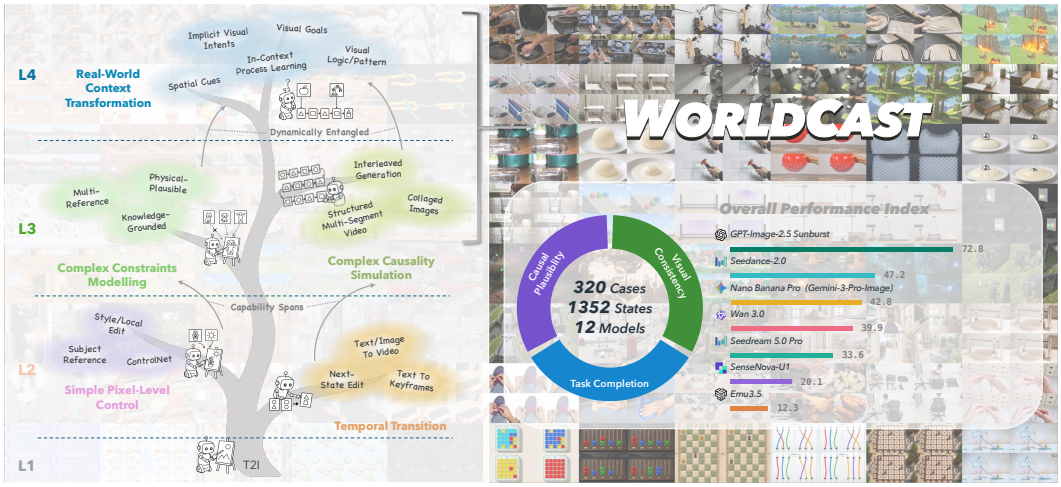
WORLDCAST: EXPLORING INTELLIGENT VISUAL GENERATION IN WORLD STATE PREDICTION

Anonymous authors
Paper under double-blind review



(a) (b)

Figure 1: Our generative capability tree and the WORLDCAST benchmark. (a) The proposed visual generative capability tree organizes intelligent visual generation into a four-layer hierarchy (L1-L4); (b) WORLDCAST targets L3-L4 level capabilities of dynamically transforming real-world context into successive future world states through unified multimodal understanding and generation.

ABSTRACT

Humans interpret multimodal observations of the world and forecast how it will evolve. Recent multimodal generative systems have demonstrated similar capabilities: they can integrate multimodal context and synthesize visual outputs that depict changes in the world. To characterize this emerging trajectory, we organize model capabilities into a four-level hierarchy and discover that higher-level tasks remain relatively underexplored despite model progress. To bridge this gap, we explore model performance on multi-step world state prediction, a high-level task that requires models to forecast successive future world states from multimodal observations and diverse references. Such a task requires models to possess entangled understanding and generation capabilities, as well as extensive world knowledge to translate human action sequences, physical transformations, and abstract visual reasoning into coherent visual outputs. We introduce WORLDCAST, comprising 320 manually designed cases across eight categories, and evaluate 12 representative models. The conducted experiments yield three key insights: (a) advances in frontier models are rapidly pushing the boundaries of intelligent world modeling, with the GPT-Image-2.5 System scoring 72.8 and demonstrating early-stage visual state prediction capabilities on real-world tasks; (b) WORLDCAST reveals substantial gaps between models: even Nano Banana Pro reaches only 42.8 and the best open-source model scores 20.1, with deficits in handling complex context, entangled references, and reasoning-heavy tasks; (c) strong generative models effectively draw on the capabilities of their underlying language models. These findings highlight the need for multimodal generative models that are capable of linguistically structured planning and reasoning while also being deeply unified to faithfully simulate real-world states.

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.

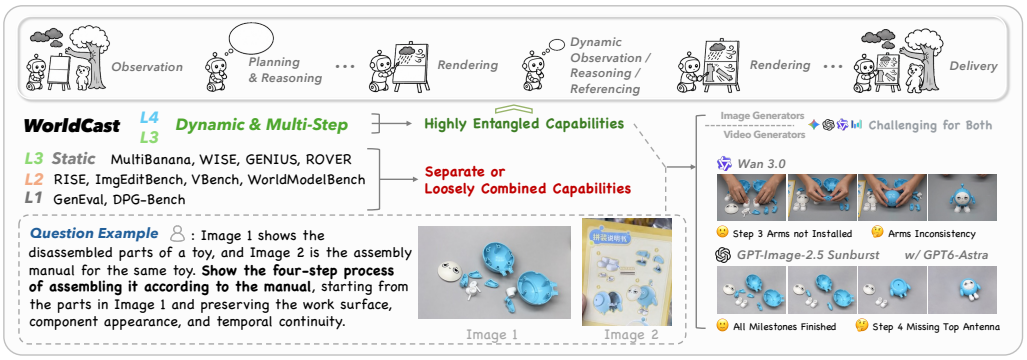


Figure 2: Evaluation scope and question example of WORLDCAST. The benchmark requires models to combine understanding and generation to dynamically produce evolving visual states. Even frontier models struggle to preserve task-critical details across the full sequence.

1 INTRODUCTION

Multimodal generative systems have advanced rapidly on both the understanding and generation sides: they can understand increasingly rich inputs and produce increasingly expressive visual outputs. Recent image and video generation systems can combine language with visual context Google DeepMind (2025); OpenAI (2025b), understand and utilize referenced content, and render multi-stage changes rather than isolated static scenes (Kuaishou Technology, 2026; Team Seedance, 2026). As with human intelligence, once these capabilities reach a sufficient level, they can in turn benefit real-world tasks (Zhang et al., 2026a). This trajectory resembles how humans learn and reason: we observe a multimodal world, draw upon relevant multimodal cues when confronting a problem, and envision complex ways in which the world may evolve.

To trace the progression of model capabilities, we organize them into four levels, illustrated in Figure 1. Basic text-to-image (T2I) ability focuses on a static generation process; editing, controllable generation, and video generation extend these capabilities to L2. As frontier models advance in both understanding and generation (Google DeepMind, 2025; OpenAI, 2025c; ByteDance Seed, 2025), they support interleaved or segmented generation and heterogeneous references, and the generation process itself becomes more intelligent by drawing on the strengths of powerful language models.

Despite rapid progress, higher-level model capabilities that require tightly integrating understanding and generation remain largely unexplored. Such capabilities, however, are highly relevant to the modeling of real-world tasks. Prior works mainly explore whether models can simulate our world from static appearance anchors (Bian et al., 2025; Li et al., 2026d), yet real-world events are mostly dynamic and predicting their progression requires references to intervene at different temporal steps. Those references may specify a state to observe, a goal, rules, patterns, object properties or relevant prior experience, rather than a starting point or appearance signals. We refer to this direction as ***world state prediction***: *utilizing heterogeneous context to guide an evolving real-world process*. Such a task requires a model not only to output successive visual states but also to reason about what each reference means and when and how it should take effect.

To explore model performance in this direction, we propose WORLDCAST, a comprehensive benchmark designed to answer the question: can visual generative models externalize L3-L4 level intelligence in world state prediction? WORLDCAST targets high-level simulation of real-world tasks, hierarchically distinguished by how deeply and dynamically a model must integrate reference understanding with control and knowledge. We collect L3-level cases focusing on complex dynamic transitions and physical plausibility, which require models to engage in implicit or explicit planning and reasoning. L4-level cases are designed for models to integrate references that may encode goals, physical states, materials, experience, semantics, or visual logic and translate them into a unified understanding that guides complex state transitions. The latter tasks provide richer contextual information yet place greater demands on tightly integrated multimodal intelligence.

To accommodate both image and video models, we evaluate all model outputs as ordered sequences of keyframes: regardless of whether generation is in a grid, interleaved, or in a continuous sequence, its success must ultimately be reflected in the resulting visual states. Based on the protocol, we fur-

ther introduce manually annotated case-specific hints for reliable sequence-level LMM-as-a-judge evaluation. The final benchmark evaluates 12 models in 320 test cases across 8 categories.

Our experiments show that frontier unified models have reached an early stage of intelligent world simulation: they can successfully handle some L3 and even L4 tasks and show emerging generalization across tasks, yet some domains remain challenging. We further find that interleaved generation improves the performance of frontier image models, enabling them to capture critical temporal transitions without relying on video architectures. We observe that the substantial model performance gaps are closely tied to their language-side capabilities. Building on this finding, we combine visual generation with agentic capabilities, including inspection-revision loops and coding tools, yielding a clearer understanding of their benefits and suggesting potential research paths toward more capable unified generation systems.

Our contributions are threefold:

(1) We formulate world state prediction as an evaluation problem for multimodal generative systems. We organize visual generative capability within a four-level hierarchy and study L3-L4 visual state prediction from heterogeneous real-life references.

(2) We introduce WORLDCAST, a benchmark of 320 manually designed cases across eight task categories and 1,352 requested visual states. We evaluate image and video systems through a common state-sequence protocol that measures three dimensions with case-specific annotations.

(3) We offer a systematic diagnosis of current generative systems. Evaluation of 12 models reveals substantial differences in model capabilities and widening gaps at higher levels. Analyses of generation schemes, language enhancement, and agentic behavior offer insights into persistent challenges and potential directions for simulating real-world states with generative intelligence.

2 WORLDCAST

WORLDCAST is a benchmark grounded in our cognitively motivated capability hierarchy. We distinguish task categories by where the information governing a future process is encoded and what reasoning is required to translate it into visual state transitions. The L3–L4 distinction reflects how tightly reference understanding and transition generation must be integrated with additional context, which may make a task easier for humans while demanding more integrated multimodal reasoning from models.

To simulate real-world tasks, WORLDCAST requires models to output causally coherent world states from heterogeneous references. Given text instructions p and one or more context images $\mathcal{X} = (x_m)_{m=1}^{M}$, the generative system G predicts an ordered visual sequence:

$$\widehat{\mathcal{Y}} = G(p, \mathcal{X}) = (\hat{y}_1, \ldots, \hat{y}_K), \tag{1}$$

where $\widehat{\mathcal{Y}}$ denotes the predicted sequence of K visual world states. As illustrated in Figure 3, WORLDCAST comprises eight task categories across two levels under this formulation:

• **Observation-Conditioned Transitions (L3).** Execution and rendering require observing the initial world rather than simply moving around. Step-level instructions force the model to ground each action in the observed world.

• **Physical Simulation (L3).** Models should depict how an event unfolds from the initial state according to physical laws or real-world dynamics, connecting observations with physical knowledge.

• **Adaptive Task Completion (L3).** Models must decompose the goal without step-level instructions, infer feasible intermediate actions, and ground their plans in the changing visual state.

• **Visual Cues (L4).** Text alone is sometimes insufficient to specify location or transitions. Direct or indirect visual cues combined with text instructions indicate where or how the world should change.

• **Physical Properties (L4).** Observations may include material properties and object states without fully verbalizing them. The model must infer the operative property, bind it to the correct entities, and carry its consequences through a new multi-state simulation.

• **Visual Goal Realization (L4).** The desired outcome is specified visually, either as a final state or hints of the final state. The model must identify the goal semantics, plan how to achieve the final state, and synthesize the intervening steps.

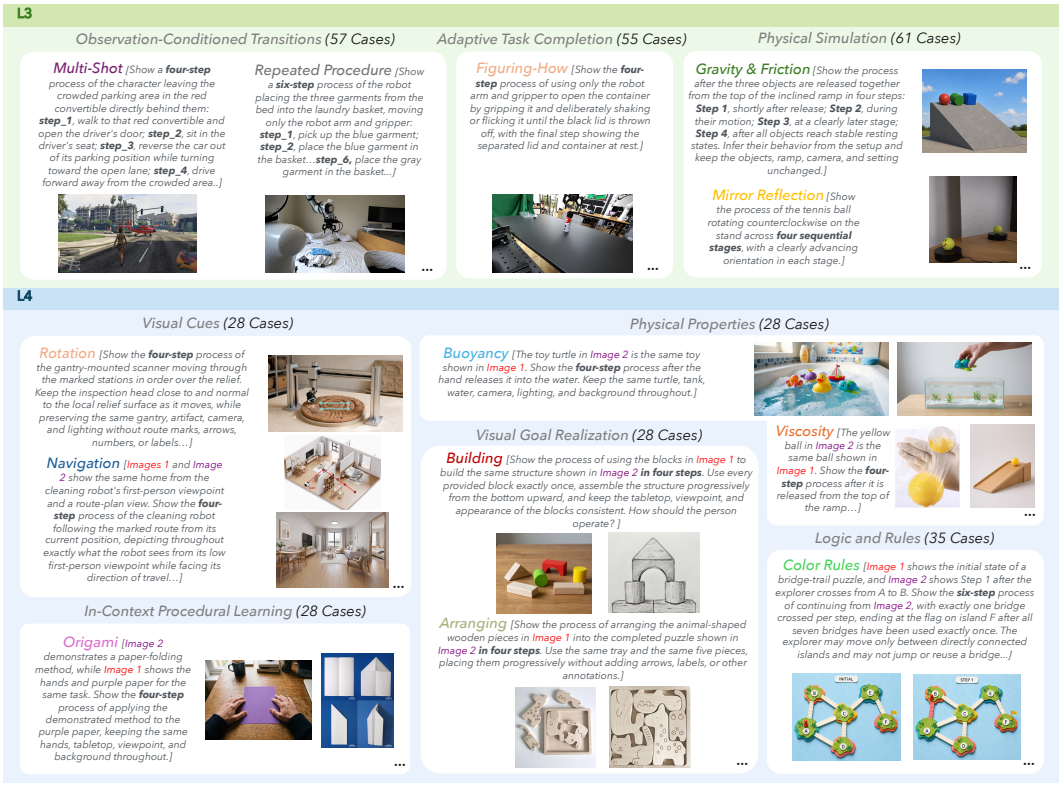


Figure 3: Representative examples of the eight WORLDCAST categories. The benchmark comprises 320 manually designed cases in total, with 1,352 requested visual states.

