_{\mathrm{GtA}}(M;\mathcal{D}) \triangleq \max\Big\{\mathrm{Perf}_{\mathrm{1shot_GtA}}(M;\mathcal{D}), \mathrm{Perf}_{\mathrm{agentic_GtA}}(M;\mathcal{D})\Big\}, \tag{9}$$

where $\mathrm{Perf}_{\mathrm{1shot_GtA}}$ instantiates (8) with a single generated artifact G, and $\mathrm{Perf}_{\mathrm{agentic_GtA}}$ instantiates it with a closed-loop rollout.

We can then decompose the overall G2U performance shift into two specific components:

$$\begin{aligned}\Delta_{\mathrm{Direct}} &= \mathrm{Perf}_{\mathrm{Direct}}(M_{UM};\mathcal{D}) - \mathrm{Perf}(\mathcal{B}(M_{UM});\mathcal{D}),\\ \Delta_{\mathrm{GtA}} &= \mathrm{Perf}_{\mathrm{GtA}}(M_{UM};\mathcal{D}) - \mathrm{Perf}_{\mathrm{Direct}}(M_{UM};\mathcal{D}).\end{aligned} \tag{10}$$

Here, Δ_{Direct} captures the observed capability difference associated with unified multimodal training (without explicit generation), while Δ_{GtA} captures the observed performance difference associated with explicitly externalizing the visual reasoning process prior to answering under the specified inference protocol.

While externalizing visual transformations can be beneficial, the GtA protocol can also degrade performance if the intermediate generated artifacts are inaccurate, physically implausible, or misaligned with the intended reasoning plan, thereby misleading the subsequent answering module.

G Full Theoretical Analysis and Proofs

This appendix gives the complete version of the analysis in Section 3, followed by the proofs. We use the setting and notation of Section 3. Propositions G.1 and G.2 restate Propositions 3.1 and 3.2.

G.1 Complete Analysis

Unified training optimizes a joint likelihood. Understanding alone maximizes $\log p_\theta(y \mid x)$. A unified model is additionally trained to generate visual content, so that for the visual part v of its training data it maximizes the joint likelihood. By the chain rule, $\log p_\theta(y \mid x) = \log p_\theta(x, y) - \log p_\theta(x)$, and the corresponding population objectives decompose as

$$\underbrace{\mathrm{KL}\big(p(x,y) \,\|\, p_\theta(x,y)\big)}_{\text{joint (unified) objective}} = \underbrace{\mathrm{KL}\big(p(x) \,\|\, p_\theta(x)\big)}_{\text{generation}} + \underbrace{\mathbb{E}_x\, \mathrm{KL}\big(p(y \mid x) \,\|\, p_\theta(y \mid x)\big)}_{\text{understanding}}. \tag{11}$$

Neither term is bounded in general; in particular, the understanding term grows without bound when the model assigns vanishing probability to the correct answer. Their relative influence on shared parameters depends on how the losses are weighted and normalized. When the generation loss is summed over visual tokens, its gradient can be large compared with the gradient of a short answer, which is why we analyze the interaction at the level of gradients.

Proposition G.1 (Gradient alignment). *Let $\mathcal{L}_U(\theta) = -\mathbb{E}\log p_\theta(y \mid x)$ and $\mathcal{L}_G(\theta) = -\mathbb{E}\log p_\theta(v)$ with gradients g_U and g_G, and consider one step $\theta^+ = \theta - \eta(g_U + \lambda g_G)$ of unified training versus $\theta^U = \theta - \eta g_U$ of understanding-only training. If $\mathcal{L}_U$ is β-smooth, then*

$$\mathcal{L}_U(\theta^+) - \mathcal{L}_U(\theta^U) = -\eta\lambda\, \|g_U\|\, \|g_G\| \cos(g_U, g_G) + O(\eta^2). \tag{12}$$

To first order, unified training improves understanding more than understanding-only training when the generative gradient is aligned with the understanding gradient and less when the two conflict, with a magnitude proportional to $\lambda\|g_G\|$. The proposition is local and does not determine the sign of the alignment for a task; the following decomposition motivates a hypothesis about it. Writing the visual content as $v = (s, n)$, where s is the structure that determines the answer and n the remaining appearance, gives $\log p_\theta(v) = \log p_\theta(s) + \log p_\theta(n \mid s)$. The first term shares its target with understanding and can contribute aligned gradients, whereas the second term spends capacity on appearance. Tasks whose answers depend on spatial layout, shape, or geometric state have a large structural term, whereas tasks that depend on symbolic content such as text, numbers, or world knowledge have a small one. The hypothesis that alignment is more frequent for the former is consistent with the pattern in Section 5.1; testing it directly requires measuring $\cos(g_U, g_G)$ per task, which we leave to future work.

Generate-then-answer commits to a single intermediate. Under GtA, the model samples $G \sim p_\theta(G \mid x)$ and answers with $p_\theta(y \mid x, G)$. Its own generation and answering distributions imply the marginal $\bar{p}_\theta(y \mid x) = \sum_G p_\theta(G \mid x)\, p_\theta(y \mid x, G)$ and the answer-conditioned posterior $p_\theta(G \mid x, y) \propto p_\theta(G \mid x)\, p_\theta(y \mid x, G)$.

Proposition G.2 (Commitment cost)**.** *For every (x, y),*

$$\mathbb{E}_{G \sim p_\theta(G \mid x)}\big[\log p_\theta(y \mid x, G)\big] = \log \bar{p}_\theta(y \mid x) - \mathrm{KL}\big(p_\theta(G \mid x) \,\|\, p_\theta(G \mid x, y)\big). \tag{13}$$

Consequently, the log-likelihood gain of GtA over Direct inference decomposes as

$$\mathbb{E}_G\big[\log p_\theta(y \mid x, G)\big] - \log q_\theta(y \mid x) = \underbrace{\log \frac{\bar{p}_\theta(y \mid x)}{q_\theta(y \mid x)}}_{\text{externalization gain}} - \underbrace{\mathrm{KL}\big(p_\theta(G \mid x) \,\|\, p_\theta(G \mid x, y)\big)}_{\text{commitment cost} \,\geq\, 0}. \tag{14}$$

The commitment cost is a Jensen gap: it is zero exactly when $p_\theta(y \mid x, G)$ takes the same value for almost all G in the support of $p_\theta(G \mid x)$, and it is large when the probability of the correct answer varies strongly across sampled intermediates. It therefore measures the risk of committing to one sample rather than the correctness of the intermediates; a generator that always outputs the same incorrect image has zero commitment cost, and its failure appears instead as a low marginal $\bar{p}_\theta(y \mid x)$. The identity concerns the log-likelihood of the correct answer; accuracy under greedy decoding is related to it but not determined by it. The externalization gain is positive only when answering from an explicit intermediate is easier for the model than answering directly. Since G is generated from x alone, y and G are conditionally independent given x, so $I(y; x, G) = I(y; x)$ and G adds no information about the answer. Any benefit of GtA is therefore computational. Proposition G.2 suggests that GtA tends to degrade performance when intermediates are uncertain and can help on transformation-intensive tasks such as mazes and sliding puzzles, where each intermediate state is nearly determined by the input so that the commitment cost is small while tracking the state explicitly is easier than doing so implicitly. Replacing generated intermediates with accurate references removes generation errors but leaves the difficulty of answering from an intermediate unchanged, so its benefit should concentrate on tasks with a large externalization gain. We compare these expectations with the results in Section 5.2 but do not estimate the two terms directly.

G.2 Proofs

Chain rule of the joint objective (Equation (2)). For any model $p_\theta(x, y) = p_\theta(x)\, p_\theta(y \mid x)$,

$$\begin{aligned}\mathrm{KL}\big(p(x, y) \,\|\, p_\theta(x, y)\big) &= \mathbb{E}_{p(x,y)}\Big[\log \frac{p(x)}{p_\theta(x)} + \log \frac{p(y \mid x)}{p_\theta(y \mid x)}\Big] \\ &= \mathrm{KL}\big(p(x) \,\|\, p_\theta(x)\big) + \mathbb{E}_{p(x)}\, \mathrm{KL}\big(p(y \mid x) \,\|\, p_\theta(y \mid x)\big).\end{aligned}$$

The decomposition holds for any model of this form and places no bound on either term.

Proof of Proposition 3.1. Assume that $\mathcal{L}_U$ is β-smooth. Let $\theta^U = \theta - \eta g_U$, so that $\theta^+ = \theta^U - \eta\lambda g_G$. By β-smoothness of $\mathcal{L}_U$,

$$\Big|\mathcal{L}_U(\theta^+) - \mathcal{L}_U(\theta^U) + \eta\lambda\, \nabla\mathcal{L}_U(\theta^U)^\top g_G\Big| \leq \tfrac{\beta}{2}\eta^2\lambda^2 \|g_G\|^2,$$

and again by smoothness $\|\nabla\mathcal{L}_U(\theta^U) - g_U\| \leq \beta\eta\|g_U\|$. Hence

$$\mathcal{L}_U(\theta^+) - \mathcal{L}_U(\theta^U) = -\eta\lambda\, g_U^\top g_G + R, \qquad |R| \leq \beta\eta^2\lambda \|g_U\|\|g_G\| + \tfrac{\beta}{2}\eta^2\lambda^2\|g_G\|^2,$$

which is Equation (3) with $g_U^\top g_G = \|g_U\|\|g_G\| \cos(g_U, g_G)$. □

Proof of Proposition 3.2. By Bayes' rule applied to the model's own distributions, for every G with $p_\theta(G \mid x) > 0$,

$$\log p_\theta(y \mid x, G) = \log \bar{p}_\theta(y \mid x) + \log p_\theta(G \mid x, y) - \log p_\theta(G \mid x).$$

Taking the expectation over $G \sim p_\theta(G \mid x)$ gives

$$\mathbb{E}_{G \sim p_\theta(G \mid x)}\big[\log p_\theta(y \mid x, G)\big] = \log \bar{p}_\theta(y \mid x) - \mathbb{E}_{G \sim p_\theta(G \mid x)}\Big[\log \frac{p_\theta(G \mid x)}{p_\theta(G \mid x, y)}\Big],$$

and the last expectation is $\mathrm{KL}(p_\theta(G \mid x) \,\|\, p_\theta(G \mid x, y)) \geq 0$. Subtracting $\log q_\theta(y \mid x)$ from both sides yields Equation (4). □

Conditional independence of generated intermediates. Under GtA, G is sampled from $p_\theta(G \mid x)$ using randomness that is independent of the data. The joint distribution of (x, y, G) is therefore $p(x, y)\, p_\theta(G \mid x)$, so y and G are conditionally independent given x and $I(y; G \mid x) = 0$. By the chain rule of mutual information, $I(y; x, G) = I(y; x) + I(y; G \mid x) = I(y; x)$, and the Bayes-optimal risk of predicting y from (x, G) equals that of predicting y from x.

H Dataset Construction and Filtering Pipeline

UniG2U is designed as a diagnostic benchmark to study when and why generation benefits multimodal understanding. Rather than focusing on a single task family, we construct a diverse dataset covering a wide spectrum of perceptual and cognitive structures. In total, UniG2U contains exactly **3,000 curated samples**, organized into **7 high-level categories** and **30 fine-grained subtasks**. This section details the strict filtering pipeline used to construct the dataset, followed by the taxonomy design and the motivation behind each category.

H.1 Strict Filtering Pipeline

To ensure UniG2U serves as a rigorous, reproducible diagnostic stress-test rather than a subjective or arbitrary collection of tasks, we implemented a systematic, three-stage filtering pipeline to curate the final instances from diverse public datasets.

Stage 1: Structural Inclusion and Exclusion Criteria. We first defined strict filtering rules based on the underlying topology of the reasoning process rather than superficial domain labels.

- *Inclusion Criteria:* Instances must theoretically possess the potential for Generation-to-Understanding (G2U) synergy. The task must necessitate visual externalization, such as spatial state tracking (e.g., sliding puzzles), structural reconstruction, geometric auxiliary constructions, or complex visual disambiguation.
- *Exclusion Criteria:* We strictly excluded pure text-logic QA, shallow perception tasks (e.g., simple OCR or basic object identification), and instances where the answer is explicitly readable from the raw image surface without requiring any intermediate reasoning steps.

The complete instance list with source splits, the filtering scripts, and the evaluation code are open-sourced and included in the supplementary material.

Stage 2: Automated Heuristic Extraction. Based on the defined criteria, we applied automated rule-based filters to the massive source datasets. This involved filtering by question type, removing instances with overly short or purely textual reasoning paths, and utilizing bounding box/element density heuristics to ensure the visual context possessed a requisite baseline of structural complexity.

Stage 3: Expert Verification and Category Balancing. The candidates surviving the automated extraction underwent rigorous human expert review. Annotators verified that an intermediate visual step could plausibly serve as a structural reasoning scaffold for each specific problem. Finally, to prevent the benchmark from being statistically dominated by massive perception-heavy datasets, we enforced strict sampling caps per subtask. This intentional balancing yielded the final, meticulously curated pool of 3,000 instances.

