VGI-BENCH: Probing Visual Intelligence in Video Generation Models

Xuan He[1] †§, Cong Wei[3,6] †, Yuhao Cheng[1] †, Linrui Ma[2,4] †, Yuxuan Zhang[5,6,10] †, Zuojun Li[2], Jize Jiang[1], Yuhao Wen[2], Zeyi Liu[1], Yuren Hao[1], Songcheng Cai[3,6], Keming Wu[2], Penghui Du[10], Kai Zou[9], Rui Yang[1], Chenkai Sun[8], Ke Yang[1,7], Ping Nie[3], Kelsey R. Allen[5,6], Chenglong Wang[7], Michel Galley[7], Jianfeng Gao[7], ChengXiang Zhai[1]

[1]University of Illinois Urbana Champaign, [2]Tsinghua University, [3]University of Waterloo, [4]Massachusetts Institute of Technology, [5]University of British Columbia, [6]Vector Institute, [7]Microsoft Research, [8]Independent, [9]NetMind.ai, [10]Etude AI,

Project Page Data Code

Abstract

Recent studies suggest that video generation models can exhibit certain forms of zero-shot visual reasoning through generated frames. Yet reliable evaluation remains challenging: benchmarks should adopt inputs aligned with the visual priors of current video models, require valid evolving processes rather than only plausible final states, and calibrate task difficulty to remain challenging yet partly feasible. To this end, we introduce **VGI-BENCH**, containing 27 tasks and 810 instances, organized by a two-level taxonomy of task domains and skill tags for fine-grained evaluation of visual reasoning capabilities of video generation models. Our evaluations show that current generative systems can solve a subset of visually grounded reasoning tasks, but remain far from reliable, with even the strongest model, Seedance 2.0, achieving only 51.0% under our evaluation criteria. Our analysis further explore the output failure modes, input condition sensitivity, performance transfer boundary from synthetic fine-tuning, and internal denoising perspective revealing limited self-correction, where later steps mainly refine early hypotheses rather than correct reasoning errors. We hope **VGI-BENCH**will help stimulate the development of next-generation video generation models.

1 Introduction

video generation models are increasingly viewed as visual world simulators (OpenAI, 2024; Bruce et al., 2024; Qin et al., 2024; Huang et al., 2025), capable of synthesizing plausible evolutions of visual scenes. Recent studies (Wiedemer et al., 2025; Tong et al., 2025) further suggest video generation models may encode more than low-level world priors like appearances and motions: they can acquire structured representations of spatial-temporal relations, rule constraints, and action-outcome dependencies from large-scale training, enabling certain forms of zero-shot visual reasoning. This has motivated a broader view of video generation models, from **passive visual simulators** toward **potential visual reasoners** that express reasoning through frame sequence. Beyond video synthesis itself, such emergent visual intelligence suggests its potential as vision foundation models, with implications for tasks including scene understanding, controllable editing, and downstream embodied learning (Gabeur et al., 2026; Wang et al., 2026a; Yang et al., 2025b; Zheng et al., 2026; Liang et al., 2025; Gao et al., 2026; Ye et al., 2026). These developments raise two key questions: how much visual intelligence is encoded in current video generation models, and how well these models can solve downstream tasks by "imagining" their step-by-step progression through generated video frames.

Bench	Reasoning Demand	Appearance Abst., Real.	Process-sensitive No, Yes	Difficulty Control
PhysGenBench	Low / Mid	●	●	✓
WorldSimBench	Low / Mid	●	●	✓
TiVi-Bench†	High	◔	◔	✓
V-ReasonBench	High	◔	◔	✗
VBVR-Bench	High	●	◔	✗
Ours	High	●	●	✓

†Data has not been publicly released and statistics are inferred from the paper.

Table 1: Comparison with related video benchmarks. Each pie encodes the fraction of tasks that satisfy a desideratum versus do not. For input appearance, Real. denotes photorealistic-style inputs, while Abs. denotes synthetic inputs, like line-art or schematic images. Appendix E.1 details the ratio assignment.

Answering these questions requires benchmarks that go beyond visual fidelity and test whether video models can use their learned visual priors for reasoning. Recent efforts have begun to evaluate

† Main Contributor. § Project Lead.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 40335–40371
October 24-29, 2026 ©2026 Association for Computational Linguistics

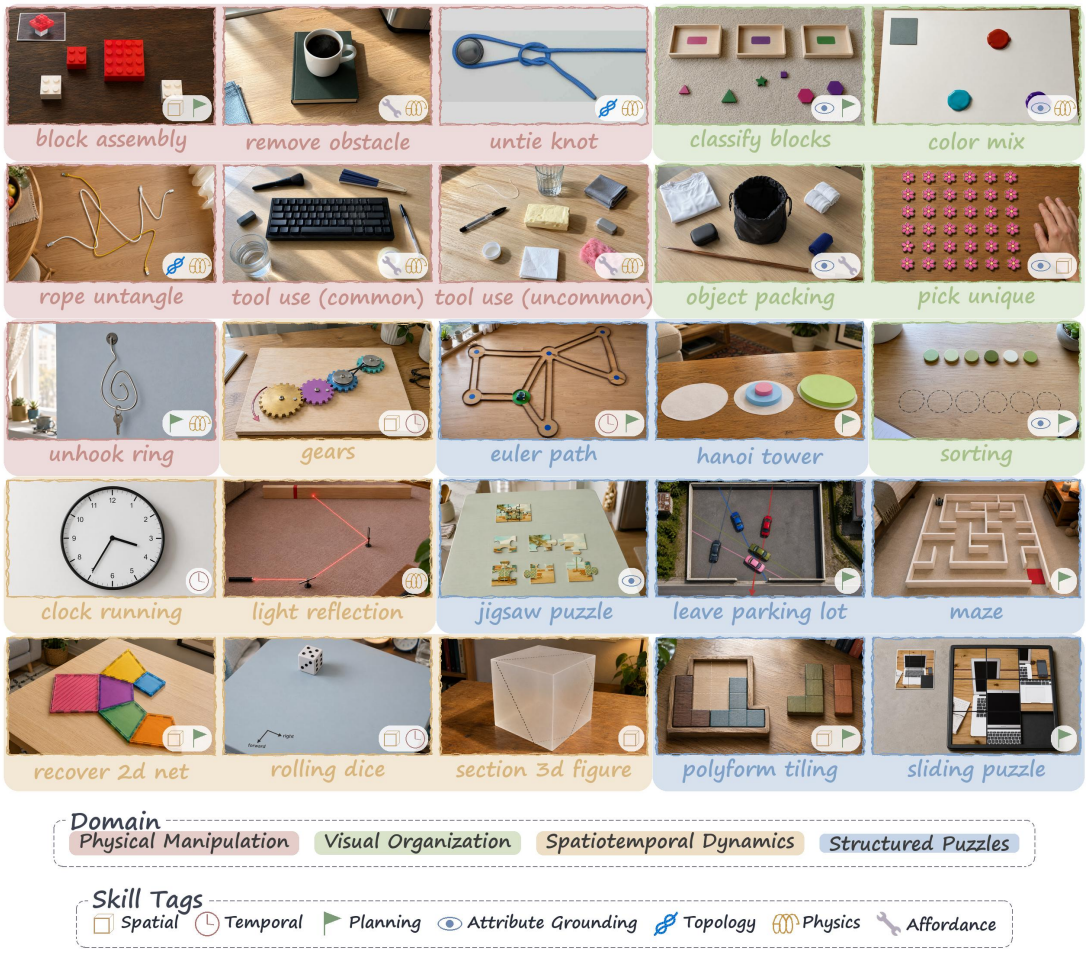


Figure 1: Overview of representative tasks in our benchmark. **VGI-BENCH**adopts a **double-level taxonomy**: the first level groups every task into one of four mutually exclusive task **domains**, while the second level annotates each task with one or more (non-exclusive) **skill tags**, as shown in the legend at the bottom. Each panel shows a representative real-scene input image; the icons under the task name encode that task's skill tags.