• **Logic and Rules (L4).** Several historical states and textual cues jointly define a transition rule. The model must combine textual reasoning with visual rules, induce the operation for successive states and render it consistently.

• **In-Context Procedural Learning (L4).** Text and images describe prior experience in the form of an illustrated procedure. The model must identify the function of each step, abstract the procedure from its original scene, and transfer it to a new world rather than imitate the examples superficially.

2.1 EVALUATION METRICS

2.1.1 EVALUATION DIMENSIONS

We evaluate each generated sequence along three complementary dimensions: Task Completion (TC), Causal Plausibility (CP), and Visual Consistency (VC).

Task Completion (TC): TC measures whether the sequence visibly realizes the user's intended task. It is computed from the sequence-level sub-metric **Progress Alignment (PA)** and the outcome-level sub-metric **Outcome Completion (OC)**. The outcome weight ranges from 0 to 0.5; at zero, only PA is evaluated. Calculation details are in Appendix D.

Causal Plausibility (CP): CP evaluates whether adjacent transitions and longer-range dependencies follow the applicable physical laws, logical rules, or task-specific world dynamics. It penalizes unsupported changes such as impossible transformations or reversed cause and effect.

Visual Consistency (VC): VC evaluates whether visual properties that should persist—including object identity, count, geometry, scale, material, camera, and background—remain stable across the sequence. Changes required by the task are exempt; unrelated appearance, disappearance, duplication, deformation, or drift is penalized.

We aggregate TC, CP, and VC derived from discrete ratings on a 0–3 scale, then convert the result to a 100-point score:

$$\text{Final} = \text{Rescale}(\alpha_1 \,\text{TC} + \alpha_2 \,\text{CP} + \alpha_3 \,\text{VC}). \tag{2}$$

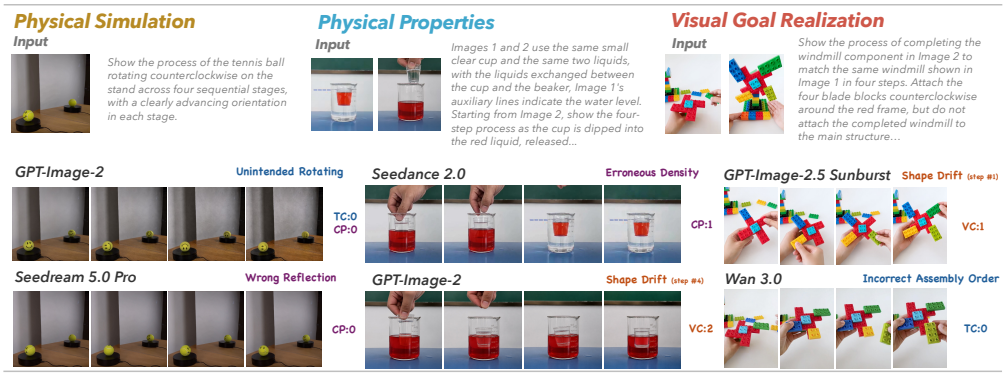


Figure 4: Qualitative examples on WORLDCAST. Models exhibit distinct failure patterns across the three evaluation dimensions; all scores are reported on a 0–3 scale.

3 EVALUATION PROTOCOL

Agentic Video Adapter. To evaluate image and video models under the same sequence-level protocol, we introduce an agentic video adapter that converts each generated video into an ordered set of task-aligned keyframes. The adapter follows an inclusion-first principle, selecting frames that cover as many critical states as possible while preserving their temporal constraints. In practice, we adopt GPT-6 Astra (OpenAI, 2026b) through Codex as the agentic video adapter. The adapter is used only to determine whether the video contains the required key states; it does not assess the plausibility of the full video sequence or its visual quality.

LMM-as-a-Judge. We use GPT-5.5 as the automatic evaluator. Given the task instruction, multimodal references, judge-only hints, and the ordered output sequence, the judge scores Progress Alignment (PA) and Outcome Completion (OC) to compute Task Completion (TC), together with Causal Plausibility (CP) and Visual Consistency (VC). The same evaluation protocol is applied to image-model outputs and video keyframes selected by the agentic video adapter. Full scoring criteria and aggregation rules are provided in Appendix D.

Human-Annotated Hints for Judges. Because we evaluate entire output sequences and some tasks are partially open-ended, judging can involve substantial reasoning. To improve evaluation consistency and accuracy, we introduce task-specific judge-only hints. Human annotators provide manual hints that identify task-critical objects, milestones, and constraints, and highlight deterministic steps the judge should not overlook. Appendix D.1 provides further details.

4 EXPERIMENTS

4.1 SETTINGS

We evaluate twelve representative image and video generation models of three types across all eight WORLDCAST categories:

• **Open-source Image Models.** The suite comprises unified models Emu3.5 (Cui et al., 2025), BAGEL (Deng et al., 2025), SenseNova-U1 (Diao et al., 2026) and an image diffusion model supporting multi-reference generation, Qwen-Image (Wu et al., 2025).

• **Proprietary Image Systems.** Proprietary image models include GPT-Image-2 (OpenAI, 2026c), GPT-Image-2.5 Sunburst (OpenAI, 2026d), Nano Banana Pro (Google DeepMind, 2025), Seedream 4.5 (ByteDance Seed, 2025), and Seedream 5.0 Pro (ByteDance Seed, 2026).

• **Video Systems.** We additionally evaluate three video generation systems: Seedance 2.0 (Team Seedance, 2026), Wan 3.0 (Alibaba Cloud, 2026), and Gemini Omni (Google DeepMind, 2026b).

Because WORLDCAST requires an ordered sequence of visual states, we adapt the inference format to each model's native output interface. Unless otherwise specified, GPT-Image-2 refers to interleaved generation through the Responses API paired with GPT-5.5 (OpenAI, 2026a), while GPT-

Table 1: Overview of WORLDCAST performance across twelve image and video generation models and eight benchmark categories. Blue cells mark the best result within each model group, while orange cells mark the best result across all models. Icons indicate the generation format: Interleaved Generation, Grid Generation, Video Generation, Plan & Iterate.

Method	L3 Tasks				L4 Tasks						Overall
	Observation-Conditioned Transitions	*Physical Simulation*	*Adaptive Task Completion*	**L3 Overall**	*Physical Properties*	*Visual Cues*	*In-Context Procedural Learning*	*Logic and Rules*	*Visual Goal Realization*	**L4 Overall**	
Open-source Image Models											
SenseNova-U1	**24.1**	**24.6**	**24.7**	**24.4**	**19.9**	**19.1**	**20.1**	**5.9**	**12.2**	**15.0**	**20.1**
Emu3.5	17.4	13.8	18.5	16.5	6.6	10.3	8.1	4.8	7.6	7.4	12.3
BAGEL	4.9	7.0	8.3	6.7	8.5	12.5	7.0	2.9	4.0	6.8	6.8
Qwen-Image	7.8	6.8	6.2	6.9	4.0	6.9	6.1	0.6	3.4	4.0	5.6
Proprietary Image Systems											
GPT-Image-2.5	**85.6**	69.6	**66.1**	**73.8**	**73.6**	**76.7**	**71.6**	**74.3**	**61.3**	**71.6**	**72.8**
GPT-Image-2	74.2	**70.6**	59.6	68.3	66.0	62.6	70.6	66.6	51.8	63.7	66.2
Nano Banana Pro	36.2	56.1	43.5	45.6	49.9	33.7	43.3	41.4	28.5	39.5	42.8
Seedream 5.0 Pro	30.3	45.9	33.4	36.8	46.7	27.7	38.9	11.9	29.1	29.9	33.6
Seedream 4.5	19.4	14.9	18.7	17.6	19.7	16.0	21.4	3.4	12.8	14.1	16.0
Video Systems											
Seedance 2.0	**53.3**	**53.4**	**48.7**	**51.9**	40.8	**56.6**	**55.4**	**17.7**	**43.3**	**41.6**	**47.2**
Gemini Omni	45.6	50.2	41.0	45.8	**47.2**	42.7	49.9	16.2	33.0	36.8	41.6
Wan 3.0	51.9	47.3	38.5	46.0	42.7	42.7	35.5	17.2	29.6	32.8	39.9

Image-2.5 refers to the Sunburst version paired with GPT-6 Astra (OpenAI, 2026b). Both language models were the respective flagship models at the time of the image models' releases. We provide generation templates for each output mode in Appendix H: image models with interleaved generation capabilities directly produce the requested sequence; models without this capability produce a grid. For BAGEL, we adopt the Plan & Iterate scheme following UEval (Li et al., 2026a).

4.2 MAIN RESULTS

Frontier unified systems exhibit early-stage L4-level visual state prediction capabilities. As shown in Table 1, the strongest proprietary image model GPT-Image-2.5 performs well on L3 tasks and scores above 60 on all five L4 tasks, indicating that frontier models already reveal an emerging ability to simulate how a world should evolve based on contextual information, and this ability is evolving rapidly. Nevertheless, even the strongest model does not consistently succeed across categories, achieving only 61.3 in the task of Visual Goal Realization, indicating that context-grounded world prediction remains a capability that has yet to be fully mastered by generative systems.

WORLDCAST reveals substantial performance gaps between models that widen at higher levels. The best open-source system trails substantially behind the strongest proprietary model in every category, and the gap generally widens on L4 tasks. For example, SenseNova-U1, the best-performing open-source model, drops from 24.4 on L3 to 15.0 on L4, whereas GPT-Image-2.5 declines by only 2.2 points, maintaining a score of 71.6 on L4. The strongest model even performs better on some complex tasks, whereas weaker models struggle. For example, GPT-Image-2.5 scores 76.7 on Visual Cues and 74.3 on Logic and Rules, whereas Nano Banana Pro scores just 33.7 and 41.4, respectively, both below its overall score. This pattern suggests that, beyond a certain level of capability, models may benefit from the additional information provided by more complex inputs. Open-source unified models outperform the generation-only baseline overall, suggesting a need for understanding and reasoning. Taken together, these results suggest that strong performance on WORLDCAST requires broad knowledge and generalizable generative intelligence.

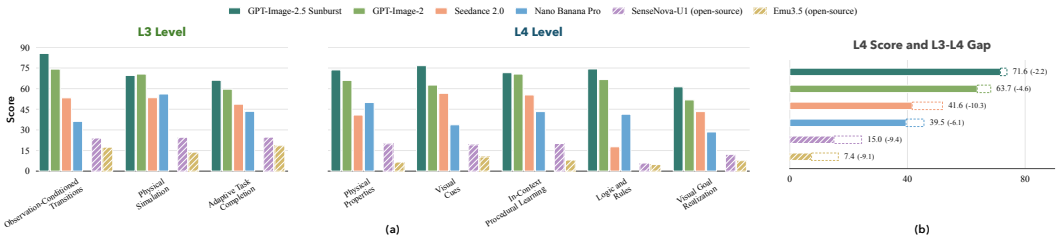


Figure 5: Performance comparison on WORLDCAST. (a) Category-wise scores across L3 and L4. (b) L4 scores, with dashed extensions and labels showing the absolute gap from L3.

Performance varies across task categories but generally degrades on reasoning-intensive tasks. Models exhibit different weaknesses across tasks. Video models struggle particularly with L4 Logic & Rules. Within L3, both GPT-Image models, Wan 3.0, and Gemini Omni score lowest on Adaptive Task Completion. Within L4, Visual Goal Realization is the weakest category for both GPT-Image 2.5 and Nano Banana Pro, with the former achieving 61.3 and the latter only 28.5. These patterns highlight the difficulty of aligning generation with temporal reasoning and abstract information.

Visual continuity does not necessarily translate into accurate key states. When tasks are evaluated through a sequence of semantically meaningful states, frontier image models that are capable of interleaved generation are surprisingly competitive with video models, although Seedance 2.0 achieves the highest overall score among models outside the GPT-Image family. Since interleaved generation also supports training on video sequences, this raises the possibility that these image models benefit from both high-quality video training data and linguistic reasoning.