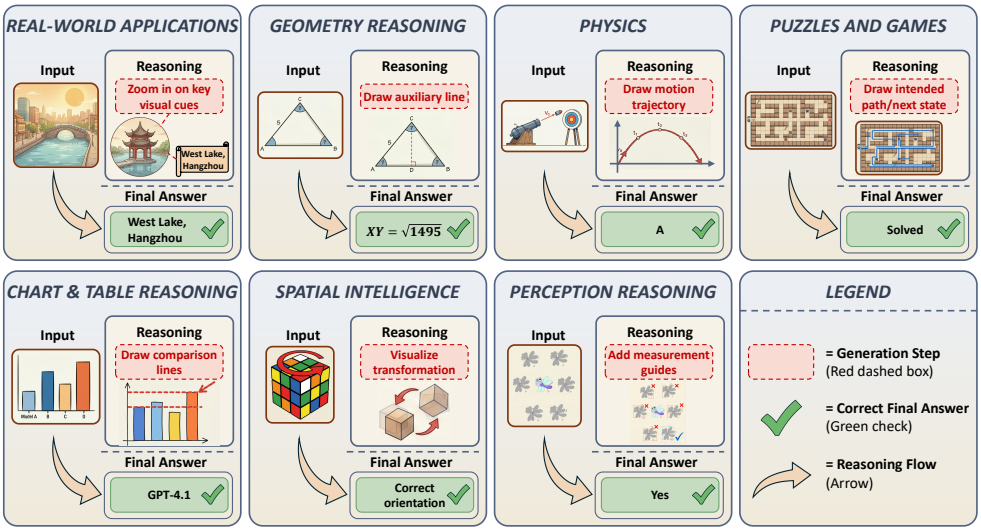


Figure 14: Task taxonomy overview of UniG2U.

H.2 Design Principles and Task Taxonomy

The dataset taxonomy is guided by two core principles, fundamentally echoing Richard Feynman's profound adage: *"What I cannot create, I do not understand."*

First, categories reflect *distinct underlying cognitive structures* rather than arbitrary dataset boundaries. Second, as enforced by our filtering pipeline, all selected categories intentionally curate scenarios where visual externalization is intrinsically coupled with comprehension—whether through drawing geometric auxiliary lines, simulating physical dynamics, tracking spatial transformations, or reconstructing fine-grained perceptual features.

The final composition of the 7 categories is detailed in Table 8 and summarized below:

1. **Real-world Applications (200 samples).** Visually grounded tasks resembling practical applications, such as attentional focusing and visual shortest-path reasoning (VSP). These tasks typically require identifying relevant regions, tracking spatial relations, or following visual constraints across complex scenes. Data are primarily sourced from VSP-style benchmarks and attentional reasoning datasets.
2. **Geometry Reasoning (200 samples).** Visual mathematics and geometry problems, including both 2D and 3D geometric reasoning. Subtasks include planar and solid geometry, curated from datasets such as Geometry3K and AuxSolidMath. These problems often benefit from structured intermediates (e.g., drawing auxiliary lines, reconstructing geometric structure), making them ideal for evaluating explicit generation-augmented understanding.
3. **Physics Reasoning (200 samples).** Known-answer scientific reasoning grounded in diagrams, including physics and optics problems. We include curated subsets from datasets such as Cloudriver and PhyX, where solving the problem requires interpreting diagrams, understanding physical processes, or reasoning over scientific illustrations.
4. **Puzzles and Games (537 samples).** Structured reasoning with state tracking and rule-based transformations, including mental reconstruction/tracking, visual tracking, and classical puzzles such as maze navigation, jigsaw puzzles, and sliding puzzles. Data are drawn from multiple sources including Uni-MMMU, RealUnify, and BabyVision-style datasets. These tasks often admit explicit intermediate visual states, making them suitable for studying both implicit and explicit generation mechanisms.
5. **Chart & Table Reasoning (100 samples).** Reasoning over structured visual data (charts/tables). We use a curated subset of ChartQA, where models must interpret bar charts, line plots, and other visualized representations; intermediate generation may help by re-expressing or simplifying the chart.
6. **Spatial Intelligence (500 samples).** Reasoning about spatial relations, geometric transformations, and object dynamics (e.g., multi-step spatial reasoning, attribute measurement/approximation, camera/object motion). Samples are drawn primarily from spatial reasoning datasets such as MMSI-Bench.
7. **Perception Reasoning (1263 samples).** Fine-grained visual perception and low-level reasoning, including scene/shape-based reasoning, in-scene vs. in-shape discrimination, logo understanding, algorithmic and deductive visual reasoning, spatial pattern recognition, and analogical/inductive reasoning. Data are curated from IllusionBench, visual puzzle datasets, and BabyVision-style benchmarks; compared to other categories, many tasks here are primarily perceptual and may benefit less from explicit generation.

By structuring the benchmark across seven categories with distinct reasoning characteristics, UniG2U enables fine-grained analysis of the relationship between generation and understanding. As illustrated in Fig. 2, the taxonomy spans both generation-friendly categories (e.g., spatial reasoning, puzzles) and more perception-heavy categories, allowing us to probe when intermediate visual generation is helpful, neutral, or harmful. Instead of merely reporting aggregate accuracy, this design allows us to investigate how the effectiveness of generation varies across cognitive categories, providing deeper insight into the mechanisms of unified multimodal models.

I Uni-MMMU Task Design

Uni-MMMU consists of three task types: Jigsaw, Maze, and Sliding Puzzle. Each task evaluates different aspects of visual reasoning through interleaved image-text generation. (following case from Model - Nano Banana)

I.1 Jigsaw Completion Task

I.1.1 Task Description

Input:

Table 8: **Overview of the UniG2U dataset.** The benchmark contains 3,000 samples organized into 7 reasoning categories and 30 fine-grained subtasks, thoughtfully curated from multiple established multimodal reasoning datasets using our strict filtering pipeline.

Category	Subtasks	Data Sources	# Samples
Real-world Applications	Attentional Focusing; Visual Shortest-Path	VSP Li et al. (2025a), RealUnify Shi et al. (2025)	200
Geometry Reasoning	Planar Geometry; Solid Geometry	Geometry3K Lu et al. (2021), AuxSolidMath-Easy Guo et al. (2026)	200
Physics Reasoning	Mechanics; Optics	PhyX Zhang et al. (2025a)	200
Puzzles and Games	Mental Reconstruction; Mental Tracking; Visual Tracking; Maze; Jigsaw; Sliding	Uni-MMMU Zou et al. (2025), RealUnify Shi et al. (2025), BabyVision Chen et al. (2026)	537
Chart & Table Reasoning	Chart Question Answering	ChartQA Masry et al. (2022)	100
Spatial Intelligence	MSR; Attribute (Measurement); Attribute (Approximation); Motion (Camera); Motion (Object)	MMSI-Bench Yang et al. (2025a)	500
Perception Reasoning	Icon-Scene; Icon-Shape; In-scene; In-shape; Logo-Scene; Logo-Shape; Algorithmic; Deductive; Spatial; Analogical; Inductive; Fine-grained Discrimination	IllusionBench Zhang et al. (2025b), Visual Puzzles Wang et al. (2025a), BabyVision Chen et al. (2026)	1263
Total	**30 fine-grained subtasks**	**Multiple sources**	**3,000**

- Reference image: 2Œ2 grid with bottom-right cell hidden
- Candidate 0: First candidate patch
- Candidate 1: Second candidate patch

Task: Select which candidate correctly completes the reference image.

Output: `<FINAL_ANSWER_JSON>{"choice": 0 or 1, "rationale": "..."}`

Metrics: `jigsaw_text_acc` (binary accuracy)

Dataset: 100 samples

I.1.2 Visual CoT Generation Flow

Step 1 - Input Images:

Step 2 - Generate Completion with Candidate 0:

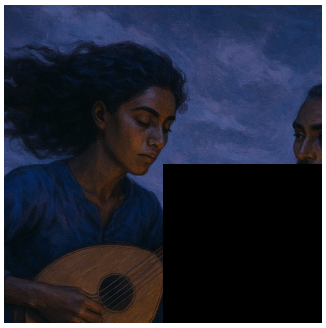
Model generates a 2Œ2 image with Candidate 0 placed in the bottom-right cell. The other three cells remain pixel-identical to the reference.

Step 3 - Generate Completion with Candidate 1:

Model generates a 2Œ2 image with Candidate 1 placed in the bottom-right cell. The other three cells remain pixel-identical to the reference.

Step 4 - Generate Final Answer:

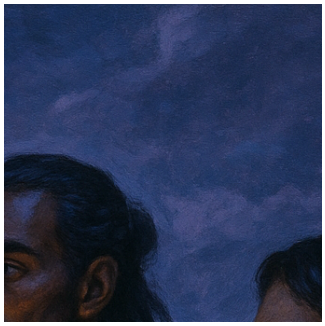
Model analyzes both generated completions and outputs:



(a) Reference (bottom-right hidden)



(b) Candidate 0



(c) Candidate 1

Figure 15: Jigsaw task input images



Figure 16: Generated image: Completion with Candidate 0

```
<FINAL_ANSWER_JSON>
{"choice": 0, "rationale": "Candidate 0 shows better seam continuity"}
</FINAL_ANSWER_JSON>
```

I.2 Maze Solving Task

I.2.1 Task Description

Input: Maze image with:

- Black squares: walls (impassable)
- White squares: paths (walkable)
- Blue dot: starting position
- Green frame: goal region

Task: Find a sequence of moves from start to goal.

Constraints: Legal moves: up, down, left, right; one cell per step; no diagonals; no wall crossing.

Output: `<ANSWER_JSON>["right", "down", ...]`

Metrics: `maze_text_exact`, `maze_text_frame_acc`

Dataset: 100 samples

I.2.2 Visual CoT Generation Flow

Step 1 - Input Image:



Figure 17: Generated image: Completion with Candidate 1

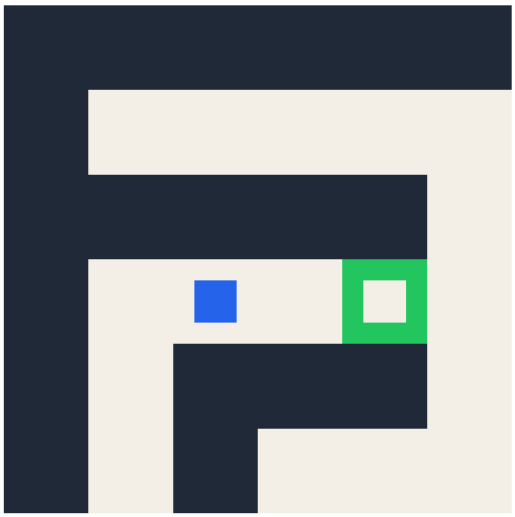


Figure 18: Initial maze state (blue dot at start, green frame marks goal)

Step 2 - Generate Step-by-Step Images:

For each move in the solution, the model generates an image showing the maze state after that move. Only the blue dot position changes; all other elements remain identical.

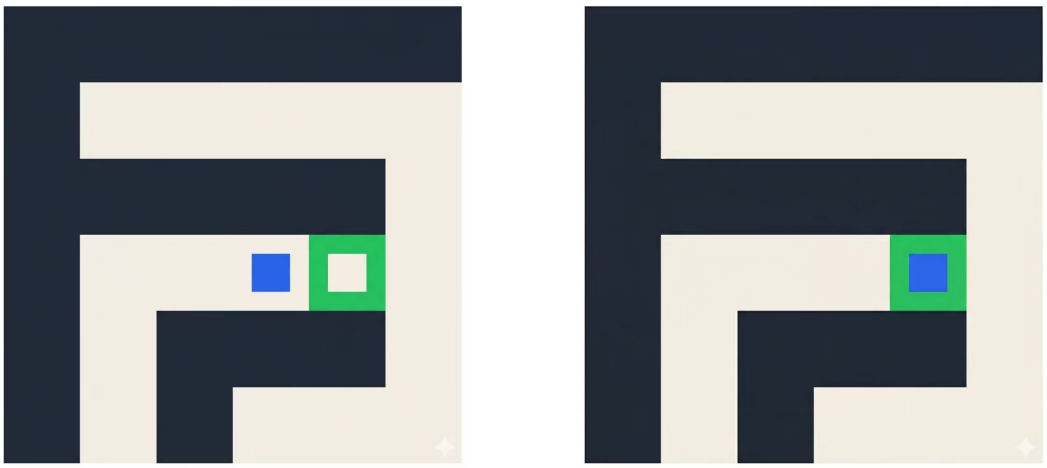
Step 3 - Generate Final Answer:

After generating all step images, the model outputs:

```
<ANSWER_JSON>["right", "down", "right", ...]</ANSWER_JSON>
```

Key Constraints:

- Number of generated images must equal number of moves
- Each image shows state after exactly one move
- Do not include initial state
- No collages, arrows, or annotations



(a) After move 1 (b) After move 2

Figure 19: Example step images (blue dot moves through the maze)

I.3 Sliding Puzzle Task

I.3.1 Task Description

Input:

- Initial state: 3Œ3 grid with 8 colored tiles and 1 empty space (red)
- Goal state: Solved puzzle configuration

Task: Find a sequence of moves to solve the puzzle.

Move Semantics: Moves are named by the direction the colored tile moves (not the empty space). Example: tile above empty moving down = "down".

Constraints: Legal moves: up, down, left, right; one tile per step.