video models as zero-shot visual reasoners (Chen et al., 2025; Liu et al., 2025b; Wang et al., 2026b; Luo et al., 2025a), but still leave several important gaps: 1 **Distribution-mismatched visual appearances**. Many benchmarks use line-art or abstract inputs for scalability and controllability, but such inputs can deviate far from the natural-image priors of video generation models. In our controlled comparisons, abstract inputs more often lead to collapse and constraint ignorance than the visually realistic counterparts. Such failures may reflect visual domain mismatch more than reasoning limitations, weakening validity of existing evaluations. 2 **Limited demand for visual rollout reasoning**. Many existing visual reasoning or visual QA tasks can be answered directly from the input, without requiring the model to simulate how the scene evolves. Therefore, they do not adequately evaluate whether a video generation model can solve a task by explicitly rolling out its visual progression and grounding the final answer in the generated trajectory. 3 **Uncontrolled task difficulty and feasibility**. Existing benchmarks include tasks that are far beyond their feasible regime, making failures less diagnostic. Examples include long-horizon tasks exceeding practical video duration and knowledge-heavy tasks relying on non-visual domain expertise like medical knowledge. A more diagnostic evaluation should calibrate task feasibility near the current capability boundary and organize tasks into graded difficulty levels. The limitations above are summarized in Table 1 and detailed in Appendix E.1.

To bridge these gaps, we introduce **VGI-BENCH**, a benchmark for evaluating visual intelligence in video generation models through meticulously designed downstream tasks. Our benchmark addresses previous issues by 1 using photorealistic-style inputs to reduce visual-domain mismatch, 2

filtering tasks whose success depends on valid intermediate trajectories rather than final states alone, and 3 calibrating difficulty through pre-generation filtering and human review to keep tasks challenging yet partially feasible for current models, as detailed in Section 3. **VGI-BENCH**further adopts a two-level taxonomy covering both task domains and skill requirements. Each task is assigned to one mutually exclusive domain based on its visual characteristics, and then annotated with one or more skill tags, as in Figure 1.

Leveraging **VGI-BENCH**, we systematically evaluated a wide range of representative video models to understand their reasoning capacity. The results show that the current generative models exhibit both emerging reasoning abilities and substantial gaps toward general-purpose visual intelligence. Even the strongest model, Seedance 2.0, achieves only 51.0 under our criteria. Current models can make partial progress on visual goals, but often fail to maintain coherent multi-step execution, with common failure modes including physical collapse, rule violation, and object/state inconsistency. Beyond performance, our diagnostic analyses examine video reasoning failure from several complementary angles. At the inference level, input conditions like prompts and visual styles substantially affect performance, with open-source models especially sensitive to the visual style gaps. At the training level, large-scale synthetic fine-tuning can transfer from abstract data to realistic tasks, but the gains are bounded by how well the training distribution covers the skill requirements. At the internal level, denoising trajectories show that current video models tend to refine early visual hypotheses, while reliable self-correction of erroneous states remains limited. Together, these analyses provide a more detailed view of reasoning behavior and help identify the factors that shape, limit, and potentially improve it in video generation models.

In summary, our contributions are threefold: 1 **VGI-BENCH**: a benchmark for evaluating visual intelligence in video generation models around their current capability boundary. 2 Broad evaluation of contemporary generative models covering both image and video generation, showing their emerging ability while exposing limitations on general-purpose intelligence. 3 Multi-faceted analyses of video model reasoning, covering failure modes, input sensitivity, performance transfer of synthetic fine-tuning , and denoising dynamics, yielding insights into the factors that shape and limit visual reasoning performance.

2 Related Works

2.1 Video Generation Models

Video generation models were first developed and evaluated primarily as content creation systems, with progress measured by visual quality, motion realism, and condition alignment (Yang et al., 2025c; Kong et al., 2024; Wan et al., 2025). As temporal coherence and physical plausibility improve, they are increasingly viewed as visual world simulators: generated videos can serve as explicit predictions of how scenes, objects, and interactions evolve over time (Brooks et al., 2024; Qin et al., 2024). While more recent studies further suggest another role beyond the above: they may act as zero-shot visual reasoners, expressing solutions through generated frame sequences (Wiedemer et al., 2025; Tong et al., 2025), showing a promising paradigm for multi-modal reasoning.

2.2 Probing Reasoning in Video Generation

Studies suggest video generation models can exhibit non-trivial zero-shot reasoning through generated frames (Wiedemer et al., 2025; Tong et al., 2025), showing the potential of visual reasoners beyond simulators. This motivated a series of follow-up evaluations, including TiVi-Bench (Chen et al., 2025), V-ReasonBench (Luo et al., 2025a), and MMGR (Cai et al., 2025a), which evaluate generative reasoning across spatial, physical, logical, and other tasks. VBVR (Wang et al., 2026b) further scales this direction from evaluation to adaptation, pairing a large task collection with benchmark-specific supervision for LoRA fine-tuning. Complementary to these efforts, **VGI-BENCH**evaluates the reasoning capability encoded in video generation models through naturalistic and goal-directed procedural tasks with feasibility controls.

3 VGI-BENCH

3.1 Taxonomy

To cover different aspects of visual intelligence, we propose a two-level taxonomy: **Domains** and **Skill-tags**. The first level organizes tasks into four mutually exclusive domains from their visual characteristics: **Visual Organization** tasks require models to arrange, group, or select objects based on visual attributes and cues. **Physical Manipulation** tasks involve object-level actions such as moving, placing, stacking, or using tools, where success depends on plausible physical interaction.

Structured Puzzles focus on rule-governed visual puzzles, where models must follow explicit constraints to transform an initial state into the valid target. **Spatiotemporal Dynamics** tasks require reasoning about how states evolve over time, including ordering and temporal dependency.

Beyond domain-level organization, we further annotate each task with one or more **Skill Tags** that capture the underlying capabilities for solving it. These tags are inspired by visual cognition theories and provide a capability-level view complementary to the domain taxonomy. Skill tags are non-exclusive and seven tags are used: **Spatial**, **Temporal**, **Planning**, **Attribute Grounding**, **Physics**, **Topology**, and **Affordance**. This two-level design enables both coarse- and fine-grained diagnosis of model capabilities.