4.3 Discussion and Analysis

4.3.1 Analysis of the language model enhancer

The main results show that the GPT-Image family substantially outperforms other model families on this benchmark. One plausible explanation is the contribution of the paired language model, whose extensive world knowledge and understanding capabilities enable it to produce enhanced, information-dense prompts. For analysis, we integrate Nano Banana Pro with GPT-6 and find that this combination substantially improves performance across L3-L4 task categories but still lags behind the GPT-Image family with gains varying across tasks, as shown in Figure 6. When we replace GPT-5.5 with GPT-6 as the enhancer for the previous-generation GPT-Image-2, the overall score increases marginally with a decline in Physical Simulation. These results highlight the importance of the intelligent enhancer with broad knowledge, while also suggesting that effective guidance in world state prediction requires a strong visual generator and a closely matched enhancer.

4.3.2 Analysis of interleaved generation

Interleaved generation outperforms single-image grids but does not fully explain performance gaps. Interleaved generation allows the language model to guide successive visual steps, potentially reducing the reasoning burden on the renderer compared with generating all steps within a single image grid. Switching from interleaved generation to grid output reduces GPT-Image-2.5's overall score by 22.7 points, with the most obvious drop in L4 Logic & Rules, from 74.3 to 23.0. However, output format alone does not explain the performance gap across model families: GPT-Image-2.5 with grid output still outperforms interleaved Nano Banana Pro overall, and Nano Banana Pro retains much of its performance under grid output, surpassing both Seedream 5.0 Pro and Seedream 4.5 with the same generation scheme in Table 1. These results suggest that interleaving is a valuable capability based on underlying visual generative performance.

Interleaved generation faces the challenge of accumulated context. While interleaving enables step-by-step guidance, it also introduces a growing context burden, challenging both the language enhancer and the renderer. To investigate this challenge, we analyze step-wise accuracy over 195 four-step cases (excluding cases that provide a direct endpoint reference), using the designated-

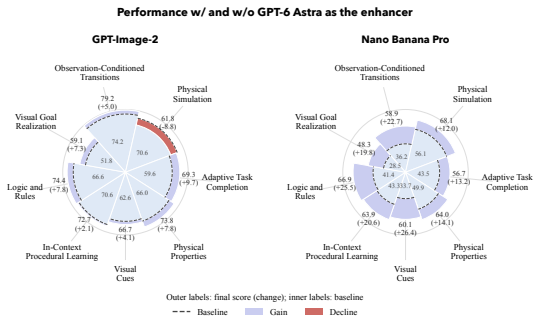


Figure 6: Category-wise scores and gains with GPT-6 Astra as the enhancer for two generators.

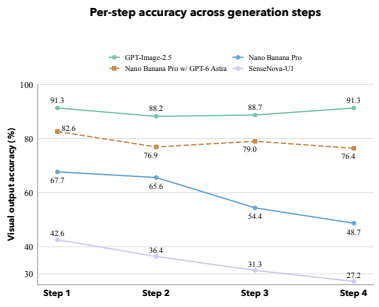


Figure 7: Visual output accuracy across four generation steps.

Table 2: Interleaved versus grid generation. Switching to grid generation reduces overall performance for both models, with a much larger drop for GPT-Image-2.5.

Model	Generation Schema	Δ Baseline	*Observation-Conditioned Transitions*	*Physical Simulation*	*Adaptive Task Completion*	*Physical Properties*	*Visual Cues*	*In-Context Procedural Learning*	*Logic and Rules*	*Visual Goal Realization*	Overall
GPT-Image-2.5	Interleaved	–	85.6	69.6	66.1	73.6	76.7	71.6	74.3	61.3	72.8
GPT-Image-2.5	Grid	Interleaved	57.8 (−27.8)	61.8 (−7.8)	48.2 (−17.9)	51.0 (−22.6)	46.0 (−30.7)	60.5 (−11.1)	23.0 (−51.3)	39.6 (−21.7)	50.1 (−22.7)
Nano Banana Pro	Interleaved	–	36.2	56.1	43.5	49.9	33.7	43.3	41.4	28.5	42.8
Nano Banana Pro	Grid	Interleaved	39.8 (+3.6)	43.2 (−12.9)	42.0 (−1.5)	53.1 (+3.2)	42.9 (+9.2)	41.4 (−1.9)	31.6 (−9.8)	29.9 (+1.4)	40.6 (−2.2)

step evaluation protocol detailed in Appendix H.3. Weaker models exhibit visible degradation: SenseNova-U1 falls from 42.6 to 27.2 (−15.4). In contrast, GPT-Image-2.5 remains relatively stable. The default Nano Banana Pro declines from 67.7 to 48.7 (−19.0). Equipping it with GPT-6 Astra reduces the first-to-last-step drop, from 82.6 to 76.4 (−6.2), although accuracy fluctuates across intermediate steps. These results suggest that maintaining reliable generation as context accumulates remains an important challenge for unified systems.

4.3.3 Towards agentic visual generation with tools: gains and challenges

The preceding experiments show that frontier visual generative models can increasingly extract and use visual information, enabling strong language models to apply the broad world knowledge they have acquired largely from digital worlds such as web pages and code repositories. Still, even strong systems struggle with L4-level tasks that require reasoning across successive stages of a process.

This raises a natural question: can a powerful language agent externalize its world knowledge when modeling real-world tasks through agentic inspection-generation loops and code-based tools? To investigate this question, we build two simple agentic harnesses based on Codex (OpenAI, 2025a) with the same generation budget. The first encourages the agent to perform agentic inspect-and-revise loops while prohibiting coding tools, whereas the second encourages the agent to use coding tools freely to construct intermediate artifacts or visual guidance.

As shown in Table 3, iterative agentic refinement directly raises GPT-Image-2.5 from 72.8 to 80.2 and GPT-6-enhanced Nano Banana Pro from 61.2 to 72.5. Coding tools further push the highest overall score to 82.4, but with less uniform gains across models and domains. The coding harness yields particularly large improvements on L4 Visual Goal Realization and Logic & Rules, and L3 Physical Simulation, where visual coding makes physical or logical intermediate states explicit. Coding artifacts can remain useful even when they do not directly enter the visual generator. Figure 8 illustrates an example trajectory, where the agent first constructs a coded simulation but decides not to use it as a direct visual reference, instead using it as prompt guidance to generate the correct first step. It then inspects the generated sequence and finds the erroneous step, uses the simulated result



Figure 8: Visual coding as procedural guidance. Left: Result comparisons with different model harnesses. Right: code-generated intermediate states guide refinements of a towel-folding sequence.

Table 3: Comparisons of harnesses for agentic visual generation using GPT-6 Astra as the language model. The inspection loop encouraging coding tools yields the best overall performance.

Model Paired with GPT-6 Astra	Harness Enhancement	*Observation-Conditioned Transitions*	*Physical Simulation*	*Adaptive Task Completion*	*Physical Properties*	*Visual Cues*	*In-Context Procedural Learning*	*Logic and Rules*	*Visual Goal Realization*	**Overall**
GPT-Image-2.5	- (Responses API, baseline)	85.6	69.6	66.1	73.6	**76.7**	71.6	74.3	61.3	72.8
	Inspection loop	**92.4** (+6.8)	86.2 (+16.6)	73.2 (+7.1)	**84.4** (+10.8)	69.5 (−7.2)	72.9 (+1.3)	81.1 (+6.8)	69.2 (+7.9)	80.2 (+7.4)
	Coding + inspection loop	90.7 (+5.1)	**88.6** (+19.0)	**74.9** (+8.8)	80.1 (+6.5)	74.5 (−2.2)	**74.3** (+2.7)	**85.7** (+11.4)	**81.3** (+20.0)	**82.4** (+9.6)
Nano Banana Pro	- (Responses API, baseline)	58.9	68.1	56.7	64.0	60.1	63.9	66.9	48.3	61.2
	Inspection loop	74.7 (+15.8)	82.1 (+14.0)	65.3 (+8.6)	77.2 (+13.2)	68.6 (+8.5)	69.8 (+5.9)	74.3 (+7.4)	60.6 (+12.3)	72.5 (+11.3)
	Coding + inspection loop	78.1 (+19.2)	82.0 (+13.9)	65.8 (+9.1)	72.1 (+8.1)	63.3 (+3.2)	66.7 (+2.8)	81.9 (+15.0)	62.3 (+14.0)	72.9 (+11.7)

as a reference but obtains implausible towels, and eventually finds a solution by working backward from the final-state reference.

While the result improves on the baselines, this trajectory reveals that incorporating complex reference information still places a considerable burden on the generator and may also help explain the uneven gains from coding, since additional guidance can sometimes fail to take effect or degrade performance. As coding agents grow more capable almost by the day, this trend presents both opportunities and challenges for future visual generative systems: they should be able to accept heterogeneous artifacts as guidance from the agent and dynamically interpret and use them, rather than treating them merely as pixel-level anchors.

4.4 Validity of the Agentic Video Adapter and LMM-as-a-Judge

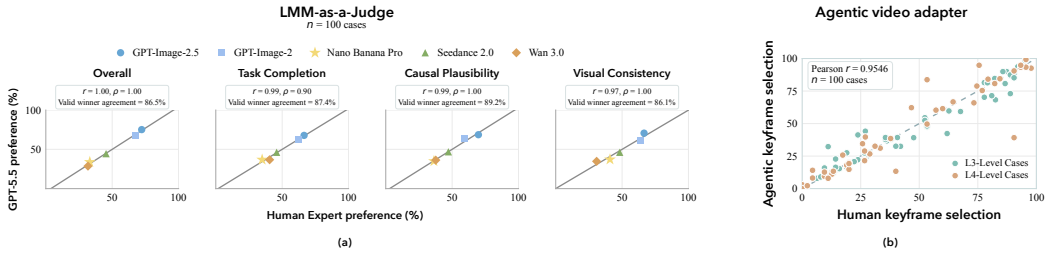


Figure 9: Validation of the automatic evaluation pipeline. (a) Correlation between human and GPT-5.5 win rates. Scores (win = 1, tie = 0.5, loss = 0) are averaged across pairwise comparisons for each model. (b) Correlation between evaluator scores from agent-selected and human-selected keyframes.

We validate both *LMM-as-a-Judge* and *Agentic video adapter*. For the *LMM-as-a-Judge*, following VBench (Huang et al., 2024), six human experts perform blind pairwise comparisons among five representative systems. As shown in Figure 9(a), WorldCast and human win rates achieve Pearson $r \geq 0.97$ across four dimensions (Overall, TC, CP, and VC), with winner agreement rates $\geq 86\%$ among valid human-majority winner votes. For the *agentic video adapter*, three annotators select keyframes for 100 videos generated on our benchmark. We score human- and adapter-selected keyframes using the same evaluator. The resulting scores achieve a Pearson correlation of 0.9546 (Figure 9(b)), supporting the reliability of the automated protocol.

5 Conclusion

In this work, we present WorldCast, a benchmark for exploring world state prediction as an emerging capability of visual generative models. WorldCast evaluates L3–L4 capabilities based on a cognitively motivated four-level hierarchy, through 320 manually designed cases across eight categories. Experiments on image and video systems with a shared evaluation interface show that frontier unified models exhibit early-stage world state prediction capabilities on real-world tasks, while substantial gaps remain across models and become more pronounced when tasks require heterogeneous context and tightly coupled reasoning and generation. Our analyses further identify potential performance gains from interleaved generation, stronger language-model reasoning, and iterative agentic refinement for this task, while highlighting challenges that emerge as models advance in these directions. We hope that WorldCast will serve as a useful testbed for developing unified generative models that can integrate heterogeneous context, agentic reasoning and visual generation to predict coherent world evolution.

AI USE STATEMENT

Generative AI is integral to the research method studied in this work: the evaluation protocol employs an LMM judge (GPT-5.5) and an agentic video adapter to score model outputs (Section 3), and the enhancement experiments use GPT-6 Astra with the Codex coding harness to iteratively refine generations (Section 4.3.3). In addition, we used generative AI tools for (1) language editing and manuscript polishing, including improvements to grammar, clarity, and readability, and (2) generating synthetic datasets, using image-generation tools to synthesize and augment benchmark cases (Appendix C). The authors take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work presents an evaluation benchmark and does not introduce new generative capabilities. Benchmark cases are drawn from public datasets, online videos, instruction manuals, and gameplay footage, or synthesized with image-generation tools, and all cases were manually reviewed by the authors before inclusion. Generative tools used to synthesize cases may introduce artifacts or biases, which manual curation mitigates but cannot fully eliminate. The human expert preferences used to validate the automatic evaluator describe the participating experts and the evaluated systems, and should not be treated as a universal standard of visual generation quality.

REPRODUCIBILITY STATEMENT

Section 3 specifies the evaluation protocol, including the agentic video adapter and the LMM-as-a-judge procedure. Appendix B formalizes the task, Appendix C describes data collection, and Appendix D provides the complete scoring rubrics, judge hints, and aggregation rules. Validation of the automatic evaluation pipeline is presented in Section 4.4 and Appendix E, and the shared generation and evaluation prompt templates are provided in Appendix H. Together with the per-category scores reported in the main tables, these specifications make the reported comparisons and the scoring procedure explicit. We will publicly release all benchmark data, evaluation code, and prompt templates.