Output: `<ANSWER_JSON>["down", "right", ...]`

Metrics: `sliding_text_exact`, `sliding_text_frame_acc`

Dataset: 54 samples

I.3.2 Visual CoT Generation Flow

Step 1 - Input Images:

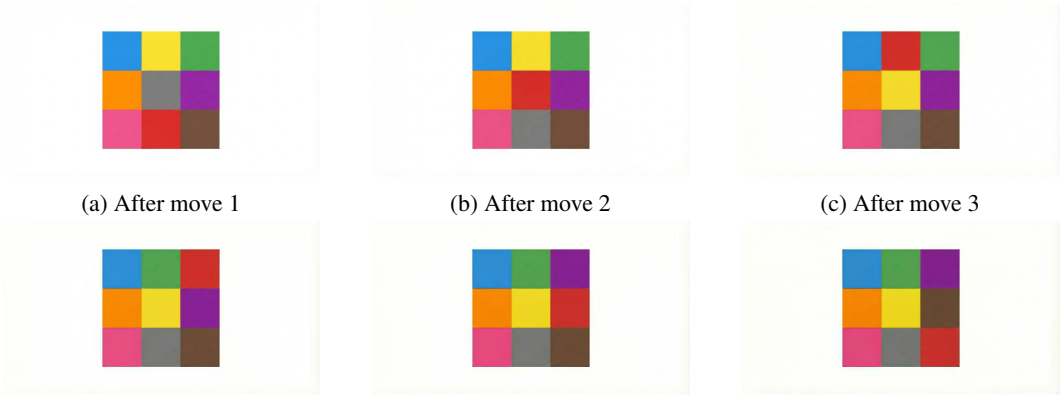


(a) Initial state (scrambled) (b) Goal state (solved)

Figure 20: Sliding puzzle input: initial and goal states

Step 2 - Generate Step-by-Step Images:

For each move in the solution, the model generates an image showing the puzzle state after that move. Only tile positions change; colors remain identical.



(a) After move 1 (b) After move 2 (c) After move 3

(d) After move 4 (e) After move 5 (f) After move 6 (solved)

Figure 21: Example step-by-step puzzle solving sequence

Step 3 - Generate Final Answer:

After generating all step images, the model outputs:

```
<ANSWER_JSON>["down", "right", "up", "left", "down",
  "right"]</ANSWER_JSON>
```

Key Constraints:

- Number of generated images must equal number of moves
- Each image shows state after exactly one move
- Do not include initial state
- No collages or multi-panel images
- Tile colors must remain pixel-identical

J Experimental Results

K Task Prompts (Non-CoT Version)

K.1 MMSI-Bench Tasks

K.1.1 mmsi_attribute_appr (Attribute - Appearance) and mmsi_attribute_meas (Attribute - Measurable) and mmsi_msr (Multi-Step Reasoning)

```
pre_prompt: "Pay close attention to the visual appearance attributes
(color, shape, texture, orientation, count, visual features) in the
images. Observe details carefully to identify and count objects.\n\n"

post_prompt: "\n\nBased on your visual observation, answer with the
option's letter from the given choices directly. Enclose the option's
letter within ``."
```

K.1.2 mmsi_motion_cam (Motion - Camera) and mmsi_motion_obj (Motion - Object)

Note: These two tasks currently share identical prompts in the non-CoT version.

```
pre_prompt: "Analyze the consecutive first-person perspective images.
Determine how the camera (viewpoint) is moving or rotating by observing
changes in the scene and object positions relative to the viewer.\n\n"
```

Table 9: **Main results on UniG2U across categories (Part 1/2).** All values are reported as percentages ($\times 100$). Baseline models (marked with $*$) are highlighted with gray background.

Model	Perception Reasoning												Real-world Apps		Chart
	icon_scene	icon_shape	in_scene	in_shape	logo_scene	logo_shape	algorithmic	analogical	deductive	inductive	spatial	Fine Discrim.	Attn. Focus.	VSP Reason.	ChartQA
qwen2.5-vl-7b*	98	5	98	10	94	20	31	24	46	26	28	11	44	36.5	86
OmniGen2(direct)	89	6	88	8	89	18	33	20	39	18	24	11	50	32.5	86
OmniGen2(GtA)	80	6	81	11	78	23	28	27	39	25	22	9.82	49	33.5	79
UniWorld-V1(GtA)	99	4	99	10	95	19	36	22	47	21	34	6.75	14	34	80
OneCAT-3b(direct)	85	2	84	8	89	13	30	29	38	20	22	10.02	46	33	76.68
OneCAT-3b(GtA)	87	1	82	7	90	4	23	29	35	21	24	9.2	33	34	74
JavisGPT(direct)	86	12	75	13	78	15	36	23	43	22	32	8.32	36	28	78.6
uni-video(GtA)	97	18	96	36	79	50	32	21	37	26	32	9.82	39	29	61
UniPic2(GtA)	94	8	88	24	90	22	32	22	40	23	28	11.66	21	14	47
STAR-7B(GtA)	95	7	98	13	93	19	32	21	46	23	29	6.75	35	33	79
Qwen2.5-VL-3B*	89	6	87	9	90	18	30	22	44	22	21	6.75	51	31	87
UAE(direct)	89	7	88	8	90	18	31	32	34	26	20	10	45	28.5	85
Ovis-U1(direct)	91	7	83	10	87	18	28	27	30	21	27	7.36	59	32	90
Ovis-U1(GtA)	75	5	73	15	48	21	30	22	30	27	30	6.13	23	19	32
yi-6B-vl*	81	17	20	16	84	23	31	18	27	25	33	11.66	27	21	19
mio(direct)	46	5	36	13	30	27	32	22	35	28	24	14.72	35	1	15
mio(GtA)	53	15	36	16	28	19	21	23	32	28	27	10.43	30	0	13
deepseek-v2*	60	8	78	13	90	19	40	23	35	22	34	12.88	22	0	74
Janus-Pro(direct)	82	4	85	21	74	32	26	21	30	25	27	9.82	27	28	29
AIA(direct)	85	5	87	18	81	28	31	22	24	31	29	11.04	25	2	33
LLaDA-Instruct*	7	0	9	8	7	5	29	24	32	24	24	3.07	32	22.5	9
MMaDA(direct)	55	1	47	3	44	4	25	29	31	18	22	5	32	25.5	14
Janus-1.3B*	57	6	35	10	35	23	21	23	25	22	22	11.04	26	1	50
FUDOK(direct)	10	10	10	5	11	4	25	24	32	28	24	8	34	1	22
Qwen3-VL-8B*	100	13	100	22	97	28	45	29	40	21	32	15	48	32	89
MammothModa2(direct)	98	5	98	10	94	20	37	26	48	20	30	4.91	45	22	91
llava-hf*	84	10	81	20	59	31	23	30	29	20	22	9.2	27	19	28
Emu3(direct)	51	0	60	2	35	5	33	25	34	24	28	9.82	31	2	67
llava_onevision*	86	6	74	18	91	34	35	25	28	32	22	11.04	54	32	78
X-Omni(direct)	91	5	97	11	85	13	27	24	35	17	27	8.59	42	24	83
bagel(direct)	91	2	88	18	91	20	31	26	46	42	35	9	50	26	92
bagel(GtA)	81	2	81	17	80	20	30	26	43	40	35	9.82	46	19.5	91
show-o2(direct)	89	11	80	22	81	37	22	24	36	21	26	7.36	38	33	45
show-o2(GtA)	56	6	51	18	53	31	21	23	32	23	25	9.82	35	31	35
ILLUME+(direct)	86	4	84	20	91	26	22	28	30	18	23	6.75	36	13	80
ILLUME+(GtA)	78	14	83	16	87	21	18	27	37	17	25	4.29	21	20	58
gpt4o	96	16	93	28	100	33	37	32	42	27	32	10	53	37.5	58
ovis2.5	99	8	92	21	97	27	34	34	49	25	32	9.82	57	31	89
GPT-4o + GPT-image	90	17	89	25	92	30	35	27	32	30	19	15.95	47	18.5	60
gemini pro	100	38	100	43	100	58	79	73	50	31	62	23.31	64	79	68
qwenedit	99	5	98	11	89	25	24	21	45	27	31	7.36	42	25.5	80
SenseNova-U1.5-8B-MoT (Direct)	96	10	96	15	93	12	30	21	39	27	35	10	57	28	85
SenseNova-U1.5-8B-MoT (Visual CoT)	95	8	91	26	90	21	33	26	35	27	34	15	51	28	82

Table 10: **Main results on UniG2U across categories (Part 2/2).** Continuing from Part 1 with the remaining columns.

Model	Geometry Reasoning		Spatial Intel.					Puzzles & Games						Physics	
	Geometry	3d Geometry	MSR	Attr. (Meas.)	Attr. (Appr.)	Motion (Cam.)	Motion (Obj.)	Mental Recon.	Mental Track.	Visual Track.	maze	jigsaw	sliding	light	mechanism
qwen2.5-vl-7b*	46	0	25	20	10	29	33	36	25	9.64	14.12	52	5.61	34	34
OmniGen2(direct)	28	0	22	33	19	12	33	31	23	7.23	4.31	48	0	42	40
OmniGen2(GtA)	26	0	29	33	17	18	35	31	23	6.02	4.6	54	0.13	41	42
UniWorld-V1(GtA)	38	0	27	22	16	21	38	39	24	6.02	14.43	47	6.14	33	36
OneCAT-3b(direct)	34	0	32	30	27	24	13	24	25	6.02	8.4	50	11.18	41	23
OneCAT-3b(GtA)	17	0	39	28	23	3	26	23	24	6.02	7.43	50	3.1	37	21
JavisGPT(direct)	42	0	16	24	18	19	31	30	33	8.74	0.48	51	10.44	25	28
uni-video(GtA)	43	0	23	18	17	10	32	28	26	9.64	21.9	57	30.79	31	27
UniPic2(GtA)	33	1	21	23	21	23	35	33	22	4.82	11.53	48	18.71	28	34
STAR-7B(GtA)	29	0	27	22	17	21	39	31	27	6.02	6.87	46	19.72	34	31
Qwen2.5-VL-3B*	31	0	22	32	24	18	36	31	24	3.61	1.64	50	0	42	39
UAE(direct)	21	0	26	22	26	20	32	33	19	4.82	5.17	38	5.91	35	26
Ovis-U1(direct)	47	0	12	28	7	18	23	31	28	8.43	13.44	49	33.39	35	26
Ovis-U1(GtA)	33	0	27	27	15	11	22	22	19	3.61	0.25	52	0.74	19	15
yi-6B-vl*	0	0	24	26	22	11	30	33	22	6.02	0	11	0	35	32
mio(direct)	0	0	21	24	24	23	25	25	20	3.61	0	21	0	30	32
mio(GtA)	0	0	32	24	14	22	16	27	17	3.61	0	1	0	27	29
deepseek-v2*	9	0	23	19	9	14	22	11	1	6.02	11.2	53	28.82	21	23
Janus-Pro(direct)	15	0	28	37	25	16	15	21	26	3.61	12.53	54	7.87	21	17
AIA(direct)	17	0	23	43	21	19	27	31	19	4.82	12.87	51	19.93	31	31
LLaDA-Instruct*	3	0	21	41	18	11	26	22	26	3.61	12.87	48	0	24	18
MMaDA(direct)	10	0	32	27	25	20	21	21	28	2.41	0	35	0	28	25
Janus-1.3B*	5	0	15	28	21	18	27	24	19	2.41	1.81	18	0.26	19	39
FUDOK(direct)	2	0	29	28	36	29	23	22	25	3.61	0	0	0	25	17
Qwen3-VL-8B*	67	11	23	35	22	20	30	32	26	8.43	15.53	45	8.59	37	37
MammothModa2(direct)	0	0	14	19	18	18	40	35	24	7.23	0	0	0	40	33
llava-hf*	0	0	28	24	30	14	27	26	20	4.82	0	50	3.35	21	38
Emu3(direct)	2	0	31	23	28	14	20	33	27	4.82	1.39	0	11.85	25	16
llava_onevision*	21	0	17	26	29	20	35	31	23	16.87	7.46	51	9.25	42	46
X-Omni(direct)	19	0	20	31	26	24	34	30	27	6.02	12.87	60	11.23	35	24
bagel(direct)	58	0	26	31	20	26	39	37	30	7.23	2.15	52	0.9	35	40
bagel(GtA)	58	2	25	36	27	24	33	41	31	4.82	28.1	55	19.45	38	43
show-o2(direct)	22	0	27	24	21	28	29	32	22	9.64	14.24	56	13.74	37	43
show-o2(GtA)	18	0	30	15	0	23	34	30	23	10.84	17.84	52	18.32	43	37
ILLUME+(direct)	15	0	28	29	14	14	31	27	19	6.02	6.93	53	4.02	38	32
ILLUME+(GtA)	4	0	29	28	15	16	17	20	18	2.41	12.87	52	0	37	34
gpt4o	53	1	33	38	32	34	43	33	24	14.46	32.15	52	21.99	36	33
ovis2.5	64	1	27	40	21	22	29	25	23	8.43	15.07	45	16.93	42	44
GPT-4o + GPT-image	47	7	23	30	23	23	36	36	20	16.87	39.84	50	29.4	37	24
gemini pro	86	84	35	60	29	28	40	39	70	26.51	49.29	77	73.57	89	93
qwenedit	34	0	19	19	15	18	37	32	23	7.23	25.57	52	19.7	33	30
SenseNova-U1.5-8B-MoT (Direct)	55	34	32	34	18	27	35	26	25	10.84	0	62	0	27	38
SenseNova-U1.5-8B-MoT (Visual CoT)	65	26	30	37	18	23	42	29	28	12.05	4.33	62	0	36	32

Table 11: **Delta results on UniG2U across categories (Sheet 3, Part 1/2).** Baseline models (marked with $*$) are highlighted with gray background, and missing values are denoted by '-'.