3.2 Task Collection

Task Proposal. Our tasks are designed to evaluate visually grounded reasoning processes that can be naturally expressed through video. Each task follows a unified I/O format: the model receives a text prompt and an input image as the first frame, then generates a video completing the specified visual procedure. We focus on **reasoning-intensive** objectives and avoid low-level recognition or localization tasks. Each task is required to be **process-sensitive**, where success depends on intermediate state evolution and rule-preserving trajectory, rather than final-state correctness alone. We also constrain the expected solution length to match the typical generation duration of current video models.

Each task is instantiated at three difficulty levels, with roughly ten instances per level. Every instance consists of an input image, a text prompt, and task-specific evaluation criteria 3.4. The resulting suite is representative rather than exhaustive: we prioritize tasks near the capability boundary of current models, yielding sharper diagnostic signals for current systems while remaining meaningful and challenging for future models.

Input Image and Prompt. Each instance contains an input image and a text prompt. Input images are collected from web images or existing datasets, or generated with image generation models such as GPT-Image-2 (OpenAI, 2026) and Nano Banana Pro (Google, 2025c). For generated inputs, we use a human-in-the-loop process: since image generation models may sometimes introduce unintended artifacts, images are manually reviewed and iterated until matching the intended task design. All images are standardized to a 16:9 aspect ratio. The text prompt specifies the task goal and the constraints for the generated videos. It describes the relevant objects, attributes, allowed actions, prohibited shortcuts, and task-specific rules that must be preserved during generation. We also includes task-agnostic controls on background, layout, camera motion, and video speed, so that the generated video remains focused on the intended procedural reasoning rather than irrelevant visual variation.

Reference Solution. Each task is paired with a reference solution specifying the intended outcome. The reference may take the form of an image or a textual description. Image references illustrate an acceptable solution, such as highlighting a valid path in MAZE task. For tasks whose solutions are better specified semantically, we provide a description for the desired final state, such as task UNTIE KNOT. The references define the target outcome and are used to guide both task proposal and the criteria construction in Section 3.4.

3.3 Quality Control

Pre-Generation. To calibrate task difficulty, we add a pre-generation stage. For each proposed task, we sample two easiest-level instances and test them on several sota video generation models, such as Sora2, Veo3.1, Kling3.0, etc. A task is accepted only if the sampled instance is solved by at least one model and failed by at least one model; otherwise, we revise its design and input materials. This procedure filters out tasks that are trivially solvable or entirely infeasible for current models, making them *challenging yet partly feasible*. We emphasize this stage serves as a sanity filter: it checks whether a task can plausibly be expressed as a video process, rather than certifying full solvability.

Manual Review. We manually review each task instance: whether it ① remains faithful to the intended goal, ② fits within the typical video duration limit like 5-10s, ③ is described by a clear prompt. Unqualified instances are revised or discarded.

3.4 Evaluation Criteria

Since our tasks are process-sensitive, evaluation must assess both **goal completion** and **process validity**. A correct-looking final state is insufficient if the trajectory violates task rules, while a locally plausible video may still fail by making little progress toward the goal. We therefore propose two complementary metrics.

Completeness (Comp.) This metric captures

Category	Level	*Commercial*						*Open Source*		
		Sdce2.0	**Sora2**	**Veo3.1**	**Kling3.0**	**Wan2.7**	**Gen4.5**	**Mnx-H3**	**HY1.5**	**Wan2.2**
Visual Organization	Easy	72.6	55.6	50.3	67.6	37.0	60.5	42.8	26.6	30.4
	Mid	56.4	37.0	43.3	47.5	32.2	38.3	42.3	17.9	15.3
	Hard	53.5	39.1	44.0	43.7	32.5	41.9	39.8	24.7	18.5
	Avg.	60.8	43.9	45.9	52.9	33.9	46.9	41.6	23.1	21.4
Spatiotemporal Dynamics	Easy	54.6	45.3	28.1	45.2	38.9	39.6	44.3	34.7	38.7
	Mid	47.4	34.6	21.2	35.2	42.6	34.6	40.7	27.2	32.0
	Hard	33.4	24.4	16.9	28.2	28.9	29.5	35.9	23.9	19.9
	Avg.	45.3	34.8	22.3	36.5	36.8	34.8	40.3	28.7	30.2
Structured Puzzles	Easy	46.9	40.5	31.8	45.3	25.6	29.2	51.9	9.9	17.1
	Mid	45.1	25.8	21.8	38.3	23.3	20.7	52.3	8.8	8.4
	Hard	41.8	22.4	13.5	28.9	24.7	18.7	47.6	7.9	5.6
	Avg.	44.6	29.5	22.4	37.5	24.5	22.9	50.6	8.9	10.4
Physical Manipulation	Easy	64.4	47.9	49.6	60.6	55.8	52.1	58.6	22.5	33.7
	Mid	59.2	41.8	40.4	51.2	48.2	43.4	41.5	16.7	22.1
	Hard	44.4	32.6	37.3	45.6	37.1	39.5	33.0	11.9	17.5
	Avg.	56.0	40.8	42.5	52.5	47.1	45.0	44.4	17.0	24.4
Overall		51.0	36.7	32.0	44.0	35.7	36.6	44.4	19.1	21.6

Table 2: Evaluation results of video generation models, the best and second-best scores are highlighted. HY1.5 for `HunyuanVideo-1.5`, Sdce2.0 for `Seedance2.0`, Mnx-H3 for `MiniMax-H3`.

the global progress toward the task goal. Since different tasks have different goal states, we define a task-specific tiered standard and map the video to one of three levels: `<complete>`, `<partial>`, or `<failed>`. The VLM-judge receives a set of uniformly sampled frames (2fps) together with the standard and returns the corresponding tier.

Rubric Score (Rub.) This metric measures local process validity throughout the video. We design a fine-grained checklist for each task that covers explicit rules and other constraints. Since some critical violations may occur briefly, we use a coarse-to-fine adaptive frame sampling strategy. The VLM-judge starts from coarse sampling (4fps), inspecting the video against the checklist and flagging intervals that require closer inspection (the per-window calls only localize evidence; the final violation counts are assigned by a later aggregation step). These intervals are then resampled at a finer rate (8fps) and re-evaluated, allowing the judge to capture transient violations without densely sampling the entire video. To keep each judgment focused, we further adopt a sliding focus window (10-frame) with edge frame overlapping, so that only a small local segment is inspected at a time rather than all the sampled frames.

Each rubric item is scored by an inverse decay penalty, $1/(x+1)$, where x is the number of violations of this item. The inverse decay reflects the intuition that once a violation occurs repeatedly, additional occurrence should have diminishing marginal effect. The Rubric Score is the average over all item-level scores. Other monotonic decay functions like exponential decay are also applicable and preserve the same qualitative trend.