REFERENCES

Alibaba Cloud. Wan 3.0: Leading AI video generation model. Model release, 2026. URL https://github.com/AlibabaCloud-Official/Wan3.0.

Ruichuan An, Sihan Yang, Ziyu Guo, Wei Dai, Zijun Shen, Haodong Li, Renrui Zhang, Xinyu Wei, Guopeng Li, Wenshan Wu, and Wentao Zhang. GENIUS: Generative fluid intelligence evaluation suite. *arXiv preprint arXiv:2602.11144*, 2026. URL https://arxiv.org/abs/2602.11144.

Anthropic. Introducing Claude 4. Anthropic, May 2025. URL https://www.anthropic.com/news/claude-4.

Yuxuan Bian, Xin Chen, Zenan Li, Tiancheng Zhi, Shen Sang, Linjie Luo, and Qiang Xu. Video-as-prompt: Unified semantic control for video generation. *arXiv preprint arXiv:2510.20888*, 2025.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024. URL https://arxiv.org/abs/2402.15391.

ByteDance Seed. Seedream 4.5. Official model page, 2025. URL https://seed.bytedance.com/en/seedream4_5.

ByteDance Seed. Seedream 5.0 Pro. Official model page, 2026. URL https://seed.bytedance.com/en/seedream5_0_pro.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. URL https://arxiv.org/abs/2405.09818.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, et al. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Dima_Damen_Scaling_Egocentric_Vision_ECCV_2018_paper.html.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, et al. Sensenova-U1: Unifying multimodal understanding and generation with NEO-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Google. Nano banana: Image editing in google gemini gets a major upgrade. Google Blog, August 2025. URL https://blog.google/products/gemini/updated-image-editing-model/.

Google DeepMind. Introducing nano banana pro. Google Blog, November 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/.

Google DeepMind. Gemini 3.1 Pro. Google DeepMind model card, February 2026a. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Google DeepMind. Gemini omni: Create anything from any input. Google DeepMind, 2026b. URL https://deepmind.google/models/gemini-omni/.

Hang Hua, Ziyun Zeng, Yizhi Song, Yunlong Tang, Liu He, Daniel Aliaga, Wei Xiong, and Jiebo Luo. MMIG-bench: Towards comprehensive and explainable evaluation of multimodal image generation models. *arXiv preprint arXiv:2505.19415*, 2025.

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

Dongzhi Jiang, Renrui Zhang, Haodong Li, Zhuofan Zong, Ziyu Guo, Jun He, Claire Guo, Junyan Ye, Rongyao Fang, Weijia Li, Rui Liu, and Hongsheng Li. DraCo: Draft as CoT for text-to-image preview and rare concept generation. *arXiv preprint arXiv:2512.05112*, 2025. URL https://arxiv.org/abs/2512.05112.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Kuaishou Technology. Kling AI launches 3.0 model, ushering in an era where everyone can be a director. Official release, February 2026. URL https://ir.kuaishou.com/news-releases/news-release-details/kling-ai-launches-30-model-ushering-era-where-everyone-can-be.

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

Haodong Li, Chunmei Qing, Huanyu Zhang, Dongzhi Jiang, et al. CoCo: Code as CoT for text-to-image preview and rare concept generation. *arXiv preprint arXiv:2603.08652*, 2026b.

Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, Ruichuan An, Dongzhi Jiang, Boxue Yang, Ji Xie, Xu Huang, Wenhao Yan, Jialv Zou, Zhengrong Yue, Yaxin Luo, Xiaotong Li, Yuzhu Wang, Junyan Ye, Jinjing Zhao, Zehui Chen, Lin Chen, Renye Yan, Feng Zhao, and Pheng-Ann Heng. VideoCoCo: Code-as-CoT for physically-consistent video generation via an agentic dual-engine system. *arXiv preprint arXiv:2607.27380*, 2026c. doi: 10.48550/arXiv.2607.27380. URL https://arxiv.org/abs/2607.27380.

Wenxue Li, Peiyan Guan, Haoyang Jiang, Junxian Cai, et al. OmniVBench: A benchmark and large-scale dataset for omni reference-to-video generation. *arXiv preprint arXiv:2609.22069*, 2026d.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Fanqing Meng, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: World knowledge-informed semantic evaluation for text-to-image generation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2503.07265.

OpenAI. Introducing codex. OpenAI, May 2025a. URL https://openai.com/index/introducing-codex/.

OpenAI. Introducing our latest image generation model in the API. OpenAI, April 2025b. URL https://openai.com/index/image-generation-api/.

OpenAI. Sora 2. Official model announcement, 2025c. URL https://openai.com/index/sora-2/.

OpenAI. GPT-5.5 model. OpenAI API documentation, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.5.

OpenAI. GPT-6 Astra model. OpenAI API documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-6-astra.

OpenAI. GPT-Image-2 model. OpenAI API documentation, 2026c. URL https://developers.openai.com/api/docs/models/gpt-image-2.

OpenAI. GPT-Image-2.5 Sunburst. OpenAI API documentation, 2026d. URL https://developers.openai.com/api/docs/models/gpt-image-2.5-sunburst.

Team Seedance. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Team Seedream. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025. URL https://arxiv.org/abs/2509.20427.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024. URL https://arxiv.org/abs/2408.14837.

Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. MS-Diffusion: Multi-subject zero-shot image personalization with layout guidance. *arXiv preprint arXiv:2406.07209*, 2024. URL https://arxiv.org/abs/2406.07209.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024. URL https://arxiv.org/abs/2410.13848.

Mingrui Wu, Hang Liu, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. MICON-bench: Benchmarking and enhancing multi-image context image generation in unified multimodal models. *arXiv preprint arXiv:2602.19497*, 2026.

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. URL https://arxiv.org/abs/2408.12528.

Wenhao Yan, Fengjia Guo, Zhuoyi Yang, and Jie Tang. SCAIL-2: Unifying controlled character animation with end-to-end in-context conditioning. *arXiv preprint arXiv:2606.10804*, 2026. URL https://arxiv.org/abs/2606.10804.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

Junyan Ye, Jun He, Zilong Huang, Dongzhi Jiang, Xuan Yang, Rui Chen, and Weijia Li. GenClaw: Code-driven agentic image generation. *arXiv preprint arXiv:2605.30248*, 2026.

Z.ai. GLM-5.3-Flash/FlashX. Z.AI Developer Documentation, 2026. URL https://docs.z.ai/guides/vlm/glm-5.3-flash.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026a.

Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-Image-Agent: Bridging the context gap in real-world image generation. *arXiv preprint arXiv:2606.26907*, 2026b. URL https://arxiv.org/abs/2606.26907.

Yubo Zhu, Yawen Shao, Ziyun Dai, Zixun Fang, Kai Zhu, Siyang Sun, Haolan Xue, Chuxin Wang, Tingyu Weng, Jingming Luo, Chen Shi, Lianghua Huang, Yufeng Ai, Yuzheng Wang, Wenyuan Zhang, Yu Shang, Yuxiang Bao, Zoubin Bi, Jie Xiao, Jinbo Xing, Jiaxing Zhao, Chongyang Zhong, Hengjian Chen, Chenwei Xie, Akide Liu, Zhehan Kan, Yu Liu, Wei Zhai, Sheng Zhong, and Wei Tong. WanPE: Towards cinematic prompt enhancement for modern text-to-video generation. *arXiv preprint arXiv:2609.30221*, 2026. URL https://arxiv.org/abs/2609.30221.

APPENDIX

A RELATED WORK

World Modeling. World models predict how an environment evolves under actions. Genie learns action-controllable video dynamics (Bruce et al., 2024), and GameNGen demonstrates real-time game simulation (Valevski et al., 2024), both emphasizing temporal continuity. WORLDCAST instead focuses on key visual states to evaluate the intelligence of generative systems. Although these world models support interactive rollouts, they do not currently support the reasoning-based transitions and heterogeneous references required by our L3-L4 tasks and are thus excluded from our benchmark.

Unified Multimodal Systems. Unified multimodal models (UMMs) bring understanding and generation into a shared interface. Chameleon explores mixed-modal early fusion (Chameleon Team, 2024), while Show-o (Xie et al., 2024) and Janus (Wu et al., 2024) develop unified architectures for understanding and generation. BAGEL (Deng et al., 2025), Emu3.5 (Cui et al., 2025), and SenseNova-U1 (Diao et al., 2026) further support interleaved multimodal inputs and outputs. In a broader sense, generative systems that connect language-side reasoning with visual generation through a shared interface can also be considered unified even without a unified architecture, for example, a dedicated prompt enhancer paired with an image or video generation model (Team Seedream, 2025; Zhu et al., 2026). WORLDCAST evaluates model performance regardless of the underlying unification approach.

Multimodal References for Generation. Multimodal references can guide appearance and identity (Ye et al., 2023; Wang et al., 2024), motion (Yan et al., 2026), or provide code-generated drafts, as in DraCo (Jiang et al., 2025), CoCo (Li et al., 2026b), and VideoCoCo (Li et al., 2026c). Such references typically provide relatively direct guidance on a specific aspect of the output. WORLDCAST instead focuses on complex reference semantics, including abstract goals, prior procedural experience, physical properties, and visual rules, which must be interpreted and integrated to infer how the world should evolve across successive states.

Evaluation of Unified Systems. Existing evaluations increasingly move beyond fidelity and shallow text–image alignment. WISE evaluates world-knowledge-informed text-to-image semantics (Niu et al., 2026), GENIUS studies pattern induction and constraint execution (An et al., 2026), UEval evaluates interleaved image–text responses (Li et al., 2026a), and MICON-Bench and MMIG-Bench study multi-image context (Wu et al., 2026; Hua et al., 2025). Some tasks in WISE and GENIUS approach L3 reasoning, but primarily assess a static visual result. WORLDCAST instead evaluates ordered world evolution from successive visual states.

B TASK FORMULATION

WORLDCAST formulates context-grounded visual world prediction as generating an ordered sequence of visual states from interleaved image–text context. Each test case consists of a natural-language instruction p, one or more input images $\mathcal{X} = (x_m)_{m=1}^{M}$, and an ordered sequence of target output images $\mathcal{Y} = (y_k)_{k=1}^{K}$. The input images may serve different functions—for example, specifying an initial state, a goal, an object or scene reference, a physical property, a visual rule, or an intermediate demonstration. Given the multimodal context $(p, \mathcal{X})$, a generative model G is required to produce

$$\widehat{\mathcal{Y}} = G(p, \mathcal{X}) = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K), \tag{3}$$

where the generated images must jointly satisfy the instruction and visual constraints while forming a temporally ordered, causally coherent process. This multimodal-context-to-visual-sequence formulation provides a common interface for different model families: an image model may generate the sequence jointly or through interleaved generation, whereas a video model may produce a continuous clip whose selected frames constitute $\widehat{\mathcal{Y}}$.

C Data Collection

We combine real-world datasets, online videos, illustrated manuals, gameplay footage, and manually designed scenarios. When suitable visual material is unavailable, we synthesize or enhance analogous cases with image-generation tools (Google, 2025; Meituan LongCat Team et al., 2025; OpenAI, 2025b) while preserving the intended spatial, physical, and procedural constraints. The resulting output-step distribution is summarized in Figure 10. The category-specific collection procedures are as follows:

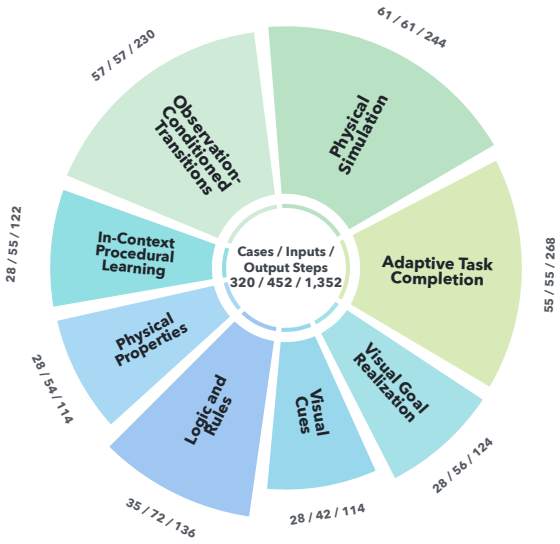


Figure 10: Data distribution across the eight WorldCast categories. Labels report cases/input images/requested visual states.

• **Observation-Conditioned Transitions.** We collect cases from egocentric and game data (Damen et al., 2018; Khazatsky et al., 2024). We retain scenarios in which the subsequent action logic must be grounded in observations from the initial frame.

• **Physical Simulation.** We source physical events from game and egocentric data, science and educational videos, and filtered Internet examples. We further synthesize analogous scenarios based on real-world experience, retaining cases whose evolution requires reasoning from the observed initial state.