Model	Perception Reasoning												Real-world Apps		Chart
	icon_scene	icon_shape	in_scene	in_shape	logo_scene	logo_shape	algorithmic	analogical	deductive	inductive	spatial	Fine Discrim.	Attn. Focus.	VSP Reason.	ChartQA
qwen2.5-vl-7b*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OmniGen2(direct)	-9	1	-10	-2	-5	-2	2	-4	-7	-8	-4	0	6	-4	0
OmniGen2(GtA)	-18	1	-17	1	-16	3	-3	3	-7	-1	-6	-1.18	5	-3	-7
UniWorld-V1(GtA)	1	-1	1	0	1	-1	5	-2	1	-5	6	-4.25	-30	-2.5	-6
OneCAT-3b(direct)	-13	-3	-14	-2	-5	-7	-1	5	-8	-6	-6	-0.98	2	-3.5	-9.32
OneCAT-3b(GtA)	-11	-4	-16	-3	-4	-16	-8	5	-11	-5	-4	-1.8	-11	-2.5	-12
JavisGPT(direct)	-12	7	-23	3	-16	-5	5	-1	-3	-4	4	-2.68	-8	-8.5	-7.4
uni-video(GtA)	-1	13	-2	26	-15	30	1	-3	-9	0	4	-1.18	-5	-7.5	-25
UniPic2(GtA)	-4	3	-10	14	-4	2	1	-2	-6	-3	0	0.66	-23	-22.5	-39
STAR-7B(GtA)	-3	2	0	3	-1	-1	1	-3	0	-3	1	-4.25	-9	-3.5	-7
Qwen2.5-VL-3B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UAE(direct)	0	1	1	-1	0	0	1	10	-10	4	-1	3.25	-6	-2.5	-2
Ovis-U1(direct)	2	1	-4	1	-3	0	-2	5	-14	-1	6	0.61	8	1	3
Ovis-U1(GtA)	-14	-1	-14	6	-42	3	0	0	-14	5	9	-0.62	-28	-12	-55
yi-6B-vl*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
mio(direct)	-35	-12	16	-3	-54	4	1	4	8	3	-9	3.06	8	-20	-4
mio(GtA)	-28	-2	16	0	-56	-4	-10	5	5	3	-6	-1.23	3	-21	-6
deepseek-v2*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Janus-Pro(direct)	22	-4	7	8	-16	13	-14	-2	-5	3	-7	-3.06	5	28	-45
AIA(direct)	25	-3	9	5	-9	9	-9	-1	-11	9	-5	-1.84	3	2	-41
LLaDA-Instruct*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MMaDA(direct)	48	1	38	-5	37	-1	-4	5	-1	-6	-2	1.93	0	3	5
Janus-1.3B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FUDOK(direct)	-47	4	-25	-5	-24	-19	4	1	7	6	2	-3.04	8	0	-28
Qwen3-VL-8B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MammothModa2(direct)	-2	-8	-2	-12	-3	-8	-8	-3	8	-1	-2	-10.09	-3	-10	2
llava-hf*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Emu3(direct)	-33	-10	-21	-18	-24	-26	10	-5	5	4	6	0.62	4	-17	39
llava_onevision*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
X-Omni(direct)	5	-1	23	-7	-6	-21	-8	-1	7	-15	5	-2.45	-12	-8	5
bagel(direct)	5	-4	14	0	0	-14	-4	1	18	10	13	-2.04	-4	-6	14
bagel(GtA)	-5	-4	7	-1	-11	-14	-5	1	15	8	13	-1.22	-8	-12.5	13
show-o2(direct)	3	5	6	4	-10	3	-13	-1	8	-11	4	-3.68	-16	1	-33
show-o2(GtA)	-30	0	-23	0	-38	-3	-14	-2	4	-9	3	-1.22	-19	-1	-43
ILLUME+(direct)	0	-2	10	2	0	-8	-13	3	2	-14	1	-4.29	-18	-19	2
ILLUME+(GtA)	-8	8	9	-2	-4	-13	-17	2	9	-15	3	-6.75	-33	-12	-20
gpt4o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gpt4o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GPT-4o + GPT-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gemini pro	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
qwenedit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SenseNova-U1.5-8B-MoT (Direct, unpaired)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SenseNova-U1.5-8B-MoT (Visual CoT, unpaired)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 12: **Delta results on UniG2U across categories (Sheet 3, Part 2/2).** Continuing from Part 1 with the remaining columns.

Model	Geometry Reasoning		Spatial Intel.					Puzzles & Games							Physics	
	Geometry	3d Geometry	MSR	Attr. (Meas.)	Attr. (Appr.)	Motion (Cam.)	Motion (Obj.)	Mental Recon.	Mental Track.	Visual Track.	maze	jigsaw	sliding	light	mechanism
qwen2.5-vl-7b*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OmniGen2(direct)	-18	0	-3	13	9	-17	0	-5	-2	-2.41	-9.81	-4	-5.61	8	6
OmniGen2(GtA)	-20	0	4	13	7	-11	2	-5	-2	-3.62	-9.52	2	-5.48	7	8
UniWorld-V1(GtA)	-8	0	2	2	6	-8	5	3	-1	-3.62	0.31	-5	0.53	-1	2
OneCAT-3b(direct)	-12	0	7	10	17	-5	-20	-12	0	-3.62	-5.72	-2	5.57	7	-11
OneCAT-3b(GtA)	-29	0	14	8	13	-26	-7	-13	-1	-3.62	-6.69	-2	-2.51	3	-13
JavisGPT(direct)	-4	0	-9	4	8	-10	-2	-6	8	-0.9	-13.64	-1	4.83	-9	-6
uni-video(GtA)	-3	0	-2	-2	7	-19	-1	-8	1	0	7.78	5	25.18	-3	-7
UniPic2(GtA)	-13	1	-4	3	11	-6	2	-3	-3	-4.82	-2.59	-4	13.1	-6	0
STAR-7B(GtA)	-17	0	2	2	7	-8	6	-5	2	-3.62	-7.25	-6	14.11	0	-3
Qwen2.5-VL-3B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UAE(direct)	-10	0	4	-10	2	2	-4	2	-5	1.21	3.53	-12	5.91	-7	-13
Ovis-U1(direct)	16	0	-10	-4	-17	0	-13	0	4	4.82	11.8	-1	33.39	-7	-13
Ovis-U1(GtA)	2	0	5	-5	-9	-7	-14	-9	-5	0	-1.39	2	0.74	-23	-24
yi-6B-vl*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
mio(direct)	0	0	-3	-2	2	12	-5	-8	-2	-2.41	0	10	0	-5	0
mio(GtA)	0	0	8	-2	-8	11	-14	-6	-5	-2.41	0	-10	0	-8	-3
deepseek-v2*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Janus-Pro(direct)	6	0	5	18	16	2	-7	10	25	-2.41	1.33	1	-20.95	0	-6
AIA(direct)	8	0	0	24	12	5	5	20	18	-1.2	1.67	-2	-8.89	10	8
LLaDA-Instruct*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MMaDA(direct)	7	0	11	-14	7	9	-5	-1	2	-1.2	-12.87	-13	0	4	7
Janus-1.3B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FUDOK(direct)	-3	0	14	0	15	11	-4	-2	6	1.2	-1.81	-18	-0.26	6	-22
Qwen3-VL-8B*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MammothModa2(direct)	-67	-11	-9	-16	-4	-2	10	3	-2	-1.2	-15.53	-45	-8.59	3	-4
llava-hf*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Emu3(direct)	2	0	3	-1	-2	0	-7	7	7	0	1.39	-50	8.5	4	-22
llava_onevision*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
X-Omni(direct)	-2	0	3	5	-3	4	-1	-1	4	-10.85	5.41	9	1.98	-7	-22
bagel(direct)	37	0	9	5	-9	6	4	6	7	-9.64	-5.31	1	-8.35	-7	-6
bagel(GtA)	37	2	8	10	-2	4	-2	10	8	-12.05	20.64	4	10.2	-4	-3
show-o2(direct)	1	0	10	-2	-8	8	-6	1	-1	-7.23	6.78	5	4.49	-5	-3
show-o2(GtA)	-3	0	13	-11	-29	3	-1	-1	0	-6.03	10.38	1	9.07	1	-9
ILLUME+(direct)	-6	0	11	3	-15	-6	-4	-4	-4	-10.85	-0.53	2	-5.23	-4	-14
ILLUME+(GtA)	-17	0	12	2	-14	-4	-18	-11	-5	-14.46	5.41	1	-9.25	-5	-12
gpt4o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ovis2.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GPT-4o + GPT-image	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gemini pro	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
qwenedit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SenseNova-U1.5-8B-MoT (Direct, unpaired)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SenseNova-U1.5-8B-MoT (Visual CoT, unpaired)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

```
post_prompt: "\n\nBased on your analysis of camera motion, answer with
the option's letter from the given choices directly. Enclose the
option's letter within ``."
```

K.2 IllusionBench Tasks

K.2.1 ICON Shape Recognition

```
shape_string = "Animal, Face_Emoji, Music, Sport, Stationery, Vehicle"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
Time_square"

prompt = f"This image contains a icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the icon that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the icon. You have to respond with a single word."
```

K.2.2 ICON Scene Recognition, LOGO Scene Recognition, and IN Scene Recognition

Note: These three tasks share identical prompts (only differ in shape_string values).

```
prompt = f"This image contains an icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the background that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}.
Provide your response by stating only the single, most accurate class
name that represents the background. You have to respond with a single
word."
```

K.2.3 LOGO Shape Recognition

```
shape_string = "Adidas, Amazon, Apple, Audi, BMW, Mercedes Benz,
Facebook, Google, Instagram, Mcdonalds, Nasa, Nike, Olympics,
Playstation, Puma, Reebok, Spotify, Starbucks, Tesla, Telegram, Ubuntu"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
Time_square"

prompt = f"This image contains a icon integrated into a background,
where elements of the background contribute to forming the logo.
Identify the logo that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the logo. You have to respond with a single word."
```

K.2.4 IN Shape Recognition

```
shape_string = "Airplane, Bicycle, Bird, Bottle, Car, Cat, Dog,
Dolphin, Fork, Guitar, Mug, Panda, Paper_clip, Sailboat, Scooter,
Teapot"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
Time_square"

prompt = f"This image contains a icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the shape that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the icon. You have to respond with a single word."
```

K.3 VISUALPUZZLES TASKS

K.3.1 ALL CATEGORIES (ALGORITHMIC, ANALOGICAL, DEDUCTIVE, INDUCTIVE, SPATIAL)

```
MULTI_CHOICE_DIRECT_PROMPT = "Answer the question with the option's
letter from the given choices directly."
```

Full prompt format:

```
question = "Question: " + doc["question"].strip()
options = doc["options"]
if options is not None:
    question += "\nOptions:\n(A) " + options[0] + "\n(B) " + options[1]
    + "\n(C) " + options[2] + "\n(D) " + options[3]
else:
    question += "\nOptions: Choose from (A) (B) (C) (D) in the image."

question += "\n" + MULTI_CHOICE_DIRECT_PROMPT
```

K.4 UNI-MMMU TASKS

ů

K.4.1 UNI_MMMU_JIGSAW100_VISUAL_COT

Full Prompt:

```
You are a unified vision-language model. You will be given:
(1) a 2x2 reference image with the bottom-right cell hidden, and
(2) two candidate patch images ("Candidate 0" and "Candidate 1").

Your job:
- For each candidate, synthesize a completed 2x2 image by placing that
  candidate EXACTLY into the bottom-right cell. Keep the other three
  cells pixel-identical to the reference (no filtering, no
  re-rendering). If sizes differ, only scale the candidate to fit that
  quadrant; do NOT rotate, mirror, or alter colors.
- Compare the two completed results and decide which candidate yields
  the correct completion.

Output EXACTLY the following, in order:

1) A single image with Candidate 0 placed in the bottom-right cell

2) A single image with Candidate 1 placed in the bottom-right cell


3) analysis comparing seam continuity, color/texture gradient,
   structural alignment, and global semantics

4) One strict JSON object with your decision, wrapped as:
<FINAL_ANSWER_JSON>
{"choice": 0 or 1, "rationale": "<=30 words decisive cue"}
</FINAL_ANSWER_JSON>

Hard constraints:
- Deterministic, single outcome. No hedging, no multiple possibilities.
- No meta talk about prompts, models, or pipelines.
- Do not restate the task as the answer; reason from visual evidence.
- The only edits allowed are pasting the candidate into the bottom-
  right cell and necessary size matching for that cell. All other
  pixels must remain identical to the reference.

Inputs:
```