Aggregation. The **Final Score** combines global completion and local process validity by multiplying Comp. with Rub.. This multiplicative design penalizes either type of failure. An almost static video preserves most local constraints (high Rub.) but make little progress towards task goal(low Comp.); Conversely, a video may appear to reach the target state (high Comp.) while violating the rules (low Rub.). The aggregation therefore treats both as jointly necessary conditions, preventing either aspect alone from dominating the evaluation.

Appendix A.2 shows examples of criteria for Comp. and Rub., while Appendix C.1 provides implementation details of the VLM-based evaluator.

3.5 Data Augmentation

Image Output Adaptation. In order to extend **VGI-BENCH**as a testbed for reasoning in image generative models, we repurpose some tasks into single-image output format while preserving task goal. This adaptation does not contradict the process-sensitive task design: the original video tasks evaluate whether a video model can express the procedure through temporal state evolution,

while the image version asks whether an image model can infer and render the target state or visual solution. For example, MAZE is adapted into drawing a valid path from start to the goal, and RECOVER 2D NET is adapted into rendering the completed 3D structure. This branch enables comparison of static and procedural reasoning under related task goals. More details are in Appendix A.3.

4 Experiments

4.1 Setup

Our evaluation covers a broad suite of closed-source and open-source video models, together with image models on the adapted subset from Section 3.5. Due to the evaluation cost, we evaluate half of the instances for each task using a fixed random seed. Appendix B.4 further analyzes stability of the half-instances protocol compared to full-set evaluation. Following the criteria in Section 3.4, We use Gemini-3-Flash (Google, 2025a) for our automatic evaluator. The full model list and generation configurations are in Appendix B.1.

4.2 Results

Commercial video models lead, but all remain far from solved. Table 2 reports the video generation model results. Commercial models consistently outperform open-source models, with `Seedance-2.0` achieving the best overall score (51.0); nevertheless, all tasks remain far from solved. **Structured Puzzles** is the most challenging domain, exposing failures in multi-step rules and state tracking. Figure 2 further shows that **Topology** and **Temporal** are the weakest skill dimensions (one task may have several skill tags), reflecting failures in connectivity preservation and multi-step state tracking. To complement the main metrics, we report a strict success rate (S.R.), where an instance is counted as successful only if both completeness and rubric scores are perfect. Besides, we also conduct a human study for estimating the benchmark ceiling to demonstrate the gap between generative models and average humans. Detailed results are in Appendix B.2 and B.3.

Image generation serves as a static goal-state diagnostic. We evaluate image models on an output-adapted subset, measuring how well the image models can realize the static goal-state when temporal process validity is removed. Table 4 shows a similar commercial–open-source gap, led by `Nano-Banana-Pro` (Avg. 55.0) and

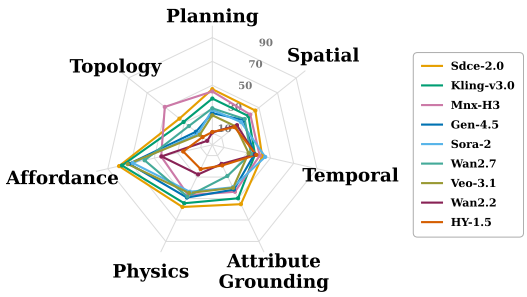


Figure 2: Per-model performance across the skill tags.

`Seedream-5.0-Pro` (Avg. 52.1). The consistent drop from easy to hard suggests meaningful difficulty gradients of our tasks. However, this setting only tests target-state inference and rendering; it does not evaluate valid intermediate process, which remains the focus of our video benchmark.

4.3 Evaluation Reliability

The reliability of our VLM-based evaluator is assessed by comparing it against human annotations and ablating the key components: adaptive frame sampling and the sliding focus window. Also, we report the ablation results of base model with comparable pricing level. Table 3 reports AUC and pairwise accuracy against human preferences for the full method and its two ablations. The full evaluator achieves the strongest agreement with human judgments, while removing either component leads to degradation, confirming the necessity of both. Details of correlation analysis are in Appendix C.3.

Judge Design	AUC	Pairwise Acc.
Main (w/ `Gemini-3-Flash`)	**0.803**	**73.2%**
w/o adaptive fps	0.772	69.5%
w/o focus window	0.753	68.3%
w/ `GPT-5-mini`	0.690	64.3%
w/ `Claude-Haiku-4.5`	0.478	47.9%
w/ `Qwen3.6-Plus`	0.624	59.1%

Table 3: Reliability of our VLM-based evaluator compared with human annotations, including key component and base model ablations.

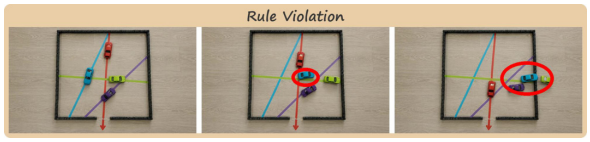
5 Discussion

Beyond aggregated scores, we further diagnose video reasoning across four stages: output failures, input conditions, training-time transfer, and internal denoising dynamics. This progression examines not only where the model fails, but also how their behavior changes across multiple aspects.

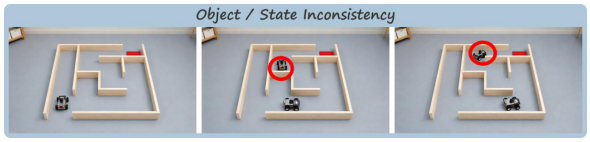
	Commercial						*Open Source*				
	Nano-Banana-Pro	**Qwen-Image-3-Pro**	**GPT-Image-2**	**MAI-Image-2.5-Pro**	**Seedream 4.5**	**Flux.2 Max Edit**	**Sense-Nova-U1**	**JoyAI-Image**	**Qwen-Image-Edit**	**BAGEL**	**Step1X-Edit**
Easy	62.5	51.9	48.8	44.9	27.5	23.8	22.2	11.2	11.2	1.2	1.2
Mid	50.0	43.8	38.8	35.1	21.2	20.3	14.4	10.0	5.0	1.2	0.0
Hard	52.5	32.5	36.2	30.3	22.5	17.7	12.2	10.0	6.2	0.0	1.2
Avg.	55.0	42.7	41.2	36.8	23.8	20.6	16.3	10.4	7.5	0.8	0.8

Table 4: Evaluation results of image models on the adapted subset. Success Rate (%) is measured and reported (different from the video branch), against the ground truth image, the best and second-best scores are highlighted.

(a) **Physical collapse.** The laptop is tilted up, yet **the cup lying on its lid does not roll down**.



(b) **Rule violation.** Each car must stay on its own colored track and may not cross walls; **the video violates both rules**.



(c) **Object/state inconsistency.** While the car drives through the maze, **a spurious second car appears partway through**.

Figure 3: Representative cases in the failure modes.