• **Adaptive Task Completion.** Most cases are collected from game and egocentric data; additional everyday tasks are designed and synthesized from common real-life experience.

• **Visual Cues.** We construct cues by drawing or manually marking images and by annotating collected photographs. Cases may contain either a single reference or multiple references. For multi-view cases, we collect home-design views and use Claude Opus (Anthropic, 2025) to operate Blender to create geometrically matched source images, which are then enhanced with Nano Banana Pro for greater realism while preserving their underlying 3D consistency.

• **Physical Properties.** We select examples from science videos and targeted Internet searches, and synthesize additional cases grounded in physical phenomena encountered in everyday experience.

• **Visual Goal Realization.** We extract initial and target states from assembly and building videos on YouTube, occasionally enhancing the target image for clarity and visual quality.

• **Logic and Rules.** We draw cases from existing games and use Claude Opus (Anthropic, 2025) to synthesize additional games with controlled visual rules.

• **In-Context Procedural Learning.** We collect illustrated manuals and pair their procedures with corresponding or analogous initial states, including cases that require transferring a procedure across visually different situations.

D Evaluation Details

D.1 Details on Human Hints

Each test case includes a set of human-written hints for evaluation. These hints are provided only to the judge and are never exposed to the generation model. They reduce ambiguity in otherwise underspecified instructions and draw the judge's attention to task-critical evidence that may appear only at a particular stage of the generated sequence.

Annotators write the hints after jointly inspecting the instruction, all multimodal references, and the expected progression. A hint may identify task-critical objects and their reference bindings, required intermediate milestones and temporal ordering, or physical and logical constraints that must remain valid across states. When a task admits multiple valid procedures, the hints specify the indispensable outcome or constraint without prescribing unnecessary appearance details. The same case-specific hints are used to evaluate every model output. They guide the assessment of PA, OC, CP, and VC, but do not substitute for visible evidence in the generated sequence. The same Task Completion (TC) hints are provided to both Progress Alignment (PA) and Outcome Completion (OC) evaluations. Figure 14 provides examples of these human-annotated hints.

D.2 Evaluation Procedure

For every case, the judge is given the natural-language instruction, any judge-only hints, all reference and initial images, and the complete ordered sequence of generated images. The judge first reads the full sequence, evaluates the relevant local transitions, and then performs a global pass to resolve dependencies that may not be apparent from an isolated frame. Credit is based on visible evidence; captions, arrows, or a plausible narrative do not by themselves establish that a state change occurred.

D.3 Task Completion

We compute TC from Progress Alignment (PA) and Outcome Completion (OC). PA evaluates the complete ordered sequence on a three-level scale. A score of 2 indicates that the intended process and all indispensable milestones are visibly realized in the required or logically necessary order. A score of 1 indicates that the core process is realized with only secondary defects, and a score of 0 indicates that at least one task-critical milestone is missing, contradictory, unsupported, or ordered in a way that breaks the process. A correct-looking endpoint does not compensate for a missing critical step.

OC evaluates the final generated state. It receives 1 when every indispensable endpoint requirement is visibly satisfied and 0 otherwise. After normalizing PA to $[0, 1]$, the two components are combined using a case-specific outcome weight $w_o \in [0, 0.5]$:

$$\mathrm{TC} = (1 - w_o)\,\mathrm{PA} + w_o\,\mathrm{OC}. \tag{4}$$

The outcome contributes at most 50% of TC. Its weight is set per case according to the importance of the requested result. Process-focused cases use $w_o = 0$: OC is not evaluated separately, and TC depends solely on PA. Detailed scoring criteria are provided in the PA and OC evaluator templates in Appendix H.3.

D.4 Causal Plausibility

CP evaluates whether the complete sequence follows the applicable physical, logical, game, or task-specific rules. The judge checks adjacent cause–effect relations as well as long-range dependencies, object conservation, prerequisites, contact and support, topology, temporal order, and task-specific constraints. Detailed scoring criteria are provided in the CP evaluator template in Appendix H.3.

D.5 Visual Consistency

VC evaluates whether task-relevant identities and scene attributes remain consistent across the sequence. It checks for unexplained changes in object identity, count, geometry, scale, material, camera, and background, including disappearance, duplication, deformation, teleportation, and scene

resets. The judge compares both adjacent and non-adjacent frames to capture cumulative drift, while exempting changes required by the task or visibly established by the causal process. Detailed scoring criteria are provided in the VC evaluator template in Appendix H.3.

D.6 Score Aggregation

We first express the scores derived from discrete ratings on a common 0–3 scale: TC from Section D.3 is multiplied by 3, while CP and VC retain their original 0–3 scores. We then compute their weighted aggregate on this scale and convert it to a 100-point score:

$$\text{Final} = \frac{100}{3}\left(\alpha_1 \,\text{TC} + \alpha_2 \,\text{CP} + \alpha_3 \,\text{VC}\right). \tag{5}$$

We set $\alpha_1 = \alpha_2 = 0.4$ and $\alpha_3 = 0.2$: TC and CP receive equal weight, while VC receives half the weight of either. These aggregation weights are fixed across categories and are distinct from the case-specific outcome weight w_o used within TC. We report both the individual metric scores and their weighted aggregate on the 100-point scale.

D.7 Performance Across Evaluation Dimensions

Figure 11 shows how representative models' scores are distributed on WorldCast.

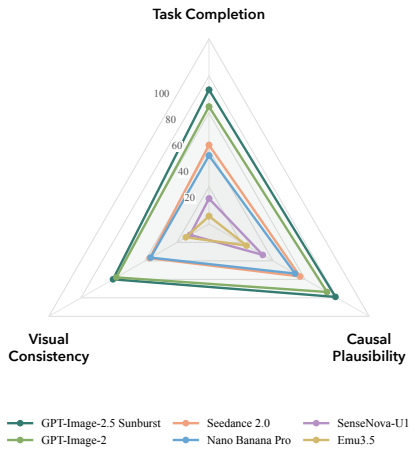


Figure 11: Performance across TC, CP, and VC on WorldCast.

D.8 Parameter Settings

We use GPT-5.5 as the judge with reasoning effort set to `medium`. All GPT-6 configurations use GPT-6 Astra with `medium` reasoning effort, whether accessed through the Responses API or used as an agent through Codex. Our Codex-based experiments use `codex-cli 0.153.4`.

E Stability of LMM-as-a-Judge

To assess sensitivity to evaluator choice, we rescore the same model outputs with the open-source GLM-5.3-Flash (Z.ai, 2026), GPT-5.5 (OpenAI, 2026a), and Gemini 3.1 Pro (Google DeepMind, 2026a). Figure 12 shows a systematic calibration difference: GLM-5.3-Flash assigns higher absolute scores, whereas Gemini 3.1 Pro is more conservative. Nevertheless, the model ordering is largely preserved across all three judges, suggesting that our relative conclusions are robust to evaluator choice even when absolute score scales differ.

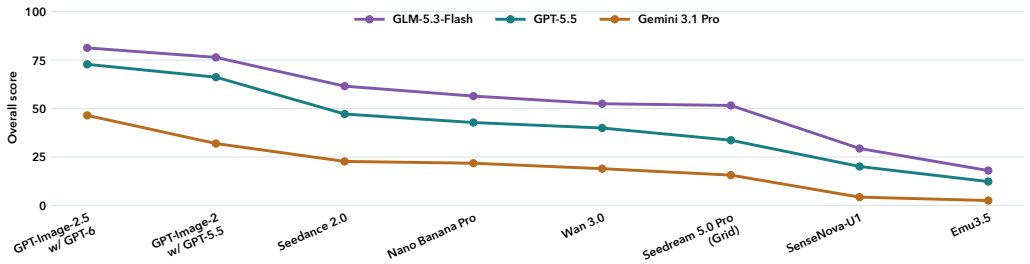


Figure 12: Overall model scores assigned by three LMM judges.

Model stability across repeated generations. We further repeat generation for three rounds and score every run with GPT-5.5. Table 4 reports the resulting mean and standard deviation for models with complete repeated runs. Means are computed over all 320 cases, and standard deviations are computed across generation-run means. Overall-score standard deviations do not exceed 0.53, although individual categories show greater variability, reaching 2.7 points.

Method	**L3 Tasks**			**L4 Tasks**					**Overall**
	Observation-Conditioned Transitions	*Physical Simulation*	*Adaptive Task Completion*	*Physical Properties*	*Visual Cues*	*In-Context Procedural Learning*	*Logic and Rules*	*Visual Goal Realization*	
SenseNova-U1	24.1 ± 0.73	24.6 ± 0.43	24.7 ± 0.39	19.9 ± 0.84	19.1 ± 1.21	20.1 ± 0.45	5.9 ± 0.34	12.2 ± 0.21	20.1 ± 0.30
Emu3.5	17.4 ± 0.43	13.8 ± 0.29	18.5 ± 0.84	6.6 ± 0.49	10.3 ± 0.71	8.1 ± 0.54	4.8 ± 0.50	7.6 ± 0.56	12.3 ± 0.16
BAGEL	4.9 ± 0.43	7.0 ± 0.42	8.3 ± 0.28	8.5 ± 0.48	12.5 ± 0.11	7.0 ± 1.07	2.9 ± 0.32	4.0 ± 0.67	6.8 ± 0.13
Qwen-Image	7.8 ± 0.29	6.8 ± 0.16	6.2 ± 0.63	4.0 ± 0.56	6.9 ± 0.31	6.1 ± 0.49	0.6 ± 0.16	3.4 ± 0.21	5.6 ± 0.12
GPT-Image-2.5	85.6 ± 1.12	69.6 ± 0.79	66.1 ± 1.01	73.6 ± 1.84	76.7 ± 1.08	71.6 ± 0.92	74.3 ± 1.13	61.3 ± 0.74	72.8 ± 0.39
GPT-Image-2	74.2 ± 0.74	70.6 ± 1.12	59.6 ± 0.95	66.0 ± 0.72	62.6 ± 0.70	70.6 ± 0.76	66.6 ± 0.16	51.8 ± 2.68	66.2 ± 0.53
Nano Banana Pro	36.2 ± 1.15	56.1 ± 0.95	43.5 ± 0.40	49.9 ± 0.42	33.7 ± 0.62	43.3 ± 1.55	41.4 ± 0.81	28.5 ± 0.50	42.8 ± 0.48
Seedream 5.0 Pro	30.3 ± 2.00	45.9 ± 0.69	33.4 ± 1.02	46.7 ± 1.40	27.7 ± 1.16	38.9 ± 0.76	11.9 ± 0.48	29.1 ± 0.63	33.6 ± 0.19
Seedream 4.5	19.4 ± 0.79	14.9 ± 0.13	18.7 ± 0.14	19.7 ± 1.14	16.0 ± 0.49	21.4 ± 1.03	3.4 ± 0.38	12.8 ± 0.71	16.0 ± 0.03
Seedance 2.0	53.3 ± 0.86	53.4 ± 0.83	48.7 ± 0.46	40.8 ± 1.32	56.6 ± 0.79	55.4 ± 1.96	17.7 ± 0.97	43.3 ± 0.39	47.2 ± 0.08
Gemini Omni	45.6 ± 0.58	50.2 ± 0.26	41.0 ± 0.23	47.2 ± 2.49	42.7 ± 0.88	49.9 ± 1.25	16.2 ± 0.56	33.0 ± 1.76	41.6 ± 0.12
Wan 3.0	51.9 ± 1.22	47.3 ± 2.20	38.5 ± 1.20	42.7 ± 1.19	42.7 ± 1.51	35.5 ± 0.98	17.2 ± 1.06	29.6 ± 0.85	39.9 ± 0.35

Table 4: Model stability across repeated generations. Scores are measured by GPT-5.5 and reported as mean ± standard deviation over three generation runs. Model names and ordering follow Table 1.

F Additional Analysis on Coding Artifacts

The main paper (Section 4.3.3) discusses indirect uses of coding artifacts and errors introduced by direct visual conditioning. As Figure 13 shows, in most cases, the agent chooses not to pass these artifacts to the generator, but uses them to guide revisions or translates them into more precise descriptions. However, some of the largest gains occur in tasks where the agent can explicitly represent or guide the process: with the coding harness, GPT-Image-2.5 improves by 20.0 points on Visual Goal Realization and 19.0 points on Physical Simulation over the Responses API baseline (Table 3). In this appendix, we therefore focus on how the agent operates when visual artifacts are used as direct inputs to the generator.

Direct use is most frequent in Logic and Rules (22 of 35 cases) and Physical Simulation (24 of 61 cases). These tasks lend themselves to externalizing otherwise latent reasoning as discrete state trajectories, geometric constructions, or physical-process sketches. The examples in Figure 13 illustrate both patterns: an overlay communicates a physically grounded progression to GPT-Image-2.5, whereas a coded state simulator supplies intermediate configurations that Nano Banana Pro renders into the requested sequence.