K.4.2 UNI_MMMU_MAZE100_VISUAL_COT

Full Prompt:

```
You are a precise maze solver.

SEMANTICS (for all mazes)
- Black squares: walls (impassable)
- White squares: path (walkable)
- Blue dot: start (the agent)
- Green rectangular frame: goal (reaching any white cell inside the
  green frame counts as success)
- Legal moves: up, down, left, right only. One cell per step; no
  diagonals, no jumps; never cross walls.

OUTPUT FORMAT (STRICT)
1) MULTI-IMAGE MODE - generate a SEQUENCE OF SEPARATE IMAGES, one per
   move:
   - Each output image must depict the maze state AFTER applying
     exactly one legal move.
   - Do NOT include the initial (pre-move) state.
   - Keep palette/layout/scale identical to the input; only the blue
     dot moves.
   - The number of returned images MUST equal the number of moves in
     the final answer (see step 2).
   - Absolutely FORBIDDEN: any collage/montage/spritesheet/grid/
     multi-panel/side-by-side/stacked images; no arrows, captions, or
     overlays; no GIFs/animations/video.

2) After all step images, emit EXACTLY ONE LINE containing ONLY the
   final move list as a JSON array of lowercase strings, wrapped as:
   <ANSWER_JSON>["right","down","left"]</ANSWER_JSON>


NO EXTRAS
- No tools, no OCR, no explanations, and no text other than the single
  <ANSWER_JSON>.....</ANSWER_JSON> line.
- Do not restate the instructions or the condition.

REMINDERS
- Decide the full path first, then emit the image sequence (one image
  per move), then the single <ANSWER_JSON> line.
- One move per image; images must be separate files/parts, not stitched
  together in any way.
```

K.4.3 UNI_MMMU_SLIDING54_VISUAL_COT

Full Prompt:

```
You are a precise sliding puzzle solver.

SEMANTICS
- The puzzle is a 3x3 grid with 8 colored tiles and one empty space
  (shown as red).
- A "move" slides an adjacent tile INTO the empty space.
- Moves are named by the direction the COLORED TILE moves (not the
  empty space).
- Legal moves: up, down, left, right only. One tile per step.

OUTPUT FORMAT (STRICT)
1) MULTI-IMAGE MODE - generate a SEQUENCE OF SEPARATE IMAGES, one per
   move:
   - Each output image must depict the puzzle state AFTER applying
     exactly one move.
   - Do NOT include the initial (pre-move) state.
```

```
   - Keep tile colors identical; only positions change.
   - The number of returned images MUST equal the number of moves in
     the final answer.
   - Absolutely FORBIDDEN: any collage/montage/spritesheet/grid/
     multi-panel/side-by-side/stacked images.

2) After all step images, emit EXACTLY ONE LINE containing ONLY the
   final move list as a JSON array of lowercase strings, wrapped as:
   <ANSWER_JSON>["down","right","up"]</ANSWER_JSON>

NO EXTRAS
- No explanations beyond the single <ANSWER_JSON>.....</ANSWER_JSON>
  line.
- Do not restate the instructions.
```

L TASK PROMPTS (VISUAL COT VERSION)

This appendix contains all the original English prompts for Visual CoT tasks.

All Visual CoT tasks use a two-stage format:

```
[GEN_PROMPT]...[/GEN_PROMPT][QUESTION]...[/QUESTION]
```

L.1 MMSI-BENCH TASKS

L.1.1 MMSI_ATTRIBUTE_APPR_VISUAL_COT

Stage 1 - Generation Prompt:

```
Create a visualization that highlights and labels the visual appearance
attributes (color, shape, texture, orientation, count) in the scene.
Use annotations, bounding boxes, and labels to make object features and
counts clearly visible.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given the original image(s) and a visualization
highlighting appearance attributes. Use both to analyze color, shape,
texture, orientation, and object counts.\n\n"

post_prompt: "\n\nBased on your visual observation, answer with the
option's letter from the given choices directly. Enclose the option's
letter within ``."
```

L.1.2 MMSI_ATTRIBUTE_MEAS_VISUAL_COT

Stage 1 - Generation Prompt:

```
Create a visualization that highlights and annotates the measurable
attributes (size, length, width, height, distance, area, volume) in
the scene. Draw measurement lines, labels, and comparison markers to
make quantitative relationships clear.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given the original image(s) and a visualization
highlighting measurable attributes. Use both to analyze size, length,
width, height, distance, area, and volume relationships.\n\n"

post_prompt: "\n\nBased on your measurement analysis, answer with the
option's letter from the given choices directly. Enclose the option's
letter within ``."
```

L.1.3 MMSI_MOTION_CAM_VISUAL_COT

Stage 1 - Generation Prompt:

```
Create a visualization showing camera motion analysis. Draw arrows
indicating the direction of camera movement/rotation, highlight
reference points that shift between frames, and annotate the type of
camera motion (pan, tilt, zoom, dolly, etc.).
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given consecutive first-person perspective images
and a visualization of camera motion. Use both to determine how the
camera (viewpoint) is moving or rotating.\n\n"

post_prompt: "\n\nBased on your analysis of camera motion, answer with
the option's letter from the given choices directly. Enclose the
option's letter within ``."
```

L.1.4 MMSI_MOTION_OBJ_VISUAL_COT

Stage 1 - Generation Prompt:

```
Create a visualization showing object motion tracking. Draw motion
trails, arrows indicating direction of movement, and highlight moving
objects with bounding boxes. Annotate relative speeds and trajectories
of different objects.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given consecutive images and a visualization of
object motion. Use both to track objects, identify which are moving,
their direction, and relative speeds.\n\n"

post_prompt: "\n\nBased on your analysis of object motion, answer with
the option's letter from the given choices directly. Enclose the
option's letter within ``."
```

L.1.5 MMSI_MSR_VISUAL_COT

Stage 1 - Generation Prompt:

```
Create a visualization that breaks down the multi-step reasoning
process. Annotate spatial relationships, object positions, and key
changes across images. Draw diagrams showing logical connections and
intermediate reasoning steps.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given the original images and a visualization
breaking down the reasoning steps. Use both to analyze spatial
relationships, object positions, and changes across images.\n\n"

post_prompt: "\n\nBased on your multi-step reasoning, answer with the
option's letter from the given choices directly. Enclose the option's
letter within ``."
```

L.2 ILLUSIONBENCH TASKS

L.2.1 ICON SHAPE RECOGNITION VISUAL COT

Stage 1 - Generation Prompt:

```
Create a visualization that highlights how the background elements
contribute to forming the icon shape. Draw outlines, arrows, and
annotations showing which parts of the scene create the icon's
silhouette or features.
```

Stage 2 - Question Prompt:

```
shape_string = "Animal, Face_Emoji, Music, Sport, Stationery, Vehicle"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
Time_square"

pre_prompt: "You are given the original image and a visualization
highlighting how background elements form the icon. Use both to
identify the icon shape.\n\n"

prompt = f"This image contains an icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the icon that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the icon. You have to respond with a single word."
```

L.2.2 ICON SCENE RECOGNITION VISUAL COT, LOGO SCENE RECOGNITION VISUAL COT, AND IN SCENE RECOGNITION VISUAL COT

Note: These three tasks share identical prompts (only differ in shape_string values).

Stage 1 - Generation Prompt:

```
Create a visualization that highlights the background scene elements.
Draw annotations identifying key features of the environment, landscape,
or setting that characterize the scene type.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given the original image and a visualization
highlighting the background scene elements. Use both to identify the
scene type.\n\n"

prompt = f"This image contains an icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the background that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}.
Provide your response by stating only the single, most accurate class
name that represents the background. You have to respond with a single
word."
```

L.2.3 LOGO SHAPE RECOGNITION VISUAL COT

Stage 1 - Generation Prompt:

```
Create a visualization that highlights how the background elements
contribute to forming the logo shape. Draw outlines, arrows, and
annotations showing which parts of the scene create the logo's
silhouette or features.
```

Stage 2 - Question Prompt:

```
shape_string = "Adidas, Amazon, Apple, Audi, BMW, Mercedes Benz,
Facebook, Google, Instagram, Mcdonalds, Nasa, Nike, Olympics,
Playstation, Puma, Reebok, Spotify, Starbucks, Tesla, Telegram, Ubuntu"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
```

```
Time_square"

pre_prompt: "You are given the original image and a visualization
highlighting how background elements form the logo. Use both to
identify the logo.\n\n"

prompt = f"This image contains an icon integrated into a background,
where elements of the background contribute to forming the logo.
Identify the logo that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the logo. You have to respond with a single word."
```

L.2.4 IN SHAPE RECOGNITION VISUAL COT

Stage 1 - Generation Prompt:

```
Create a visualization that highlights how the background elements
contribute to forming the shape. Draw outlines, arrows, and annotations
showing which parts of the scene create the shape's silhouette or
features.
```

Stage 2 - Question Prompt:

```
shape_string = "Airplane, Bicycle, Bird, Bottle, Car, Cat, Dog,
Dolphin, Fork, Guitar, Mug, Panda, Paper_clip, Sailboat, Scooter,
Teapot"
scene_string = "Ocean, Origami, Forest, Cloud, Sand_dune,
Medieval_Village, City, Underwater_ruins, Museum, Bazaar_market,
Time_square"

pre_prompt: "You are given the original image and a visualization
highlighting how background elements form the shape. Use both to
identify the shape.\n\n"

prompt = f"This image contains an icon integrated into a background,
where elements of the background contribute to forming the icon.
Identify the shape that is represented in the image by choosing
exclusively among the following options:{shape_string},{scene_string}
Provide your response by stating only the single, most accurate class
name that represents the icon. You have to respond with a single word."
```

L.3 VISUALPUZZLES TASKS

L.3.1 ALGORITHMIC VISUAL COT

Stage 1 - Generation Prompt:

```
Create a visualization that breaks down the algorithmic pattern or
sequence. Draw step-by-step diagrams, number sequences, or flowcharts
showing the logical progression and rules governing the pattern.
```

Stage 2 - Question Prompt:

```
pre_prompt: "You are given the original puzzle image(s) and a
visualization breaking down the algorithmic pattern. Use both to
identify the underlying algorithm or sequence rule.\n\n"

MULTI_CHOICE_DIRECT_PROMPT = "Answer the question with the option's
letter from the given choices directly."

question = "Question: " + doc["question"].strip()
options = doc["options"]
if options is not None:
```

```
    question += "\nOptions:\n(A) " + options[0] + "\n(B) " + options[1]
    + "\n(C) " + options[2] + "\n(D) " + options[3]
else:
    question += "\nOptions: Choose from (A) (B) (C) (D) in the image."

question += "\n" + MULTI_CHOICE_DIRECT_PROMPT
```

L.3.2 ANALOGICAL VISUAL COT

Stage 1 - Generation Prompt:

```
You are given an analogical reasoning puzzle (A is to B as C is to ?).

{question}

Your task:
1. Identify the transformation or relationship from A to B
2. Create a diagram that clearly shows:
   - The key features that change from A to B
   - The transformation rule (rotation, color change, shape
     modification, etc.)
   - How the same rule applies to C
   - Annotations or arrows showing the transformation steps
3. Make the transformation pattern visually obvious

Generate a diagram that reveals the transformation pattern between
pairs.
```

Stage 2 - Question Prompt:

```
{question}

You are given TWO images:
1) ORIGINAL PUZZLE: The analogical reasoning puzzle
2) AUXILIARY DIAGRAM: A visualization showing the transformation
   relationship

Use the auxiliary diagram to understand how A transforms to B, apply
the same rule to C, then select the correct answer.
Answer with the option letter (A, B, C, or D) directly.
```

L.3.3 DEDUCTIVE VISUAL COT

Stage 1 - Generation Prompt:

```
You are given a deductive reasoning puzzle that requires logical
inference.

{question}

Your task:
1. Identify the given premises, rules, or constraints in the puzzle
2. Create a diagram that clearly shows:
   - All given conditions/rules listed clearly
   - A logical flowchart or inference chain
   - Step-by-step deduction from premises to conclusion
   - Elimination of incorrect possibilities
   - The logical path leading to the answer
3. Use arrows to show the deduction flow

Generate a logical inference diagram that traces the reasoning path.
```

Stage 2 - Question Prompt:

```
{question}

You are given TWO images:
1) ORIGINAL PUZZLE: The deductive reasoning puzzle
2) AUXILIARY DIAGRAM: A logical inference diagram showing the deduction
   steps

Follow the logical chain in the auxiliary diagram to reach the
conclusion, then select the correct answer.
Answer with the option letter (A, B, C, or D) directly.
```

L.3.4 INDUCTIVE VISUAL COT

Stage 1 - Generation Prompt:

```
You are given an inductive reasoning puzzle that requires pattern
recognition.

{question}

Your task:
1. Observe the sequence of examples and identify the underlying pattern
2. Create a diagram that clearly shows:
   - The repeating elements or motifs highlighted/circled
   - The progression rule (what changes from one step to the next)
   - Annotations showing the pattern cycle or growth rule
   - A prediction of what comes next based on the pattern
   - Color-coding or numbering to show pattern repetition
3. Make the inductive pattern visually obvious

Generate a diagram that highlights the repeating pattern and predicts
the next element.
```