5.1 Failure Modes

We firstly examine representative failure cases and summarize several recurring failure modes, as shown in Fig 3. 1 **Physical collapse.** Generated videos sometimes contain unrealistic deformation, object penetration, or sudden disappearance. Under strong task-goal or constraint pressure, physical causality becomes fragile and may be sacrificed for a goal-like visual outcome, suggesting current models may encode local visual dynamics, but still struggle to maintain physically coherent processes under goal-directed generation. 2 **Rule violation.** Models may generate videos that remain physically plausible and coherent, yet violate the explicit rules. For example, they may perform prohibited actions or skip essential intermediate steps. In some cases, the model reaches a visually plausible final state by directly altering the target scene, rather than following the expected procedure. 3 **Object/state inconsistency.** Models often fail to preserve object identities, positions, or intermediate states over time. Objects may disappear, transform, or reset to earlier states, even when the overall motion appears smooth. This reveals weak temporal state tracking, which is critical for goal-directed process generation. See more examples in Appendix D.1.

5.2 Input Condition Sensitivity

We probe input robustness of video reasoning along two axes: **text prompt** with varying levels of detail and **input image** in different visual styles. For both aspects, we evaluate a fixed subset of tasks.

Oracle Prompting. Prompt optimization is widely used to improve the quality and controllability of visual generation. We test its potential ceiling for video reasoning by building an *oracle prompt* describing the intended solution as explicitly as possible. This reduces the high-level reasoning burden and probes whether the model can follow a goal-directed solution description and render the corresponding visual process. As shown in Figure 5, oracle prompting improves performance for some models, especially closed-source ones, but the gains remain limited. Even with the full solution provided, video models often fail to render the complete solution trajectory. This reflects two bottlenecks: some solutions are intrinsically difficult to specify precisely in language, especially when involving fine-grained spatiotemporal transitions; and current video models still struggle with instruction following and physical simulation under strong rule constraints. The performance gain is particularly weak for `HunyuanVideo 1.5`, whose lower base reasoning capability leaves little room for oracle prompts to help. Details on task and model selection are provided in Appendix D.2.

Visual Style. Using the metadata of selected tasks, we generate line-art variants that preserve the original task structure, and compare performance against the realistic-style setting. As shown in Figure 5, model rankings under line-art inputs differ noticeably from realistic-style. This discrepancy is especially pronounced for open-source models,

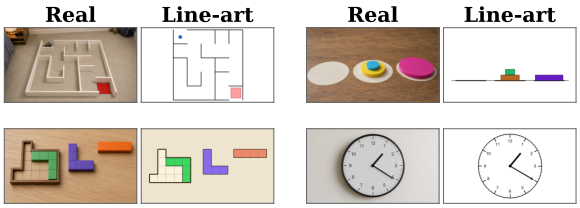


Figure 4: Style variants (Real / Line-art) from 4 tasks: maze, hanoi tower, polyform tiling, clock running.

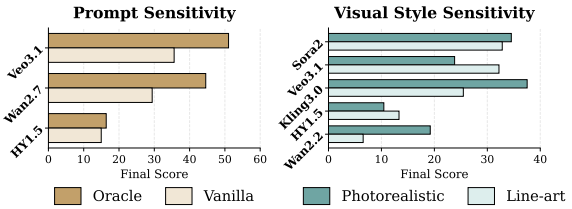


Figure 5: Performance comparison in oracle prompting and in varying input visual styles.

indicating stronger sensitivity to input visual appearance. These results also support one of our motivations: visual style can substantially affect the measured performance, so evaluations dominated by abstract inputs may conflate reasoning limitations with visual-domain mismatch. See more details in Appendix D.2.

5.3 How Well does Scaling Tuning Transfer?

Given that pretrained video models already encode useful visual priors, an interesting question is how much downstream fine-tuning can further improve the reasoning capability (Wang et al., 2026b; Zhu et al., 2026; Chen et al., 2026b). This question becomes more practical when considering data scalability: curating large-scale real videos with explicit reasoning goals is expensive, and synthetic data offers a more controllable and scalable alternative, but it remains unclear whether such scaling can systematically improve video reasoning in realistic scenarios. We therefore study it through a synthetic-data scaling setting, by comparing the released VBVR models (Wang et al., 2026b), fine-tuned on 1M-sample abstract-style dataset, with its respective base models `Wan2.2-I2V-A14B`, `Wan2.1-I2V-14B`, and `LTX-2.3`.

Table 5 groups our tasks by the structural overlap with the VBVR training data distribution, with details and examples in Appendix D.3. At the group level, performance gains decrease with structural overlap for all three base models, indicating that synthetic fine-tuning transfers more effectively across aligned task structures. However, this trend

Model	Overlap	Semi-overlap	Non-overlap
`Wan2.2-I2V` (base)	15.3	17.8	29.2
`VBVR-Wan2.2`	**55.8** ↑40.5	**34.9** ↑17.1	**35.4** ↑ 6.2
`Wan2.1-I2V` (base)	11.3	17.0	25.6
`VBVR-Wan2.1`	**25.1** ↑13.8	**20.2** ↑ 3.2	22.0 ↓ 3.6
`LTX-2.3` (base)	15.0	18.2	15.5
`VBVR-LTX2.3`	**24.1** ↑ 9.1	**23.6** ↑ 5.4	**19.2** ↑ 3.7

Table 5: Performance of the three released VBVR models and their respective base models, grouped by structural overlap with training distribution. Per-cell values are mean final score across tasks in the group.

is not uniform at the task level: overlap does not guarantee reasoning improvement, while some non-overlap tasks still benefit. For example, the task UNTIE_KNOT shows little change in overall success rate, yet its rubric score rises substantially from 0.02 to 0.34. We attribute this to more controlled behavior in the video and fewer rule violations after the fine-tuning, together with limited transfer of basic spatial or logical capabilities. More results and details are provided in Appendix D.3.

Table 6 further shows the domain- and skill-level breakdown and the uneven gains. Improvements are concentrated on skills well represented in the synthetic training set, such as planning and spatial reasoning, while capabilities like physical interaction or strong temporal dependency remain hard to improve and may even degrade.

Model	Structured Puzzles	Visual Organization	Spatiotemporal Dynamics	Physical Manipulation	Overall
`Wan2.2-I2V`	11.6	21.4	30.2	24.4	21.5
`VBVR-Wan2.2`	53.2 ↑41.6	37.6 ↑16.2	29.0 ↓1.2	42.6 ↑18.1	41.2 ↑19.7

Model	Planning	Spatial	Temporal	Attribute Grounding
`Wan2.2-I2V`	10.4	26.4	37.0	17.8
`VBVR-Wan2.2`	44.1 ↑33.7	37.3 ↑10.9	25.8 ↓11.2	37.9 ↑20.1

Model	Physics	Affordance	Topology
`Wan2.2-I2V`	27.6	43.7	5.4
`VBVR-Wan2.2`	39.9 ↑12.3	55.2 ↑11.5	26.3 ↑20.9

Table 6: Per-domain (top) and per-skill (bottom) performance (final score) breakdown for `VBVR-Wan2.2` vs. base `Wan2.2-I2V`.

These results demonstrate the promise of synthetic data as a scalable source of supervision, while also revealing that its transfer is bounded by the structural coverage of the training distribution, leaving more effective scaling strategies and the transfer limits of video reasoning for future exploration.