These cases highlight how capable current agents can be on tasks they can simulate through code. Previous work emphasizes composition through structural drafts in CoCo (Li et al., 2026b), programmable visual sketches in GenClaw (Ye et al., 2026), and context acquisition through search in Qwen-Image-Agent (Zhang et al., 2026b). Our simple harness on WORLDCAST, meanwhile, demonstrates the strength of LLMs in simulating processes. Their world knowledge also enables guidance to be adapted dynamically to the task. In the Physical Simulation example, the agent recognizes that the original reference alone provides insufficiently fine-grained guidance for subtle changes in piston height, and therefore introduces guide lines and labels to specify those changes. This suggests that future enhancers should extend well beyond language enhancement to intelligently provide task-appropriate multimodal information. The underlying visual generators must evolve alongside these advances to effectively interpret and use such guidance.

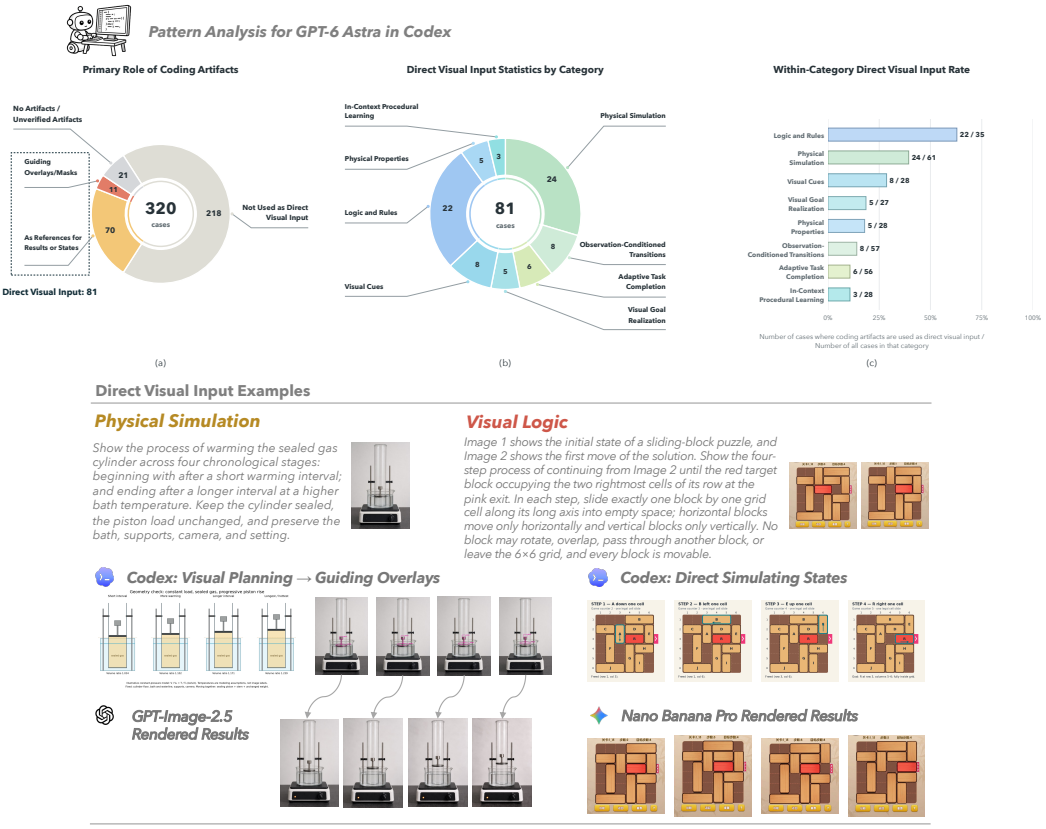


Figure 13: Usage patterns of coding artifacts under GPT-6 Astra with Codex. Top: the role and category distribution of coding artifacts across 320 cases. Bottom: representative examples using guiding overlays and simulated intermediate states as direct visual inputs.

G Detailed Qualitative Examples



Figure 14: Detailed Qualitative Examples, Model Outputs and Error Patterns.

Visual Cues





Prompt: Images 1 and 2 show the same home from the cleaning robot's first-person viewpoint and a route-plan view. Show the four-step process of the cleaning robot following the marked route from its current position, depicting throughout exactly what the robot sees from its low first-person viewpoint while facing its direction of travel. Preserve the same rooms, furniture, topology, scale, and lighting. The red route, point markers, and start label in Image 2 are guidance cues only and should not appear in the generated robot-view process.

Hints for judge

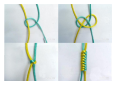
TC: Move from the living-room start toward its center, stop at the next marked point, then turn right and continue into the center of the bedroom.

CP: Move continuously through traversable floor space and make one right turn. Do not pass through walls or furniture, or teleport.

VC: First-person furniture positions must match the overhead plan. Do not misplace or duplicate rooms or objects. Remove the red route, path nodes, and start text from generated views.



In-Context Procedural Learning





Prompt: Image 1 demonstrates a knotting method, while Image 2 shows the same kind of black and blue straps in an open starting arrangement. Show the four-step process of continuing from the state in Image 2 by following the demonstrated method, keeping the same straps, viewpoint, and background throughout.

Hints for judge

TC: Continue from the single X-shaped crossing in Image 2. Form a loop using the reference method, then follow its threading direction, gradually tighten, and form successive knots. Show four distinct, correctly ordered knot states ending in the reference structure.

CP: Keep the black and blue straps' over-under relationships continuous between steps. Loops shrink plausibly when tightened, without unexplained swaps.

VC: Preserve the same black and blue flat straps, widths, textures, overhead view, and light background. Total length must not change without cause.



Visual Goal Realization





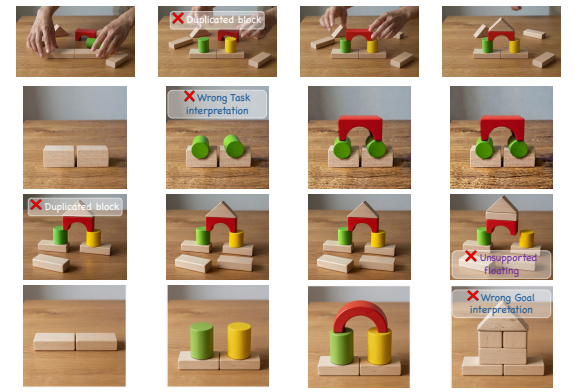
Prompt: Show the process of using the blocks in Image 1 to build the same structure shown in Image 2 in four steps. Use every provided block exactly once, assemble the structure progressively from the bottom upward, and keep the tabletop, viewpoint, and appearance of the blocks consistent.

Hints for judge

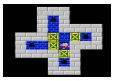
TC: Use the two rectangular blocks as a base, place the two cylindrical supports, add the arch, and place the triangular prism as the roof. Steps can be swapped as long as the order remains bottom-up. Match Image 2's structure and use all the six blocks.

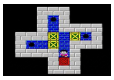
CP: Build from the bottom upward. Upper blocks must have stable support below, without floating or interpenetration.

VC: Preserve the colors, shapes, sizes, and number of the six blocks from Image 1. Do not add, remove, or duplicate blocks.



Logic and Rules





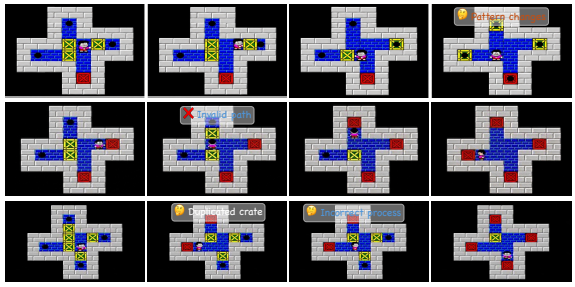
Prompt: Image 1 shows the initial state and Image 2 shows the first completed placement. Starting from Image 2, show the four-step process of the character pushing the remaining crates into their destination holes. Crates cannot be pulled, crossed, overlapped, or pushed into walls.

Hints for judge

TC: Starting from Image 2, place the three remaining crates on the remaining holes. Keep the bottom crate already on its goal. The final frame must visibly show a red crate on each of the four holes. Judge the visible state in each frame: a crate on a hole must be red; a crate off a hole must be yellow. Accept any legal solution and do not infer missing placements.

CP: Push only one crate at a time. Crates cannot be pulled, pass through walls, or overlap one another.

VC: Preserve the wall and floor layout, fixed hole positions, character identity, and the same four crates. No additions, removals, or duplication. Crate color indicates placement status, not a change of identity.



H PROMPT TEMPLATES

H.1 GENERATION-INTERFACE ADAPTERS

Grid generation.

```
{{ORIGINAL_TASK}}

Output N steps, and arrange the N steps in a grid.
Please collage all the steps into one image. Do not render any step labels,
captions, instructions, explanatory text, arrows, or guide graphics inside the
image unless visible text is explicitly required by the user's task.
```

Interleaved generation.

```
{{ORIGINAL_TASK}}

Output each step with one textual instruction and one image. Don't output all
steps in one image, instead output N images interleaved with text description.
Do not render the textual instructions, step labels, captions, explanatory
text, arrows, or guide graphics inside the images unless visible text is
explicitly required by the user's task.
```

Video generation.

```
{{ORIGINAL_TASK}}

Create a single coherent video that visually carries out the requested process
from beginning to end. Show every requested stage in the specified
chronological order, with each stage clearly visible and connected through
natural motion. Do not use a split screen or other instructional overlays.
```

H.2 AGENTIC VIDEO ADAPTER

```
You are selecting semantic benchmark frames from a generated video.

Inspect the source MP4 itself. You may repeatedly use ffmpeg to sample any
timestamps and compare frames. Use the text instruction, reference
images, complete video, and neighboring video states to identify the
{{STEP_COUNT}} frames that most clearly demonstrate the requested key steps or
milestones. The selected timestamps must be strictly increasing. Do not select
evenly spaced timestamps merely for convenience. If a requested state never
appears, select the closest visible state in its correct chronological region
and mark it as achieved=false. This is frame selection, not quality scoring.

You MUST always select and extract exactly {{STEP_COUNT}} frames, even when the
video is poor, incomplete, ambiguous, or does not visibly achieve one or more
requested states. In that extreme case, you may choose the temporal point based
on when the milestone is expected to occur as specified in the prompt. Never
omit a step, refuse selection, stop early, or leave an output file unwritten
because no ideal frame exists. For every missing state, choose the semantically
closest visible frame from the most appropriate chronological region; use
neighboring frames and the requested order to make the best approximation. Set
achieved=false and briefly explain the mismatch in evidence, but still extract
that frame and include it in the JSON result.

Use this ffmpeg executable:
{{FFMPEG_EXECUTABLE}}

Extract the final full-resolution frames to exactly these paths:
{{OUTPUT_FRAME_PATHS}}

Return the matching JSON only after every file has been written. Each filename
must be the basename only and exactly step_1.jpg through
step_{{STEP_COUNT}}.jpg; do not return a temporary sampling path such as
frames/t2.4.jpg. Before returning, verify that all {{STEP_COUNT}} required JPEG
files exist, are non-empty, and correspond one-to-one with the JSON entries.

Evaluation input:
{{EVALUATION_INPUT_JSON}}
```

```
## Output Contract

{
  "steps": [
    {
      "step": 1,
      "timestamp_seconds": 0.0,
      "filename": "step_1.jpg",
      "achieved": true,
      "evidence": "Visible evidence for the requested milestone, or a brief mismatch
     explanation."
    }
  ],
  "overall_notes": "Brief notes about the selection."
}
```