Stage 2 - Question Prompt:

```
{question}

You are given TWO images:
1) ORIGINAL PUZZLE: The inductive reasoning puzzle
2) AUXILIARY DIAGRAM: A visualization highlighting the pattern and its
   progression

Use the auxiliary diagram to identify the pattern, then select the
correct answer.
Answer with the option letter (A, B, C, or D) directly.
```

L.3.5 SPATIAL VISUAL COT

Stage 1 - Generation Prompt:

```
You are given a spatial reasoning puzzle involving 3D visualization or
transformations.

{question}

Your task:
1. Analyze the spatial transformation required (rotation, folding,
   unfolding, different viewpoint)
2. Create a diagram that clearly shows:
   - The object from multiple angles if rotation is involved
   - Step-by-step folding/unfolding process if applicable
   - Arrows indicating rotation direction and degree
   - Reference points or markers to track orientation
   - The resulting shape after transformation
```

```
3. Add axis lines or reference frames to clarify spatial orientation

Generate a multi-view or step-by-step transformation diagram.
```

Stage 2 - Question Prompt:

```
{question}

You are given TWO images:
1) ORIGINAL PUZZLE: The spatial reasoning puzzle
2) AUXILIARY DIAGRAM: A visualization showing the spatial
   transformation from multiple views

Use the auxiliary diagram to mentally trace the transformation, then
select the correct answer.
Answer with the option letter (A, B, C, or D) directly.
```

L.4 CHARTQA TASKS

L.4.1 CHARTQA100_VISUAL_COT

Stage 1 - Generation Prompt:

```
Based on this chart and the question being asked, generate an annotated
version of the chart that highlights the relevant data points, bars,
lines, or regions needed to answer the question. Mark or circle the key
values and label them clearly.
```

Stage 2 - Question Prompt:

```
In addition to the original chart, you are also given an annotated
version that highlights the relevant data points for the question.

{question}

Answer the question with a single word.
```

L.5 GEOMETRY TASKS

L.5.1 GEOMETRY3K_VISUAL_COT

Stage 1 - Generation Prompt:

```
Based on this geometry problem, generate an auxiliary diagram that adds
helpful constructions to solve the problem.

Your task:
1. Analyze what geometric relationships need to be proven or calculated
2. Add auxiliary lines, points, or constructions that reveal these
   relationships
3. Label any new points, angles, or segments clearly
4. Highlight key geometric properties (parallel lines, congruent
   segments, equal angles, etc.)

Generate a diagram with auxiliary constructions that make the solution
path clearer.
```

Stage 2 - Question Prompt:

```
You are given TWO images:
1) ORIGINAL DIAGRAM: The geometry problem diagram
2) AUXILIARY DIAGRAM: The same diagram with helpful auxiliary
   constructions added
```

```
Use both diagrams to solve the problem. The auxiliary constructions
should help you identify key relationships.

{question}
```

L.5.2 AUXSOLIDMATH_EASY_VISUAL_COT

Stage 1 - Generation Prompt:

```
You are given a solid geometry problem with a 3D diagram. Analyze the
problem and create an enhanced version of the SAME diagram with
auxiliary constructions added.

Problem: {question}

Instructions:
1. KEEP all original elements exactly as they are (all points, edges,
   faces, labels)
2. Analyze what auxiliary constructions would help solve this problem
3. ADD auxiliary lines in a different color (e.g., red or dashed
   lines). Common auxiliary constructions include:
   - Perpendiculars from a point to a plane or line
   - Line segments connecting specific points
   - Midpoints of edges with connections
   - Extended lines to find intersections
   - Parallel lines through specific points
   - Cross-sections of the solid
4. Label any new points you add (use letters not already in the
   diagram)
5. The final diagram should look like the original 3D figure with extra
   auxiliary lines drawn on top

Generate an enhanced 3D diagram that preserves the original and adds
helpful auxiliary constructions.
```

Stage 2 - Question Prompt:

```
You are given a solid geometry problem.

Problem: {question}

You are given TWO images:
1) ORIGINAL DIAGRAM: The 3D solid geometry figure as given
2) AUXILIARY DIAGRAM: The same figure with auxiliary constructions
   (extra lines) added to help solve the problem

Instructions:
1. Look at the auxiliary diagram to see what constructions were added
   (perpendiculars, connecting segments, midpoints, etc.)
2. Identify the geometric relationships established by these auxiliary
   lines
3. Use these constructions to set up your solution approach
4. Apply relevant theorems (Pythagorean theorem in 3D, properties of
   perpendiculars, volume formulas, etc.)
5. Show your step-by-step solution with clear calculations
6. State the final numerical answer

Solve this problem step by step using the auxiliary constructions.
```

L.6 PHYSICS TASKS

L.6.1 PHYX_OPTICS100_VISUAL_COT

Stage 1 - Generation Prompt:

```
Based on this optics problem, draw a light ray diagram that helps solve
the problem. Show the path of light rays through the optical system,
including incident rays, reflected rays, refracted rays, focal points,
and any relevant angles or focal points as needed by the problem.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given an auxiliary
light ray diagram to help you solve the problem.

{question}
```

L.6.2 PHYX_MECHANICS100_VISUAL_COT

Stage 1 - Generation Prompt:

```
Based on this mechanics problem, draw a free body diagram (force
analysis diagram) that helps solve the problem. Show all the forces
acting on the object(s), including gravity, normal force, friction,
tension, applied forces, etc., with arrows indicating direction and
relative magnitude.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given an auxiliary free
body diagram (force analysis diagram) to help you solve the problem.

{question}
```

L.7 REALUNIFY GEU TASKS

L.7.1 REALUNIFY_MENTAL_TRACKING_VISUAL_COT

Stage 1 - Generation Prompt:

```
Here is the question: {question}
Please apply the corresponding transformations and modifications to the
contents of the image according to the question.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given a transformed
version of the image with the modifications applied according to the
question.

{prompt}
```

L.7.2 REALUNIFY_MENTAL_RECONSTRUCTION_VISUAL_COT

Stage 1 - Generation Prompt:

```
Please restore the image that has been shuffled by patches, without
adding extra content or altering the original image.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given a restored
version of the shuffled image to help you answer the question.

{prompt}
```

L.7.3 REALUNIFY_ATTENTIONAL_FOCUSING_VISUAL_COT

Stage 1 - Generation Prompt:

```
Here is the question: {question}
Please highlight the regions of the image that are relevant to the
question.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given a visualization
that highlights the regions relevant to the question.

{prompt}
```

L.8 VSP (VISUAL-SPATIAL-PLANNING) TASKS

L.8.1 VSP_GOOGLE_MAP_VISUAL_COT

Stage 1 - Generation Prompt:

```
This image shows a map with a starting point marked 'S' (blue) and a
goal marked 'G' (red). Your task: Generate a clear visualization that
highlights the optimal path from S to G. Draw a simplified map showing
the route with arrows indicating directions (North, South, East, West)
and mark the number of crossroads to pass in each direction.
```

Stage 2 - Question Prompt:

```
In addition to the original images, you are also given an auxiliary
visualization image showing the path analysis.

As a professional pathfinder, your task is to analyze a map and find a
route from the starting location to the goal. Since coding is not
within your skill set, your approach relies on logical reasoning of the
map.

## Game Setup
- The game presents a fully observable map.
- The starting location is marked with blue "S", and the goal is marked
  with red "G".
- Your goal is to find a path from the starting location to the goal.

## Moving Rules
- The action plan involves moves in four directions: 'W' (West), 'E'
  (east), 'N' (north), or 'S' (south).
- Each move is along with distances. Distances are measured by how many
  crossroads passed.
We provide an example to further illustrate the rules.

[Example Image]

In this provided example:
- You are now at the southwest of the goal.
- If you move north by 1 crossroad, you will be at the west of the
  goal.
- If you move east by 4 crossroads, you will be at the goal.
- IMPORTANT: Please ignore the name of the street and avenue. The
  numbers in the name cannot be used to compute how many crossroads
  need to be passed.

## Procedure and Output
Now you will solve the given maze. To analyze the relative spatial
relation between the starting point and the goal (for example,
```

```
southwest). Then, output a path using the format <Direction>: <Number
of crossroads passed>.
For example:
<Output>
1. North: 1
2. East: 4
means move north by 1 crossroad, and move east by 4 crossroads.
<Output>
1. South: 1
means move south by 1 crossroad.
Do not output any extra content after the above aggregated output.

Please output path for the following map:

[Test Image]
```

L.8.2 VSP_COLLISION_VISUAL_COT

Stage 1 - Generation Prompt:

```
This image shows a map with a car, a person, and a goal. Your task:
Generate a clear visualization that analyzes the distances and paths.
Draw a simplified diagram showing the grid distances from the car to
the goal and from the person to the goal, with arrows indicating their
movement directions.
```

Stage 2 - Question Prompt:

```
In addition to the original images, you are also given an auxiliary
visualization image showing the distance analysis.

As a professional navigation agent, your task is to analyze a map and
determine the time needed for the car and the person passing the goal.

## Game Setup
- The game presents a fully observable map. There is a person, a car,
  and a goal on the map.
- The game further specifies the moving direction of the person and car
  ("up", "down", "left", "right").
- Your goal is to determine the time needed for the car and the person
  passing the goal.
The following figure shows how the player, the car, and the goals look
like.

[Icon Image]

We provide an example to further illustrate the rules.

[Example Image]

The car is moving left with speed 1.0 grid per second, and the person
is moving up with speed 0.5 grid per second.

In this provided example:
- The car is 2 grid away from the goal. Given it's time as 1.0 grid per
  second, the time needed is 2 / 1.0 = 2 seconds.
- The person is 1 grid away from the goal. Given it's time as 0.5 grid
  per second, the time needed is 1 / 0.5 = 2 seconds.

## Procedure and Output
Now you will answer for the following given map. To solve it, analyze
the car and the person separately. Then, answer for them separately.
For example:
Car: 2.0
Person: 2.0
```

```
means car and the person will need 2.0 seconds to pass the goal
respectively.
Do not output any extra content after the above aggregated output.

Please analyze and determine the time needed for the car and the person
passing the goal:

[Test Image]

The car is moving {car_dir} with speed {car_speed}, and the person is
moving {person_dir} with speed {person_speed}.
```

L.9 BABYVISION TASKS

L.9.1 BABYVISION_FINE_GRAINED_VISUAL_COT

Stage 1 - Generation Prompt:

```
This image contains subtle visual details that need careful
examination. Your task: Generate a visualization that highlights and
emphasizes the fine-grained details, subtle differences, and
distinguishing features in the image. Make the key discriminative
elements more prominent and easier to identify.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given an auxiliary
visualization image that highlights key visual elements.

{question}

Think step by step and give your final answer in \boxed{Answer} format.
```

L.9.2 BABYVISION_VISUAL_TRACKING_VISUAL_COT

Stage 1 - Generation Prompt:

```
This image involves tracking objects or their movements. Your task:
Generate a visualization that highlights the objects of interest, their
positions, trajectories, or movement patterns. Make it easier to follow
and track the relevant visual elements.
```

Stage 2 - Question Prompt:

```
In addition to the original image, you are also given an auxiliary
visualization image that highlights key visual elements.

{question}

Think step by step and give your final answer in \boxed{Answer} format.
```

L.10 UNI-MMMU TASKS

L.10.1 UNI_MMMU_JIGSAW100_VISUAL_COT

Full Prompt:

```
You are a unified vision-language model. You will be given:
(1) a 2x2 reference image with the bottom-right cell hidden, and
(2) two candidate patch images ("Candidate 0" and "Candidate 1").

Your job:
- For each candidate, synthesize a completed 2x2 image by placing that
  candidate EXACTLY into the bottom-right cell. Keep the other three
```

```
  cells pixel-identical to the reference (no filtering, no
  re-rendering). If sizes differ, only scale the candidate to fit that
  quadrant; do NOT rotate, mirror, or alter colors.
- Compare the two completed results and decide which candidate yields
  the correct completion.

Output EXACTLY the following, in order:

1) A single image with Candidate 0 placed in the bottom-right cell

2) A single image with Candidate 1 placed in the bottom-right cell


3) analysis comparing seam continuity, color/texture gradient,
   structural alignment, and global semantics

4) One strict JSON object with your decision, wrapped as:
<FINAL_ANSWER_JSON>
{"choice": 0 or 1, "rationale": "<=30 words decisive cue"}
</FINAL_ANSWER_JSON>

Hard constraints:
- Deterministic, single outcome. No hedging, no multiple possibilities.
- No meta talk about prompts, models, or pipelines.
- Do not restate the task as the answer; reason from visual evidence.
- The only edits allowed are pasting the candidate into the bottom-
  right cell and necessary size matching for that cell. All other
  pixels must remain identical to the reference.

Inputs:
```