5.4 When Is Visual Reasoning Decided?

In the section, we go beyond the performance and examine reasoning behavior along the denoising trajectory. Recent work on diffusion large language models (dLLMs) (Ye et al., 2024; Zhao et al., 2026a; Nie et al., 2026) connects iterative denoising with reasoning behaviors such as "self-consistency" and "self-correction", where the latter refers to revising incorrect intermediate states in the later denoising steps. This perspective has also been extended to video generative reasoning (Wang et al., 2026c), suggesting that the reasoning may unfold along denoising steps and that self-correction may emerge during the generation.

Transition	1→2	2→3	3→4	4→10	10→20	20→40
♠ Unrecognizable	69.2	41.9	22.2	6.8	0.0	0.0
♣ Stable	17.9	49.6	70.1	69.2	75.2	90.6
■ (a) Correct → Wrong	2.6	0.9	0.0	0.9	0.0	0.0
■ (b) Wrong → Correct	0.9	0.0	0.9	0.0	0.0	0.0
■ (c) Wrong → Wrong′	9.4	7.7	6.8	23.1	24.8	9.4

Table 7: Distribution (%) of solution-state transitions between consecutive decoded denoising steps. Row ■ (b) (shaded) is self-correction. Protocols are in Appendix D.4.

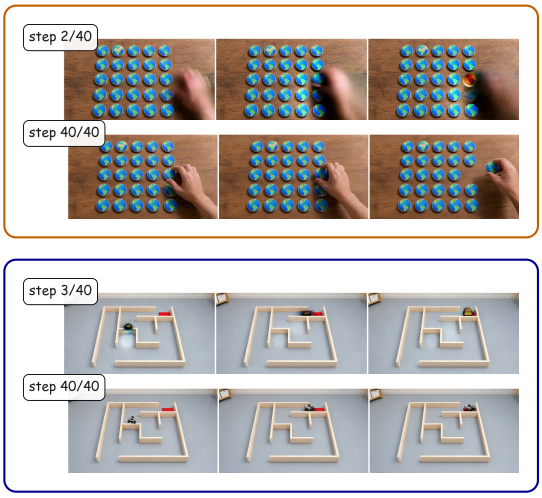


Figure 6: Decoded frames at different denoising stages.

Our empirical analysis offers a more cautious view. We decode intermediate denoising states from four open-source models and label how the visible solution state changes between consecutive decoded checkpoints: ♠ unrecognizable, ♣ stable, or ■ changed, with changes split into correct→wrong, wrong→correct (i.e. self-correction), and wrong→wrong′ (one incorrect hypothesis replaced by another). More details about the protocol are detailed in Appendix D.4.

As shown in Table 7, the solution state does change during denoising, but almost never toward a correct one. Self-correction stays below 1% everywhere and stops occurring in later steps, while wrong→wrong′ is an order of magnitude more frequent (23.1% at 4 → 10 and 24.8% at 10 → 20). Once the state is readable at all, it mostly stays put, with stability rising to 90.6% over the second half of denoising. Revision therefore moves between wrong solutions rather than toward the right one; later steps mostly lock in and refine the early hypothesis (Newman et al., 2026; Zhu et al., 2026), even when it already violates task rules, as shown in Figure 6. Both these quantitative and qualitative results provide a more grounded view of reasoning behavior in video generation.

6 Conclusion

We introduce **VGI-BENCH**, a benchmark for evaluating visual intelligence in video generation models. With realistic-style inputs, process-sensitive task design, and calibrated difficulty levels, **VGI-BENCH**provides a diagnostic testbed for assessing whether video models can solve tasks through valid visual rollouts. Extensive evaluation shows current models exhibit emerging reasoning ability, but still far from general visual intelligence. Further analyses reveal sensitivity to input conditions, bounded transfer from scaling synthetic fine-tuning, and limited self-correction during denoising process. We hope **VGI-BENCH**can support more systematic diagnosis and development of future video and multimodal foundation models.

Limitations

Our benchmark has several scope-related limitations we want to make explicit. First, all tasks are designed around the typical generation length of current video models (roughly 5–10s); longer-horizon procedural reasoning, such as multi-minute multi-step assembly or long-trajectory planning, is therefore out of scope. Second, the benchmark covers only the image-to-video (i2v) setting with a fixed 16:9 aspect ratio; text-to-video, multi-image conditioning, and audio-conditioned generation are left to future work. Third, all task prompts and rubrics are written in English, so multilingual or cross-lingual evaluation is not addressed. Finally, the task suite is intentionally *representative rather than exhaustive*: it captures a focused slice of visual reasoning domains and difficulty levels, but

does not aim to enumerate every conceivable visual reasoning scenario, and may need to be extended as model capabilities improve.

References

Alibaba Cloud. 2026. Qwen-Image-3.0: Text-to-image and image editing api reference. https://www.alibabacloud.com/help/en/model-studio/qwen-image-generation-and-editing-api-reference.

Black Forest Labs. 2025. FLUX.2 [max].

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Wing Yin Ng, Ricky Wang, and Aditya Ramesh. 2024. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/.

Jake Bruce, Michael D 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, and 6 others. 2024. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

ByteDance Seed. 2026. Seedream 4.5.

Zefan Cai, Haoyi Qiu, Tianyi Ma, Haozhe Zhao, Gengze Zhou, Kung-Hsiang Huang, Parisa Kordjamshidi, Minjia Zhang, Wen Xiao, Jiuxiang Gu, Nanyun Peng, and Junjie Hu. 2025a. Mmgr: Multi-modal generative reasoning. *Preprint*, arXiv:2512.14691.

Zikui Cai, Andrew Wang, Anirudh Satheesh, Ankit Nakhawa, Hyunwoo Jae, Keenan Powell, Minghui Liu, Neel Jay, Sungbin Oh, Xiyao Wang, Yongyuan Liang, Tom Goldstein, and Furong Huang. 2025b. Morse-500: A programmatically controllable video benchmark to stress-test multimodal reasoning. *arXiv preprint arXiv:2506.05523*.

Harold Haodong Chen, Disen Lan, Wen-Jie Shu, Qingyang Liu, Zihan Wang, Sirui Chen, Wenkai Cheng, Kanghao Chen, Hongfei Zhang, Zixin Zhang, Rongjin Guo, Yu Cheng, and Ying-Cong Chen. 2025. Tivibench: Benchmarking think-in-video reasoning for video generative models. *arXiv preprint arXiv:2511.13704*.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, and 10 others. 2026a. Babyvision: Visual reasoning beyond language. *Preprint*, arXiv:2601.06521.

Xinyan Chen, Ziyu Guo, Renrui Zhang, Dongzhi Jiang, and Hongsheng Li. 2026b. Opencof: Learning to reason through video generation. *Preprint*, arXiv:2607.08763.

Junhao Cheng, Liang Hou, Tianxiong Zhong, Xin Tao, Pengfei Wan, Kun Gai, and Jing Liao. 2026. Vlms are good teachers for video reasoning via adaptive test-time optimization. *Preprint*, arXiv:2606.02564.