H.3 LMM-AS-A-JUDGE TEMPLATES

TC Sub-Metric: Progress Alignment (PA)

```
# Progress Alignment Evaluation

Your task is to act as a PROGRESS ALIGNMENT AUDITOR in Strict Evaluation Mode.
Inspect the complete ordered image sequence and judge whether the overall task
unfolds in accordance with the user's intended process. Use the text
instruction, HINTS, and reference images to determine the task-critical
milestones, then judge whether the sequence visibly realizes those milestones
in the required or logically necessary order and develops in the way the user
expected. You should assess whether the whole sequence unfolds correctly and
give a sequence-level evaluation; a correct final result alone is insufficient.

## Scoring Hierarchy

SCORE 2 [INTENDED PROCESS AND ALL MILESTONES REALIZED]

- The sequence's overall task unfolds in the way the user intended, and every
  required milestone is visibly realized rather than merely approached,
  initiated, implied, or described.
- The process shown across the sequence matches the expected interpretation of
  the user's request, rather than merely containing isolated correct actions,
  a plausible start, or a correct-looking endpoint.
- The task-critical stages or milestones appear with sufficient visible
  specificity and in the required or logically necessary order, with no
  important omission, skipped dependency, premature outcome, unrelated
  segment, contradiction or regression. Correct start and end states do not
  compensate for a missing critical process stage.

SCORE 1 [CORE ALIGNMENT REALIZED WITH SECONDARY DEFECTS]

- The user's intended core process is visibly recognizable and every
  indispensable task-critical milestone is substantially realized, but the
  overall process still has secondary incompleteness or imprecision.
- Any missing, ambiguous, premature, out-of-order, interrupted, regressed,
  unrelated, or visually unsupported content is limited to a non-core stage or
  secondary detail whose failure does not prevent realization of the user's
  central intent.
- If even one indispensable task-critical milestone is absent, only approached
  or initiated, visually ambiguous, skipped, contradicted, carried out in a
  fundamentally wrong way, or claimed without direct visual support, assign
  SCORE 0 rather than SCORE 1.
- When all indispensable milestones are realized, prominent unrequested
  instructional text, captions, step labels, arrows, or guide graphics may
  still reduce the sequence to SCORE 1.

SCORE 0 [KEY MILESTONE NOT REALIZED]

- At least one indispensable task-critical milestone is absent, only
  approached or initiated, visually ambiguous, skipped, premature, out of
  order in a way that breaks the process, interrupted, regressed,
  contradicted, carried out in a fundamentally wrong way, or claimed without
  direct visual support.
- A milestone is indispensable when it is explicitly emphasized by the text
  instruction or HINTS, is necessary to realize the user's central intent, or
  is a required dependency for a later key state or result.
- Other correct milestones, a matching subject or setting, isolated relevant
  actions, a plausible start, or a correct-looking endpoint do not compensate
  for one uncompleted indispensable milestone.
```

```
- The sequence also receives SCORE 0 when it remains essentially static despite
  required change, or when its changes do not form a recognizable realization
  of the user's intent.

## Supplementary Rules

Base your judgment solely on the image itself; in most cases, any text or
guiding cues added on top of the image should not affect your evaluation. When
the image is only partially visible, evaluate it according to the information
that is visible.

## Evaluation Steps

Use the text instruction, HINTS, and reference images to identify the
task-critical milestones, mark as indispensable every milestone explicitly
emphasized by the text instruction or HINTS, necessary to realize the user's
central intent, or required as a dependency for a later key state or result,
and identify any required or logically necessary ordering. Inspect the complete
ordered sequence and assign the scores according to the Scoring Hierarchy.
Assign an initial score using the scoring criteria. Then challenge that score
by actively looking for previously missed problems, counterevidence, or
alternative interpretations in the complete ordered sequence. Check whether
these objections are supported by visible evidence and warrant changing the
score; do not force a score change. Finally, give the revised final score (0,
1, or 2), retaining the initial score if it remains supported.

## Evaluation Data

- TEXT INSTRUCTION
- HINTS
- REFERENCE AND INITIAL IMAGES
- ALL ORDERED GENERATED IMAGES

## Output Contract

Return only a JSON object. Use an integer score (0, 1, or 2) and briefly
explain the judgment with specific visual evidence.

{
  "score": 0,
  "evidence": "Brief visual evidence supporting the score."
}
```

TC Sub-Metric: Outcome Completion (OC)

```
# Outcome Completion Evaluation

Your task is to act as an OUTCOME COMPLETION AUDITOR in Strict Evaluation Mode.
Given the text instruction, HINTS, reference images, and the final generated
image, judge whether the final result fulfills the requested outcome. The
original text instruction is for generating a sequence of frames, but no
intermediate generated images are provided here, so you should only evaluate
the final image and check if it meets the outcome of the instruction.

## Scoring Hierarchy

SCORE 1 [REQUESTED OUTCOME FULLY REALIZED]

- The final image visibly realizes the user's requested endpoint, and every
  indispensable outcome requirement is satisfied rather than merely
  approached, initiated, implied, or described.
- The required final entities, quantities, attributes, configuration, spatial
  relations, placement, and reference-specific features are correct wherever
  they are specified by the text instruction or HINTS or necessary to the
  user's central goal.
- The result is complete with sufficient visible specificity to verify it. No
  required final-state element is missing, contradictory, incorrectly
  substituted, or hidden in a way that prevents verification.

SCORE 0 [REQUESTED OUTCOME NOT FULLY REALIZED]

- At least one required endpoint condition or specified final-state detail is
  missing, incorrect, contradicted, incomplete, or visually unverifiable.
  Secondary defects in required details also receive SCORE 0; there is no
  partial credit.
- An outcome requirement is indispensable when it is explicitly emphasized as
  a required final condition by the text instruction or HINTS, or when its
  absence prevents realization of the user's central requested result. An
  intermediate milestone is not automatically an endpoint requirement.
```

```
- The outcome is only asserted by a caption, label, arrow, model-authored plan,
  or plausible narrative rather than shown. Do not infer successful placement,
  transfer, removal, or completion solely because an object is missing or
  obscured. When absence itself is the requested endpoint, credit it only if
  the relevant final region is visible and supports that requirement.

## Evaluation Steps

Read the text instruction, HINTS, and reference images to extract the requested
endpoint. Separate final-state requirements from instructions about how to
reach that state; do not require the final image to repeat intermediate
actions. Inspect the final generated image and match each endpoint requirement
to direct visible evidence. Assign an initial score using the scoring criteria.
Then challenge that score by actively looking for previously missed problems,
counterevidence, or alternative interpretations in the final generated image
only. Check whether these objections are supported by visible evidence and
warrant changing the score; do not force a score change. Finally, give the
revised final score (0 or 1), retaining the initial score if it remains
supported.

## Evaluation Data

- TEXT INSTRUCTION
- HINTS
- REFERENCE AND INITIAL IMAGES
- FINAL GENERATED IMAGE ONLY (ONE IMAGE; NO INTERMEDIATE GENERATED IMAGES)

## Output Contract

Return only a JSON object. Use an integer score (0 or 1) and briefly explain
the judgment with specific visual evidence.

{
  "score": 0,
  "evidence": "Brief visual evidence supporting the score."
}
```

Causal Plausibility (CP).

```
# Causal Plausibility Evaluation

Your task is to act as a SEQUENCE-LEVEL CAUSAL AND WORLD-LAW AUDITOR in Strict
Evaluation Mode. Inspect the complete ordered image sequence produced by the
generation model and judge whether the overall process follows the object
transformations, physical laws, game rules, everyday phenomena, logical
dependencies, and task-world rules emphasized by the text instruction and
HINTS. Apply realistic everyday physics by default, but follow explicit
fictional, game, diagrammatic, symbolic, or other task-specific world rules
when the task establishes them.

## Scoring Hierarchy

SCORE 3 [FULLY PLAUSIBLE]

- All concerns in the text instruction or HINTS are satisfied with direct
  visual evidence.
- The complete process is fully plausible, and every critical adjacent
  transition visibly follows from the state and action before it.
- The core mechanisms, constraints, object topology, and long-range
  dependencies all hold consistently, with no identifiable deviation from the
  governing rules and no skipped mechanism disguised by a plausible final
  state.

SCORE 2 [KEY MECHANISM PLAUSIBLE WITH OTHER MINOR FLAWS]

- Some new local rule deviation, weakened mechanism, incomplete long-range
  dependency, or minor unsupported transition is still found.
- The concerns in the text instruction or HINTS are partially satisfied, but
  the intended causal mechanism remains recognizable: the critical states and
  transitions broadly follow the expected governing rules.
- The overall process remains fluent and broadly instruction-aligned; no
  critical mechanism is skipped, contradicted, or replaced by an abrupt
  structural transformation.

SCORE 1 [CORE MECHANISM RECOGNIZABLE BUT VISIBLY FLAWED]

- The intended core causal mechanism remains visibly recognizable, and at
  least one critical cause-to-effect transition is supported, but a core
  concern in the text instruction or HINTS or a key verifiable condition in the
```

```
  HINTS is clearly violated at a critical state or transition.
- The broad phenomenon follows the applicable physical law, game rule,
  everyday phenomenon, or task-world rule, but a critical action, prerequisite,
  constraint, or transition is missing, unsupported, or mechanically
  incompatible.
- Alternatively, the process itself may be broadly plausible but lacks a
  strong connection to the reference images, text instruction, or HINTS.
- Mere thematic resemblance or a plausible final state is insufficient for
  SCORE 1 when the central causal mechanism is not visibly realized.

SCORE 0 [CORE MECHANISM ABSENT, CONTRADICTORY, OR IMPOSSIBLE]

- The central causal mechanism required to produce the main change is absent,
  reversed, contradicted, or replaced by an unsupported transformation, so the
  intended process is not causally recognizable as realized.
- The main result depends on a task-critical object, action, constraint, or
  state appearing, disappearing, teleporting, duplicating, or changing
  incompatibly without a visible or ordinarily inferable cause.
- The overall process is self-contradictory, or a key mechanism makes the main
  change impossible under the applicable world rules.
- A matching subject, broad phenomenon, or plausible-looking endpoint does not
  rescue a sequence whose core causal chain is missing or fundamentally
  invalid.

## Supplementary Rules

- If the output is static or changes too little, judge it according to the text
  instruction; when the task requires visible change, Causal Plausibility
  cannot exceed 1.
- If any task-critical transition cannot be causally reconciled from the
  visible states under the applicable world rules, Causal Plausibility cannot
  exceed 1. Ordinary unshown intermediate motion may be inferred only when the
  visible endpoints are mutually compatible and require no invented
  intervention, constraint change, or hidden state.
- Grade an abrupt or omitted transition by its causal severity rather than
  abruptness alone. A harmless compressed intermediate action needs no penalty
  when the visible endpoints remain mutually reachable; a local weakened or
  imprecise mechanism supports SCORE 2; an unsupported task-critical
  transition is capped at SCORE 1; and a self-contradictory or world-impossible
  core mechanism supports SCORE 0.
- If a task-relevant object or tool visible in one generated image is absent
  from a later image without a visible or ordinarily inferable removal,
  placement, consumption, transformation, occlusion, or exit from the frame,
  object conservation and causal continuity are violated and Causal
  Plausibility cannot exceed 1|even if the object is no longer needed after
  the action. If the unexplained disappearance removes the central mechanism,
  makes the main result causally unsupported, or is part of an impossible or
  self-contradictory core process, assign SCORE 0.

## Evaluation Steps

- Follow the visible changes from the first generated image to the last. First
  identify the applicable world assumptions and the task-critical states,
  constraints, and dependencies from the text instruction, HINTS, and reference
  images. Use realistic everyday physics by default, but follow explicit
  fictional, game, diagrammatic, symbolic, or other task-specific rules when
  provided.
- For every adjacent pair, identify the visible prior state, the visible action
  or cause, and the resulting state. Track every task-relevant object and tool
  across the pair, including where it remains, how it moves, and whether any
  appearance, disappearance, duplication, consumption, transformation,
  occlusion, or exit from the frame has visible or ordinarily inferable causal
  support. Inspect the relevant prerequisites and constraints, including
  support, contact, containment, object topology and component connectivity,
  material behavior, conservation of object count, temporal ordering, tool
  availability, geometry, and task-specific logical rules. Judge whether each
  cause can plausibly produce the state that follows.
- Check non-adjacent and long-range dependencies, including whether an early
  action enables or prevents a later state. Grade deviations by causal
  severity: distinguish harmless temporal compression, a local weakened or
  imprecise mechanism, an unsupported task-critical transition, and an
  impossible or self-contradictory core process. Keep Causal Plausibility
  separate from visual continuity: count an identity, shape, or appearance
  change here only when it breaks the causal mechanism or makes the resulting
  state unreachable under the applicable world rules.
- Assign an initial score using the scoring criteria. Then challenge that score
  by actively looking for previously missed problems, counterevidence, or
  alternative interpretations in the complete ordered sequence. Check whether
  these objections are supported by visible evidence and warrant changing the
  score; do not force a score change. Finally, give the revised final score (0,
  1, 2, or 3), retaining the initial score if it remains supported.
```

```
## Evaluation Data

- TEXT INSTRUCTION
- HINTS
- REFERENCE AND INITIAL IMAGES
- ALL ORDERED GENERATED IMAGES

## Output Contract

Return only a JSON object. Use an integer score (0, 1, 2, or 3) and briefly
explain the judgment with specific visual evidence.

{
  "score": 0,
  "evidence": "Brief visual evidence supporting the score."
}
```