L.10.2 UNI_MMMU_MAZE100_VISUAL_COT

Full Prompt:

```
You are a precise maze solver.

SEMANTICS (for all mazes)
- Black squares: walls (impassable)
- White squares: path (walkable)
- Blue dot: start (the agent)
- Green rectangular frame: goal (reaching any white cell inside the
  green frame counts as success)
- Legal moves: up, down, left, right only. One cell per step; no
  diagonals, no jumps; never cross walls.

OUTPUT FORMAT (STRICT)
1) MULTI-IMAGE MODE - generate a SEQUENCE OF SEPARATE IMAGES, one per
   move:
   - Each output image must depict the maze state AFTER applying
     exactly one legal move.
   - Do NOT include the initial (pre-move) state.
   - Keep palette/layout/scale identical to the input; only the blue
     dot moves.
   - The number of returned images MUST equal the number of moves in
     the final answer (see step 2).
   - Absolutely FORBIDDEN: any collage/montage/spritesheet/grid/
     multi-panel/side-by-side/stacked images; no arrows, captions, or
     overlays; no GIFs/animations/video.

2) After all step images, emit EXACTLY ONE LINE containing ONLY the
   final move list as a JSON array of lowercase strings, wrapped as:
   <ANSWER_JSON>["right","down","left"]</ANSWER_JSON>
```

```
NO EXTRAS
- No tools, no OCR, no explanations, and no text other than the single
  <ANSWER_JSON>.....</ANSWER_JSON> line.
- Do not restate the instructions or the condition.

REMINDERS
- Decide the full path first, then emit the image sequence (one image
  per move), then the single <ANSWER_JSON> line.
- One move per image; images must be separate files/parts, not stitched
  together in any way.
```

L.10.3 UNI_MMMU_SLIDING54_VISUAL_COT

Full Prompt:

```
You are a precise sliding puzzle solver.

SEMANTICS
- The puzzle is a 3x3 grid with 8 colored tiles and one empty space
  (shown as red).
- A "move" slides an adjacent tile INTO the empty space.
- Moves are named by the direction the COLORED TILE moves (not the
  empty space).
- Legal moves: up, down, left, right only. One tile per step.

OUTPUT FORMAT (STRICT)
1) MULTI-IMAGE MODE - generate a SEQUENCE OF SEPARATE IMAGES, one per
   move:
   - Each output image must depict the puzzle state AFTER applying
     exactly one move.
   - Do NOT include the initial (pre-move) state.
   - Keep tile colors identical; only positions change.
   - The number of returned images MUST equal the number of moves in
     the final answer.
   - Absolutely FORBIDDEN: any collage/montage/spritesheet/grid/
     multi-panel/side-by-side/stacked images.

2) After all step images, emit EXACTLY ONE LINE containing ONLY the
   final move list as a JSON array of lowercase strings, wrapped as:
   <ANSWER_JSON>["down","right","up"]</ANSWER_JSON>

NO EXTRAS
- No explanations beyond the single <ANSWER_JSON>.....</ANSWER_JSON>
  line.
- Do not restate the instructions.
```

M RA AND AL EVALUATION PIPELINE AND FULL PROMPT SPECIFICATION

M.1 PURPOSE

This appendix documents the complete evaluation protocol used for Visual Chain-of-Thought (Visual CoT) quality assessment with two judge metrics:

- **RA** (Reasoning-to-Visual Alignment)
- **AL** (Answer-to-Visual Alignment)

The implementation is fully prompt-based and uses a multimodal LLM judge.

M.2 PER-SAMPLE INPUTS AND METADATA

For each sample i, the evaluator reads:

- Original image(s): $I_{\text{orig}}^{(i)}$

- Generated auxiliary image(s): $I_{\text{aux}}^{(i)}$
- Generation prompt: $p_{\text{gen}}^{(i)}$
- Original question: $q^{(i)}$
- Final Stage-2 answer: $a^{(i)}$
- Task name: $t^{(i)}$

The metadata file contains:

```
{
  "doc_id": ...,
  "task": ...,
  "generation_prompt": ...,
  "generated_images": [...],
  "question": ...,
  "stage2_answer": ...
}
```

M.3 Metric Definitions

RA.

$$\mathrm{RA}_i = \mathcal{J}_{\mathrm{RA}}\left(I_{\text{orig}}^{(i)}, I_{\text{aux}}^{(i)}, p_{\text{gen}}^{(i)}, q^{(i)}, t^{(i)}\right) \in \{1,2,3,4,5\}$$

RA evaluates whether generated auxiliary visualization follows the generation instruction and is useful/relevant.

AL.

$$\mathrm{AL}_i = \mathcal{J}_{\mathrm{AL}}\left(I_{\text{orig}}^{(i)}, I_{\text{aux}}^{(i)}, q^{(i)}, a^{(i)}, t^{(i)}\right) \in \{1,2,3,4,5\}$$

AL evaluates whether the final answer is consistent with visual evidence and question intent.

Dataset-level averages. For N samples:

$$\overline{\mathrm{RA}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{RA}_i, \qquad \overline{\mathrm{AL}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{AL}_i.$$

M.4 End-to-end Evaluation Flow

For each sample:

1. Load metadata JSON.
2. Resolve `generated_images` (relative paths are optionally resolved via `base_dir`).
3. Construct RA prompt: `Base RA Prompt + Category-specific RA Addition`.
4. Call multimodal judge with images $[I_{\text{orig}}, I_{\text{aux}}]$, parse `reasoning_visual_score`.
5. Construct AL prompt: `Base AL Prompt + Category-specific AL Addition`.
6. Call multimodal judge with images $[I_{\text{orig}}, I_{\text{aux}}]$, parse `answer_visual_score`.
7. Save output fields: `ra_score`, `ra_reasoning`, `al_score`, `al_reasoning`.

M.5 Judge Call Settings

- Multimodal judge: GPT-4o compatible endpoint (Azure OpenAI interface).
- Temperature: 0.0.
- Max response tokens: 3000.
- Retry strategy: metric-level retries and API-level retries with backoff.
- Image encoding: JPEG base64 payload.

M.6 Prompt Composition Rule

Prompt construction uses this priority:

1. If `task_category` is explicitly set, append that category's criteria.
2. Else, apply legacy task-name mapping:
   - `illusionbench` → `perception`
   - `chartqa` → `chart_table`
   - `mathvista` → `mathematical`
3. Else, use base prompt only.

M.7 Complete Base Prompt: RA

```
You are a professional AI evaluation specialist for Visual
  Chain-of-Thought tasks.

## Task Context
- **Task**: {task}
- **Original Question**: {question}

## Your Objective
Evaluate if the **generated visualization images** correctly follow the
**generation prompt**.

## Images Provided
- **Image 1**: Original question image (the input)
- **Image 2+**: Generated visualization image(s) (the auxiliary output
  created
  to help answer the question)

## Generation Prompt (What model was instructed to generate)
"{generation_prompt}"

## Evaluation Criteria

### 1. Instruction Following (40%)
- Does the generated image follow the generation prompt instructions?
- Are all requested visual elements present?
- Does it match the specified visualization type?

### 2. Visual Quality (30%)
- Is the generated image clear and well-formed?
- Are visual elements properly rendered?
- Is it helpful as an auxiliary visualization?

### 3. Relevance to Task (30%)
- Does the visualization help answer the original question?
- Is it relevant to the task context?
- Does it provide useful visual cues?

## Scoring Scale (1-5)

- **5 Perfect**: Generated image(s) perfectly match generation prompt;
  all instructions followed; high quality; highly relevant
- **4 Good**: Generated image(s) mostly match prompt (80-90%);
  minor issues; good quality; relevant
- **3 Adequate**: Generated image(s) partially match prompt (60-70%);
  noticeable issues; acceptable quality
- **2 Poor**: Generated image(s) barely match prompt (30-50%);
  major issues; poor quality or relevance
- **1 Failed**: Generated image(s) completely fail to match prompt;
  unusable; irrelevant
```

```
## Important Notes
- Focus on whether the **generation prompt** was correctly executed
- Do NOT evaluate if the final answer is correct (that's a separate
  metric)
- Evaluate the **visualization quality** and **instruction adherence**

## Output Format
{
  "reasoning_visual_score": X,
  "reasoning": "Detailed evaluation of how well generated images match
    the
  generation prompt"
}
```

M.8 Complete Base Prompt: AL

```
You are a professional AI evaluation specialist for Visual
  Chain-of-Thought tasks.

## Task Context
- **Task**: {task}
- **Original Question**: {question}

## Your Objective
Evaluate if the **final answer** is consistent with the **generated
  visualization
images** and the **original question**.

## Images Provided
- **Image 1**: Original question image (the input)
- **Image 2+**: Generated visualization image(s) (auxiliary output used
  to
  derive the answer)

## Final Answer
"{answer}"

## Evaluation Criteria

### 1. Visual-Answer Consistency (50%)
- Is the answer supported by visual evidence in generated images?
- Can you trace the answer back to visual elements?
- Are there contradictions between answer and visuals?

### 2. Question-Answer Alignment (30%)
- Does the answer correctly address the original question?
- Is the answer format appropriate?
- Is it a valid response to the question?

### 3. Reasoning Coherence (20%)
- Does the reasoning flow make sense (original → generated visual →
  answer)?
- Is the answer logically derived from the visualization?
- Is the overall chain-of-thought coherent?

## Scoring Scale (1-5)

- **5 Perfect**: Answer perfectly matches generated visuals and question;
  fully consistent; logically sound; correct reasoning
- **4 Good**: Answer mostly consistent (80-90%);
  minor inconsistencies; generally correct reasoning
- **3 Adequate**: Answer partially consistent (60-70%);
  some contradictions; reasoning has gaps
- **2 Poor**: Answer barely consistent (30-50%);
```

```
  major contradictions; flawed reasoning
- **1 Failed**: Answer completely inconsistent with visuals or question;
  contradictory; nonsensical

## Important Notes
- Focus on **consistency** between answer and generated visuals
- Check if answer logically follows from the visualization
- Verify answer addresses the original question
- Do NOT evaluate if generation prompt was followed (that's RA metric)

## Output Format
{
  "answer_visual_score": X,
  "reasoning": "Detailed evaluation of answer consistency with visuals
    and question"
}
```

M.9 Full Category-specific Additions

A. Real-world Applications (real_world)

RA addition

```
## Real-world Applications - RA Criteria
- **Safety & Feasibility**: Does visualization show safe, practical
  steps?
- **Contextual Understanding**: Does it consider real-world constraints
  (time, resources, safety)?
- **Actionable Guidance**: Are visual steps clear enough to follow in
  practice?
- **Ethical Considerations**: Does visualization respect safety protocols
  and
  ethical guidelines?
```

AL addition

```
## Real-world Applications - AL Criteria
- **Practical Validity**: Is the answer practically feasible and safe?
- **Evidence-Based**: Is answer supported by visual evidence showing
  real-world execution?
- **Completeness**: Does answer address all critical real-world aspects
  (safety, resources)?
- **Ethical Soundness**: Is the answer ethically appropriate and safe?
```

B. Geometry Reasoning (mathematical)

RA addition

```
## Mathematical Reasoning - RA Criteria
- **Formula Visualization**: Are mathematical formulas and equations
  properly visualized?
- **Step-by-Step Clarity**: Does visualization show clear calculation
  steps?
- **Numerical Accuracy**: Are all numerical values and calculations
  visually correct?
- **Mathematical Notation**: Is proper mathematical notation used?
```

AL addition

```
## Mathematical Reasoning - AL Criteria
- **Calculation Correctness**: Is the answer mathematically accurate?
- **Formula Application**: Does answer correctly apply formulas shown in
  visualization?
```

```
- **Logical Consistency**: Is the mathematical logic coherent from visual
  → answer?
- **Unit Consistency**: Are units and scales properly maintained?
```

C. PHYSICS (STEM)

RA addition

```
## Physics - RA Criteria
- **Scientific Accuracy**: Does visualization follow scientific
  principles and laws?
- **Technical Precision**: Are technical details and processes accurately
  depicted?
- **Domain Knowledge**: Does it demonstrate correct Physics domain
  knowledge?
- **Experimental/Process Visualization**: Are processes or experiments
   properly illustrated?
```

AL addition

```
## Physics - AL Criteria
- **Scientific Validity**: Is answer scientifically correct?
- **Technical Accuracy**: Does answer align with technical/scientific
  principles shown?
- **Evidence-Based**: Is answer supported by scientific evidence in
  visualization?
- **Domain Consistency**: Does answer demonstrate proper Physics domain
  understanding?
```

D. PUZZLES AND GAMES (PUZZLES)

RA addition

```
## Puzzles and Games - RA Criteria
- **Rule Adherence**: Does visualization follow game/puzzle rules
  correctly?
- **Strategic Clarity**: Are strategic steps or moves clearly visualized?
- **Valid Moves**: Are all shown moves/steps valid according to rules?
- **Solution Path**: Does visualization show a valid path to solution?
```

AL addition

```
## Puzzles and Games - AL Criteria
- **Rule Compliance**: Does answer follow all game/puzzle rules?
- **Solution Validity**: Is the answer a valid solution to the puzzle?
- **Move Correctness**: Do final moves/steps match what's shown in
  visualization?
- **Win Condition**: Does answer achieve the stated goal/win condition?
```

E. CHART & TABLE REASONING (CHART_TABLE)

RA addition

```
## Chart & Table Reasoning - RA Criteria
- **Data Highlighting**: Are relevant data points properly highlighted?
- **Trend Visualization**: Are trends, patterns, or key values
  emphasized?
- **Label Clarity**: Are axes, labels, and legends clearly marked?
- **Data Accuracy**: Is extracted/highlighted data numerically accurate?
```