Xuanlang Dai, Yujie Zhou, Long Xing, Jiazi Bu, Xilin Wei, Yuhong Liu, Beichen Zhang, Kai Chen, and Yuhang Zang. 2026. Endocot: Scaling endogenous chain-of-thought reasoning in diffusion models. *arXiv preprint arXiv:2603.12252*.

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

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, Yan Li, Yubo Wang, Zhijie Cao, Zhiqian Lin, Zhitao Yang, Zhongang Cai, Yuwei Niu, Yue Zhu, Bo Liu, and 39 others. 2026. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, and 10 others. 2026. Molmoact2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*.

Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T. Barron, Kyle Genova, Nithish Kannen, Sherry Ben, Yandong Li, Mandy Guo, Suhas Yogin, Yiming Gu, Huizhong Chen, Oliver Wang, and 6 others. 2026. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K.R. Zentner, and 11 others. 2026. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*.

Google. 2025a. Gemini 3 Flash: Frontier intelligence built for speed.

Google. 2025b. Generate videos with Veo 3.1 in the Gemini API.

Google. 2025c. Introducing Nano Banana Pro.

Xuming He, Zehao Fan, Hengjia Li, Fan Zhuo, Hankun Xu, Senlin Cheng, Di Weng, Haifeng Liu, Can Ye, and Boxi Wu. 2025a. Ruler-bench: Probing rule-based reasoning abilities of next-level video generation models for vision foundation intelligence. *arXiv preprint arXiv:2512.02622*.

Zefeng He, Xiaoye Qu, Yafu Li, Tong Zhu, Siyuan Huang, and Yu Cheng. 2025b. Diffthinker: Towards generative multimodal reasoning with diffusion models. *arXiv preprint arXiv:2512.24165*.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. 2025. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*.

Joowon Kim, Seungho Shin, Joonhyung Park, and Eunho Yang. 2026. Collabvr: Collaborative video reasoning with vision-language and video generation models. *Preprint*, arXiv:2605.08735.

Kling AI. 2026. Kling VIDEO 3.0 Model User Guide.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, and 33 others. 2024. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*.

Chengzu Li, Zanyi Wang, Jiaang Li, Yi Xu, Han Zhou, Huanyu Zhang, Ruichuan An, Dengyang Jiang, Zhaochong An, Ivan Vulić, Serge Belongie, and Anna Korhonen. 2026. Thinking in frames: How visual context and test-time scaling empower video reasoning. *arXiv preprint arXiv:2601.21037*.

Yifan Li, Yukai Gu, Yingqian Min, Zikang Liu, Yifan Du, Kun Zhou, Min Yang, Wayne Xin Zhao, and Minghui Qiu. 2025. Viper: Process-aware evaluation for generative video reasoning. *arXiv preprint arXiv:2512.24952*.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. 2025. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, and 5 others. 2025a. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*.

Xinxin Liu, Zhaopan Xu, Ming Li, Kai Wang, Yong Jae Lee, and Yuzhang Shang. 2025b. Can world simulators reason? gen-vire: A generative visual reasoning benchmark. *arXiv preprint arXiv:2511.13853*.

Yang Luo, Xuanlei Zhao, Baijiong Lin, Lingting Zhu, Liyao Tang, Yuqi Liu, Ying-Cong Chen, Shengju Qian, Xin Wang, and Yang You. 2025a. V-reasonbench: Toward unified reasoning benchmark suite for video generation models. *Preprint*, arXiv:2511.16668.

Yang Luo, Xuanlei Zhao, Baijiong Lin, Lingting Zhu, Liyao Tang, Yuqi Liu, Ying-Cong Chen, Shengju Qian, Xin Wang, and Yang You. 2025b. V-reasonbench: Toward unified reasoning benchmark suite for video generation models. *arXiv preprint arXiv:2511.16668*.

Microsoft AI Superintelligence Team. 2026. Introducing MAI-Image-2.5-Pro and MAI-Voice-2-Flash.

MiniMax. 2026. MiniMax H3: Multimodal video generation model. `https://www.minimax.io/`.

Kaleb Newman, Tyler Zhu, and Olga Russakovsky. 2026. Video models reason early: Exploiting plan commitment for maze solving. *arXiv preprint arXiv:2603.30043*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2026. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646.

OpenAI. 2024. Video generation models as world simulators.

OpenAI. 2025. Sora 2 System Card.

OpenAI. 2026. GPT Image 2 (gpt-image-2). `https://developers.openai.com/api/docs/models/gpt-image-2`.

Yu Qi, Xinyi Xu, Ziyu Guo, Siyuan Ma, Renrui Zhang, Xinyan Chen, Ruichuan An, Ruofan Xing, Jiayi Zhang, Haojie Huang, Pheng-Ann Heng, Jonathan Tremblay, and Lawson L.S Wong. 2026. Mme-cof-pro: Evaluating reasoning coherence in video generative models with text and visual hints. *arXiv preprint arXiv:2603.20194*.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, Lei Bai, Wanli Ouyang, and Ruimao Zhang. 2024. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*.

Runway. 2025. Introducing Runway Gen-4.5.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, Mojie Chi, Xuyan Chi, Jian Cong, Qinpeng Cui, Fei Ding, Qide Dong, Yujiao Du, Haojie Duanmu, Junliang Fan, and 152 others. 2026. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*.

Lin Song, Wenbo Li, Guoqing Ma, Wei Tang, Bo Wang, Yuan Zhang, Yijun Yang, Yicheng Xiao, Jianhui Liu, Yanbing Zhang, Guohui Zhang, Wenhu Zhang, Hang Xu, Nan Jiang, Xin Han, Haoze Sun, Maoquan Zhang, Haoyang Huang, and Nan Duan. 2026. Awaking spatial intelligence in unified multimodal understanding and generation. *arXiv preprint arXiv:2605.04128*.

Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, and Xipeng Qiu. 2025. Thinking with video: Video generation as a promising multimodal reasoning paradigm. *arXiv preprint arXiv:2511.04570*.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, and 43 others. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Wan AI. 2026. Wan 2.7.

Letian Wang, Chuhan Zhang, Rishabh Kabra, Jasper Uijlings, Steven Waslander, Andrew Zisserman, Joao Carreira, Kaiming He, Misha Andriluka, Eduard Gabriel Bazavan, Andrei Zanfir, and Cristian Sminchisescu. 2026a. Video generation models are general-purpose vision learners. *Preprint*, arXiv:2607.09024.

Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, Jiahui Ge, Qianli Ma, Hang He, Yifan Zhou, Lingzi Guo, Lantao Mei, Jiachen Li, Hanwen Xing, Tianqi Zhao, and 36 others. 2026b. A very big video reasoning suite. *arXiv preprint arXiv:2602.20159*.

Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, Hokin Deng, Dahua Lin, Ziwei Liu, and Lei Yang. 2026c. Demystifing video reasoning. *arXiv preprint arXiv:2603.16870*.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. 2025. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. 2025. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, and 62 others. 2025a. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, and 20 others. 2025b. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*.