Visual Consistency (VC).

```
# Visual Consistency Evaluation

Your task is to act as a MULTI-FRAME CONTINUITY AUDITOR in Strict Evaluation
Mode. Compare the complete ordered image sequence and evaluate whether visual
state that should persist remains consistent across frames. Only the specific
state change explicitly requested by the user or visibly required by the causal
process is exempt from Visual Consistency penalties; unrelated changes to
identity, geometry, material, topology, mechanism structure, camera, or
background remain Visual Consistency errors.

## Scoring Hierarchy

SCORE 3 [FULLY CONSISTENT]

- Task-relevant invariants such as identity, count, shape, scale, color,
  material, topology, mechanism structure, relative position, camera, and
  background remain stable.
- Every change is directly justified by the visible process, with no
  identifiable appearance, disappearance, duplication, teleportation,
  structural replacement, or cumulative drift.
- Unless explicitly requested, no new instructional caption, step label,
  explanatory text, arrow, or guide graphic appears in the generated images;
  scene-native text already present in the reference is preserved normally.

SCORE 2 [MINOR ISSUES BUT KEY OBJECTS STABLE]

- Key objects emphasized by the HINTS or text instruction show only slight
  drift that can recover in later frames.
- Objects not emphasized by the HINTS or text instruction may have a small but
  real identity, geometry, scale, position, or background continuity deviation.

SCORE 1 [CONTINUITY ISSUE BUT KEY OBJECTS RECOGNIZABLE]

- The target remains broadly recognizable, but continuity is poor; a key
  object has at least one clear color, identity, geometry, scale, topology,
  mechanism-structure, position, or background continuity deviation.
- Objects not emphasized by the HINTS or text instruction have strong
  deviations that affect the whole sequence.

SCORE 0 [TOTAL FAILURE]

- The target object is not preserved at all, or there is unexplained loss,
  addition, or duplication of important objects, a major scale jump,
  teleportation, background reset, severe deformation, topology or mechanism
  replacement, bizarre texture, or inexplicable distortion.
- An important object disappears without a visible removal or placement action.
  Do not infer that it was safely moved merely because it is absent later.
- If an object or tool explicitly named by the HINTS, or required to remain
  consistent in instruction disappears, appears, or duplicates without a
  visible removal, placement, entry, exit, consumption, transformation, or
  occlusion, assign SCORE 0.

## Evaluation Steps

- Compare adjacent and non-adjacent images across the complete ordered
  sequence. Track every task-relevant object and tool from frame to frame,
  including its count and visible location, and form a score by checking for
  unjustified changes in identity, count, shape, scale, color, material,
  topology, mechanism structure, relative position, camera, or background,
```

```
  including appearance, disappearance, duplication, teleportation, structural
  replacement, cumulative drift, distortion. Exempt only the specific visual
  changes directly required by the text instruction, or visibly established
  causal process.
- Assign an initial score using the scoring criteria. Then challenge that score
  by actively looking for previously missed problems, counterevidence, or
  alternative interpretations in the complete ordered sequence. Check whether
  these objections are supported by visible evidence and warrant changing the
  score; do not force a score change. Finally, give the revised final score (0,
  1, 2, or 3), retaining the initial score if it remains supported.

## Evaluation Data

- TEXT INSTRUCTION
- HINTS
- REFERENCE AND INITIAL IMAGES
- ALL ORDERED GENERATED IMAGES

## Output Contract

Return only a JSON object. Use an integer score (0, 1, 2, or 3) and briefly
explain the judgment with specific visual evidence.

{
  "score": 0,
  "evidence": "Brief visual evidence supporting the score."
}
```

Designated-Step Accuracy.

This evaluator is used for the main-text analysis *Interleaved generation faces the challenge of accumulated context* in Section 4.3.2. It measures how generation accuracy changes across successive steps as interleaved context accumulates.

```
# Step Accuracy Evaluation

Your task is to act as a DESIGNATED-STEP ACCURACY AUDITOR in Strict Evaluation
Mode. Use the text instruction, HINTS, reference images, and the model's text
plan paired with the designated generated image to evaluate that image only.
Extract the concrete milestone or task-relevant change required for this step,
then judge whether the image visibly realizes that requirement, makes positive
task-relevant progress, and matches its paired plan.

## Scoring Hierarchy

SCORE 1 [STEP REQUIREMENT REALIZED]

- The image visibly realizes the requirement for this exact step in the ordered
  sequence. Any explicitly required action, state, entity, quantity, attribute,
  spatial relation, or reference relation for this step is satisfied.
- The visible action or state agrees with the model's text plan paired with this
  image and produces positive progress toward the user's task. The model's plan
  does not override the user's instruction, HINTS, or reference requirements;
  following a wrong or irrelevant plan is not sufficient.
- The designated step satisfies the constraints applicable to its own action or
  state and is physically plausible. Required contact, support, containment,
  geometry, object count, and material behavior are visibly compatible with the
  depicted action.
- No task-relevant object appears, disappears, duplicates, teleports, or
  abruptly deforms without a task-valid explanation supported by the supplied
  visual evidence. Legitimate task-required transformations and supported
  occlusion are not errors.
- Requirements in HINTS that are declared explicitly relevant to this step are
  satisfied.

SCORE 0 [STEP REQUIREMENT NOT REALIZED]

- The visible change is irrelevant to constitute the required milestone. A
  repeated or essentially unchanged image fails unless maintaining that state
  is explicitly required for this step.
- The designated image omits or violates a required action, state, entity,
  quantity, attribute, spatial relation, reference relation, or local
  constraint for this step; uses the wrong reference; or depicts a different
  action or state from the one required for this step.
- The image contradicts its paired model-authored plan, or the planned and
  depicted action fails to produce task-relevant positive progress. Do not
  treat the plan's claim of success as evidence that the image succeeded.
```

```
- The designated action or state has a visible physical or task-world
  violation, such as impossible contact or support, incompatible geometry, or
  an unexplained appearance, disappearance, duplication, teleportation, or
  abrupt deformation of a task-relevant object.
- Requirements in HINTS that are declared explicitly relevant to this step are
  not satisfied or only partially satisfied.

## Evaluation Steps

### Identify the Designated Step

- Read the designated step index, original text instruction, HINTS, reference
  images, and model-authored text plan paired with the designated image.
  Evaluate only that step; do not produce judgments for other steps.
- Extract the concrete action or state promised for this step, its task-relevant
  visual requirements, and relevant HINTS. The user's task takes priority over
  an incorrect model plan.

### Evaluate the Designated Generated Image

- Inspect the designated image's action, entities, quantities, attributes, and
  relations. Check that it substantially realizes its own step requirement and
  makes meaningful positive progress.
- Compare the image with its paired textual plan, then check local physical
  plausibility and task-relevant object integrity. Any deduction must cite
  visible evidence attributable to the designated step.
- Other supplied images are context only: do not penalize the designated step
  because another step fails. Do not invent an unseen comparison state to
  claim disappearance or deformation.

### Recheck and Return One Judgment

- Reinspect the designated image and challenge the initial judgment against
  both score definitions. Correct any unsupported inference or cross-step
  blame. Output the accuracy in SCORE 0 or 1.

## Evaluation Data

- TEXT INSTRUCTION
- HINTS
- REFERENCE AND INITIAL IMAGES
- DESIGNATED STEP INDEX
- MODEL-AUTHORED TEXT PLAN PAIRED WITH THE DESIGNATED IMAGE
- DESIGNATED GENERATED IMAGE

## Output Contract

Return only a JSON object. Use an integer score (0 or 1) and briefly explain
the judgment with specific visual evidence. Set step_index to the designated
step; do not score other steps.

{
  "step_index": 1,
  "score": 0,
  "evidence": "Brief visual evidence for the designated step."
}
```

H.4 AGENTIC GENERATION HARNESSES

Inspection Loop without Coding Tools.

```
Tool Configuration: code_mode: off; shell: off; web_search: disabled
```

```
Main Prompt:
{{ORIGINAL_TASK}}

## Required Workflow

1. Plan the intended image sequence using the task and supplied references.
2. Generate the images using the image-generation tool or API assigned to the
   current baseline.
3. Inspect each image and the full sequence for task compliance, geometry,
   orientation, and continuity. Edit or regenerate when useful, then inspect
   again.
4. Select the best existing sequence within the shared call and time budgets.
   Do not revise images that are already correct.
```

```
## Constraints

- Use the image-generation and editing tool or API assigned to the current
  baseline; viewing images is unrestricted. Do not switch to another image
  model or provider.
- Do not use code or modeling tools to construct, render, or use intermediate
  visual artifacts.
- All image-generation and editing attempts, including failures, share one
  per-Case budget of 2 × {{STEP_COUNT}} calls
  ({{MAX_IMAGE_CALLS_2N}} submissions). Initial generations, regenerations,
  and edits all count. Never reset or bypass this budget.
- Non-image-generation tool use does not consume image-generation calls, but
  must remain within the Case time limit. Any image-generation or editing API
  call counts regardless of its purpose.
- At the cap, make no further image-generation or editing calls. Select from
  existing outputs and report any remaining shortfall.
- You have 150 minutes total, with reminders every 15 minutes.
- Use the supplied task and references and your own work from this Case. Do not
  read other Cases, sessions, baseline results, scores, Judge hints, or
  evaluation files. Do not modify source data, reference files, other runs'
  outputs, the budget gate, ledger, or configuration, or start other sessions.

## Output Contract

- Return exactly {{STEP_COUNT}} accepted final images in step order, one
  absolute Markdown image link per step:
  ![Step N](/absolute/path/to/image.png).
- Include only the selected final images, not discarded drafts or superseded
  revisions. Verify that each path points to an existing image.
- If the required sequence cannot be completed within the budget, return the available
  selected images and briefly state the shortfall; do not invent files or
  exceed the budget.
```

Inspection Loop Encouraging Coding Tools.

```
Tool Configuration: code_mode: on; shell: on; web_search: disabled
```

```
Main Prompt:
{{ORIGINAL_TASK}}

Each step contains one textual instruction and one image. Don't output all
steps in one image, instead output {{STEP_COUNT}} images interleaved with text
description. Do not render the textual instructions, step labels, captions,
explanatory text, arrows, or guide graphics inside the images unless visible
text is explicitly required by the user's task.

## Required Workflow

1. Plan each intended output. Use the task and references to describe the
   content, composition, action or state, viewpoint, and continuity
   requirements of every image. Check that the descriptions form the requested
   progression and distinguish intended changes from properties that must
   remain stable. Gather any additional context from the supplied task and
   references when useful.
2. Build a task-specific visual reasoning artifact before the first
   image-generation call. Use code or a modeling tool to represent the relevant
   geometry, arrangement, state progression, motion, or causal relationships.
   Choose a suitable method, such as SVG, Canvas, Three.js, Python, or another
   renderer; no language is required. Render a meaningful visual result, not
   just code, logs, or prose.
3. Open and visually inspect the rendering. Check assumptions, spatial
   relationships, state transitions, and task constraints against what is
   actually visible. Correct empty, misleading, or inaccurate artifacts and
   inspect them again. Use the findings to refine the image descriptions. A
   successful script exit alone is not visual inspection.
4. Prepare and generate the sequence. Consolidate the descriptions, references,
   and artifact findings into clear generation instructions. Explicitly decide
   whether the inspected artifact should be supplied as an image reference;
   attaching it is optional, but constructing and inspecting it are required.
   Use the image-generation and editing tool or API assigned to the current
   baseline.
5. Inspect and revise. Examine every generated image and the complete ordered
   sequence for task compliance, direction, geometry, causal plausibility,
   visual continuity, and the final outcome. When useful, revise the plan,
   supporting context, code artifact, or generated images. Render and inspect
   changed artifacts before using them, and recheck the sequence after each
   material revision. Do not revise outputs that are already correct.
6. Select the final sequence. Stop within the shared call and time budgets,
```

```
   choose the best existing outputs, and verify their order and quantity. State
   unresolved limitations rather than bypassing the limits.

## Constraints

- Use the image-generation and editing tool or API assigned to the current
  baseline; viewing images is unrestricted. Do not switch to another image
  model or provider.
- You may use code or modeling tools to construct, render, inspect, and revise
  intermediate visual artifacts.
- All image-generation and editing attempts, including failures, share one
  per-Case budget of 2 × {{STEP_COUNT}} calls ({{MAX_IMAGE_CALLS_2N}}
  submissions). Initial generations, regenerations, and edits all count. Never
  reset or bypass this budget.
- Non-image-generation tool use does not consume image-generation calls, but
  must remain within the Case time limit. Any image-generation or editing API
  call counts regardless of its purpose.
- At the cap, make no further image-generation or editing calls. Select from
  existing outputs and report any remaining shortfall.
- You have 150 minutes total, with reminders every 15 minutes.
- Use the supplied task and references and your own work from this Case. Do not
  read other Cases, sessions, baseline results, scores, Judge hints, or
  evaluation files. Do not modify source data, reference files, other runs'
  outputs, the budget gate, ledger, or configuration, or start other sessions.

## Artifact Records

Save the source and rendered visual files in the Case workspace. Record
source_path, rendered_path, render_command, and inspection_finding in
visual_artifacts.json, along with whether the artifact is used as generation
context. For interactive or animated artifacts, save representative frames or
a video. Preserve useful versions and revision history separately from final
outputs.

Bare code, JSON, console output, prose screenshots, unchanged references,
copies of generated images, and artifacts created only after image generation
do not satisfy the pre-generation requirement. If an artifact does not help
resolve uncertainty or validate the plan, improve it before proceeding. Do not
modify source data, reference files, or other runs' outputs.

## Output Contract

- Return exactly {{STEP_COUNT}} accepted final images in step order,
  interleaved with text, using absolute Markdown image paths.
- Include only selected final images, not code artifacts, discarded drafts, or
  superseded revisions. Verify that the image files exist and that the output
  order and count are correct.
- If the required sequence cannot be completed within the budget, return the available
  selected images and briefly state the shortfall; do not invent files or
  exceed the budget.
```