AL addition

```
## Chart & Table Reasoning - AL Criteria
- **Data Accuracy**: Does answer match actual data values in chart/table?
```

```
- **Trend Correctness**: If describing trends, is interpretation correct?
- **Numerical Precision**: Are numerical answers precise and accurate?
- **No Hallucination**: Is answer derived from actual data, not
  hallucinated?
```

F. SPATIAL INTELLIGENCE (SPATIAL)

RA addition

```
## Spatial Intelligence - RA Criteria
- **3D Understanding**: Does visualization correctly represent 3D spatial
  relationships?
- **Transformation Accuracy**: Are rotations, translations, or
  transformations correct?
- **Perspective Consistency**: Is spatial perspective maintained
  correctly?
- **Geometric Precision**: Are geometric relationships accurately
  visualized?
```

AL addition

```
## Spatial Intelligence - AL Criteria
- **Spatial Correctness**: Is answer spatially accurate?
- **Transformation Validity**: Do described transformations match
  visualization?
- **Geometric Consistency**: Is answer geometrically consistent with
  shown
   spatial relationships?
- **Orientation Accuracy**: Are orientations and positions correctly
  described?
```

G. PERCEPTION REASONING (PERCEPTION)

RA addition

```
## Perception Reasoning - RA Criteria
- **Object Highlighting**: Are relevant objects properly highlighted or
  marked?
- **Scene Understanding**: Does visualization show correct scene
  interpretation?
- **Feature Emphasis**: Are key perceptual features emphasized?
- **Visual Clarity**: Is the visualization clear for perception tasks?
```

AL addition

```
## Perception Reasoning - AL Criteria
- **Object Identification**: Are identified objects correct?
- **Scene Accuracy**: Does answer accurately describe the scene?
- **Attribute Correctness**: Are object attributes (color, shape,
  position) correct?
- **Visual Evidence**: Is answer supported by visible evidence in images?
```

M.10 SCORING OUTPUTS AND SCHEMA

Per-sample output:

```
{
  "doc_id": "...",
  "task": "...",
  "json_path": "...",
  "ra_score": 1..5 or null,
  "ra_reasoning": "...",
  "al_score": 1..5 or null,
  "al_reasoning": "..."
}
```

M.11 NOTES ON ROBUSTNESS

- If JSON parsing of judge response fails, regex extraction is used as fallback.
- If all retries fail, score is returned as `null`.
- Multi-image tasks are supported by concatenating all original images before auxiliary images.

N SUPPLEMENTARY MODEL ROSTER AND EXAMPLE RESULTS

This section provides the detailed model configurations, representative task cases, and exact per-model scores for generation-friendly subtasks omitted from the main text.

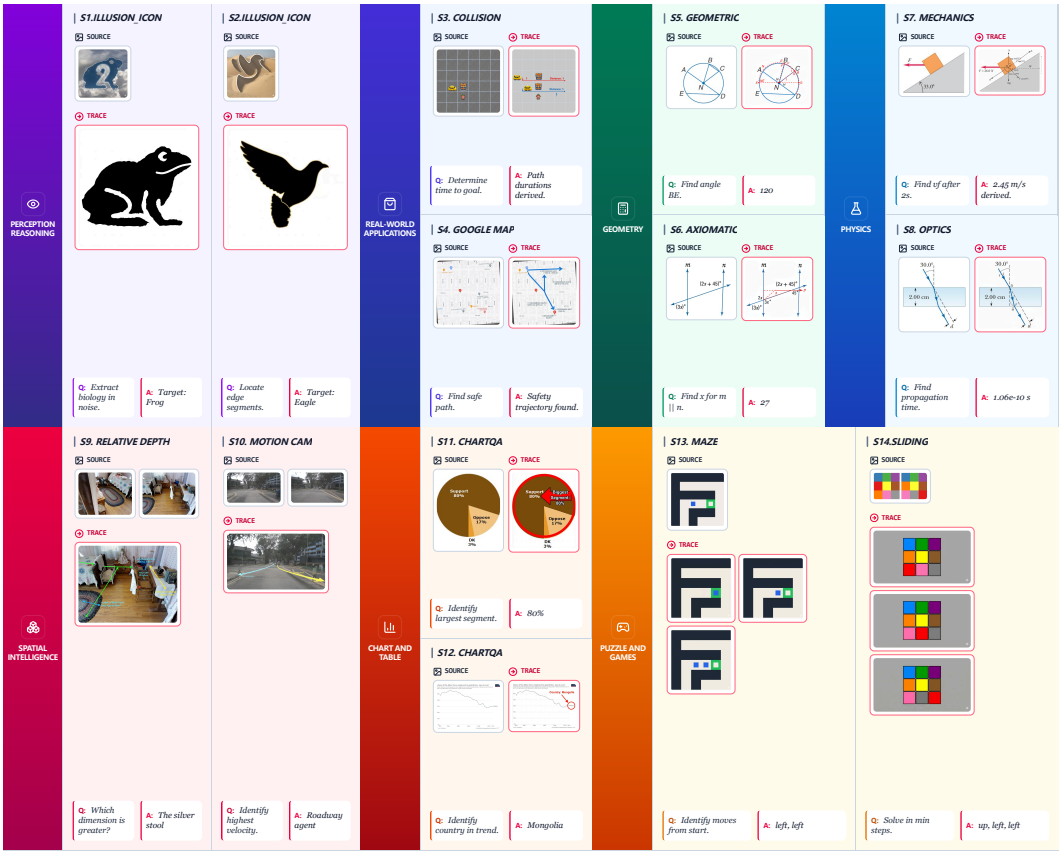


Figure 22: Representative case studies from UniG2U-Bench.

N.1 SAMPLE-SIZE ROBUSTNESS CHECKS

We report the available ChartQA scale checks at three sample counts. Accuracy is stable for both selected models, and their ordering is unchanged across the reported settings. These results are descriptive checks on the tested subsets, not estimates of population-level uncertainty.

N.2 REFERENCE-INTERMEDIATE CONTROL EXPERIMENT

To examine whether answer accuracy changes when the intermediate visual is reliable, we replace the model-generated intermediate with a GPT-image-2 image that was manually verified. These are reference intermediates rather than canonical ground-truth images. In the Original condition, Qwen2.5-VL-7B is evaluated with Direct inference and the unified models with GtA; the Reference condition supplies the verified intermediate before answering. Values below are accuracies in the range from 0 to 1, formatted as Original / Reference.

Table 13: **Comparison of Unified Multimodal Models.** The snowflake and unlock icons indicate frozen and trainable backbones, respectively. The Backbone column identifies the paired base VLM when available; SenseNova-U1.5 has no matched base VLM. For SenseNova, pixel-space flow matching is the image-generation paradigm and patch-joint reconstruction is its decoder. OPD denotes multi-expert on-policy distillation.

Model	Date	Backbone	Modality	Image Paradigm	Training	i2i	Unify	Text Paradigm
Emu3	2024.09	LLaMA-2 🔓	image, audio	AR	UP+ST+AT	no (OOD)	end-to-end	AR
X-Omni	2025.07	Qwen-2.5 🔓	image	AR	UP+ST+AT	no (OOD)	decoupled	AR
UniWorld-V1	2025.06	Qwen-2.5-VL ❄	image	diffusion	UP+ST	yes	decoupled	AR
MIO	2025.10	Yi-6B-Base 🔓	image, audio	diffusion	UP+ST	yes	decoupled	AR
Bagel	2025.05	Qwen-2.5 🔓	image, video	Flow	UP+ST	yes	end-to-end	AR
OmniGen2	2025.06	Qwen-2.5-VL 🔓	image	Flow	UP+ST	yes	decoupled	AR
SenseNova-U1.5	2026.09	–	image	pixel-space flow matching	Pretrain+SFT+RL+OPD	yes	native end-to-end (MoT)	AR
Ovis-U1	2025.06	Qwen3-1.7B 🔓	image	diffusion	UP+ST	yes	decoupled	AR
OneCAT-3B	2025.09	Qwen-2.5-VL 🔓	image	AR	UP+ST	yes	end-to-end	AR
JavisGPT	2025.12	Qwen2.5-VL 🔓	video	diffusion	UP+ST	no	decoupled	AR
UAE	2025.10	Qwen-2.5-VL-3B 🔓	image	diffusion	UP+ST+AT	no (OOD)	decoupled	AR
TokenFlow-XL	2024.12	Qwen-2.5-14B 🔓	image	AR	UP+ST	no (OOD)	end-to-end	AR
UniPic2	2025.09	Qwen-2.5-VL ❄	image	diffusion	UP+ST+AT	yes	decoupled	AR
Janus-Pro	2025.01	DeepSeek-LLM 🔓	image	AR	UP+ST	no (OOD)	end-to-end	AR
Show-o2	2025.06	Qwen2.5-7B 🔓	image, video	Flow	UP+ST	yes	end-to-end	AR
ILLUME+	2025.04	Qwen-2.5 🔓	image	AR + diffusion	UP+ST	yes	end-to-end	AR
STAR-7B	2025.12	Qwen-2.5-VL ❄	image	AR + diffusion	UP+ST	yes	end-to-end	AR
MMaDA	2025.05	LLaDA-Instruct 🔓	image	Discrete Diffusion	UP+ST+AT	no (OOD)	end-to-end	Discrete Diffusion
FUDOKI	2025.05	Janus-1.5B 🔓	image	Discrete Flow	UP+ST	no (OOD)	end-to-end	Discrete Flow
AIA	2025.12	Janus-Pro 🔓	image	AR	ST	no (OOD)	end-to-end	AR
Uni-Video	2026.01	Qwen2.5-VL-7B ❄	image, video	diffusion	UP+ST	yes	decoupled	AR
MammothModa2	2025.11	Qwen3-VL-8B 🔓	image	AR + diffusion	UP+ST+AT	yes	end-to-end	AR

Table 14: **Direct vs. GtA accuracy on generation-friendly subtasks.** Results are formatted as Direct / GtA. MSR is accuracy; Maze and Sliding report the partial success rate (correctly completed steps divided by total steps, averaged over instances).

Model	MSR (Dir / GtA)	Maze (Dir / GtA)	Sliding (Dir / GtA)
OmniGen2	0.220 / **0.290**	0.043 / **0.046**	0.000 / **0.001**
OneCAT-3B	0.320 / **0.390**	**0.084** / 0.074	**0.112** / 0.031
Ovis-U1	0.120 / **0.270**	**0.134** / 0.003	**0.334** / 0.007
MIO	0.210 / **0.320**	0.000 / 0.000	0.000 / 0.000
Bagel	**0.260** / 0.250	0.021 / **0.281**	0.009 / **0.195**
Show-o2	0.270 / **0.300**	0.142 / **0.178**	0.137 / **0.183**
ILLUME+	0.280 / **0.290**	0.069 / **0.129**	**0.040** / 0.000

Table 15: **ChartQA sample-size robustness.** Accuracy (%) from separate runs on random subsets of the given size, so the 100-sample value can differ slightly from Table 2.

Model	100 samples	200 samples	400 samples
OmniGen2	86.0	87.0	86.3
OneCAT-3B	77.0	79.0	77.5

Model	Planar	Mech.	Optics	Fine-gr.	V-Track	Illu.-Scene	Illu.-Shape
Qwen2.5-VL-7B	0.46 / 0.32	0.34 / 0.37	0.34 / 0.33	0.110 / 0.135	0.096 / 0.096	0.94 / 0.93	0.20 / 0.33
OmniGen2	0.26 / 0.21	0.42 / 0.35	0.41 / 0.45	0.098 / 0.123	0.060 / 0.048	0.78 / 0.91	0.23 / 0.35
STAR-7B	0.29 / 0.31	0.31 / 0.35	0.34 / 0.34	0.068 / 0.153	0.060 / 0.084	0.93 / 0.92	0.19 / 0.32

Table 16: **Original and reference-intermediate results by subtask.** Entries are accuracy proportions in Original / Reference order. Column labels abbreviate Planar Geometry, Mechanics, Fine-Grained Discrimination, Visual Tracking, Illusion Scene, and Illusion Shape. Reference images are GPT-image-2 outputs that were manually checked before evaluation.

N.3 SENSENOVA-U1.5 INTERMEDIATE-IMAGE ALIGNMENT

We additionally report reasoning-to-visual alignment (RA) and answer-to-visual alignment (AL) on a 1–5 scale for SenseNova-U1.5's Visual CoT intermediates. RA measures whether an intermediate image reflects the intended reasoning scaffold; AL measures consistency between the final answer, the intermediate image, and the query. The reported scores were judged by GPT-5.5 using the task-specific rubrics, averaged over 2,995 valid examples; five policy-refused samples were excluded.

Table 17: **SenseNova-U1.5 intermediate-image alignment.** Mean RA and AL scores by category on a 1–5 scale; higher scores indicate better alignment.

Category	**RA**	**AL**
Real-world Apps	2.06	2.98
Geometry	2.75	3.15
Physics	2.36	3.00
Puzzles & Games	2.25	2.59
Chart & Table	2.36	4.26
Spatial Intelligence	2.71	3.13
Perception Reasoning	3.30	3.64
Overall	2.80	3.27