Keming Wu, Yijing Cui, Wenhan Xue, Qijie Wang, Xuan Luo, Zhiyuan Feng, Zuhao Yang, Sudong Wang, Sicong Jiang, Haowei Zhu, Zihan Wang, Ping Nie, Wenhu Chen, and Bin Wang. 2026a. Worldreasonbench: Human-aligned stress testing of video generators as future world-state predictors. *Preprint*, arXiv:2605.10434.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. 2026b. Kris-bench: Benchmarking next-level intelligent image editing models. *Advances in Neural Information Processing Systems*, 38.

Cheng Yang, Haiyuan Wan, Yiran Peng, Xin Cheng, Zhaoyang Yu, Jiayi Zhang, Junchi Yu, Xinlei Yu, Xiawu Zheng, Dongzhan Zhou, and Chenglin Wu. 2025a. Reasoning via video: The first evaluation of video models' reasoning abilities through maze-solving tasks. *arXiv preprint arXiv:2511.15065*.

Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yan Huang, Min Xu, and Qiang Wu. 2025b. Unified video editing with temporal reasoner. *arXiv preprint arXiv:2512.07469*.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. 2025c. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pages 83048–83077.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. 2024. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. *Advances in Neural Information Processing Systems*, 37:105345–105374.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, and 17 others. 2026. World action models are zero-shot policies. *Preprint*, arXiv:2602.15922.

Jana Zeller, Thaddäus Wiedemer, Fanfei Li, Thomas Klein, Prasanna Mayilvahanan, Matthias Bethge, Felix Wichmann, Ryan Cotterell, and Wieland Brendel. 2026. Mentisoculi: Revealing the limits of reasoning with mental imagery. *arXiv preprint arXiv:2602.02465*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2026a. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:56729–56762.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, Hua Yang, Xue Yang, and Haodong Duan. 2026b. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *Advances in Neural Information Processing Systems*, 38.

Shenghe Zheng, Junpeng Jiang, and Wenbo Li. 2026. V-bridge: Bridging video generative priors to versatile few-shot image restoration. *arXiv preprint arXiv:2603.13089*.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Yejin Choi, James Zou, Chaorui Deng, Shen Yan, Haoqi Fan, Cihang Xie, Huaxiu Yao, and Qinghao Ye. 2025. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. *Preprint*, arXiv:2511.02779.

Tinghui Zhu, Sheng Zhang, James Y. Huang, Selena Song, Xiaofei Wen, Yuankai Li, Hoifung Poon, and Muhao Chen. 2026. Video models can reason with verifiable rewards. *Preprint*, arXiv:2605.15458.

Appendix Contents

A Benchmark Construction Details **15**
- A.1 Task Material Collection 15
- A.2 Evaluation Criteria 16
- A.3 Benchmark Augmentation 17
- A.4 More Details 18

B Evaluation Details **20**
- B.1 Model Generation Configuration . 20
- B.2 Full Results of Evaluation 20
- B.3 Human Ceiling Evaluation 21
- B.4 Stability Analysis under Full-Set Evaluation 21
- B.5 Evaluation Results Gallery 22

C VLM-as-Judge Details **25**
- C.1 Implementation 25
- C.2 Adaptation for Image Outputs . . 25
- C.3 Human Correlation 26

D Details of Discussion and Analysis **31**
- D.1 Failure Modes: More Examples . 31
- D.2 Input Condition Sensitivity 32
- D.3 Scaling Fine-tuning on Synthetic Data 33
- D.4 Reasoning along Denoising Trajectory 35

E Other Details **36**
- E.1 Detailed Comparison with Related Works 36

F Submission Checklist **37**
- F.1 Potential Risks 37
- F.2 Licenses and Terms of Use 37
- F.3 Artifact Use Consistent With Intended Use 37
- F.4 Personally Identifying Information and Offensive Content 37
- F.5 Human Annotation Recruitment, Payment, and Data Consent 37
- F.6 Ethics Review 37
- F.7 Information About Use Of AI Assistants 37
- F.8 Package Usage and Parameters . . 37

A Benchmark Construction Details

A.1 Task Material Collection

Complementing Section 3.2, we further provide more details about the task material collection.

Input Image Construction. Our input images come from two sources. The first source consists of web images or existing visual datasets, selected and adapted when they naturally match the task setting. The second source consists of images generated by image generation models such as GPT-Image-2 (OpenAI, 2026) and Nano Banana Pro (Google, 2025c). For generated inputs, we use two construction pipelines. Some images are generated directly from textual scene descriptions when the target scene can be clearly specified in language. For tasks requiring precise spatial layouts or structured configurations, we first create an intermediate sketch or schematic using scripts or rendering engines, and then use it as a visual reference for photorealistic-style image generation. In both cases, we aim to produce inputs with natural lighting, plausible object appearance, realistic backgrounds, and stable spatial layouts.

Since generated images are not always faithful to the intended task design, we use a human-in-the-loop refinement process. Each generated image is manually checked for object and attribute correctness, spatial layout, and visual artifacts. Unqualified images are regenerated or edited through multiple iterations, until the input image supports the intended visual procedure. All input images are standardized to a 16:9 aspect ratio for compatibility across video generation models.

Input Image Construction Prompt: MAZE

Convert this line-art schematic of a square grid maze into a realistic indoor photo, viewed top-down at ∼75°.
- **Walls**: where the input shows thin black line segments, render low white wooden-plank walls (painted or light natural wood, a single material); keep each wall segment's position and length, but make the walls clearly low so the corridors stay visible from above.
- **Floor**: one piece of light short-pile carpet (solid or faint pattern, e.g. off-white, light grey, or pale khaki) covering the whole maze; its pattern must not blur the wall boundaries.
- **Robot car**: the blue circle in the input marks the car's starting cell. Replace it with a small realistic engineering-style robot car (wheels, visible sensors / camera, metal or plastic shell), placed in the same cell and facing the adjacent open corridor; its length must be under 2/3 of the corridor width and it must not touch any wall.
- **Goal**: the highlighted pink-red filled square is the unique exit. Place a distinct rectangular red mat on that cell (clearly separate from the carpet, vivid colour); no other coloured markers anywhere.
- **Background**: a simple indoor room (faint furniture edges, baseboards), no outdoor elements; edges and corners may faintly show a lived-in bedroom corner (bed sheet, nightstand, lamp), soft and

Input Image Construction Prompt: MAZE *(continued)*

slightly blurred, never a plain white or grey studio backdrop.
- Keep the viewpoint and the maze geometry (every cell, every wall segment, the exit gap) strictly unchanged; no artifacts, no distortion.

Input Image Construction Prompt: RECOVER_2D_NET

Convert this polygon-net schematic into a realistic ∼25°–30° top-down photo so each tile's thickness is visible.
- **Tiles**: each polygon is an independent semi-transparent matte Magna-Tiles-style plate (∼3–4 mm thick); per-tile shape, colour, and count match the input 1-to-1.
- **Rigid pieces**: visible plastic rim, silver magnetic strips along the sides, faint seams between adjacent tiles; never a single printed sheet.
- **Hatched face**: keep the 45° diagonal lines as a silk-screened pattern on that tile's top face.
- **Lift / shadow**: tiles sit slightly above the table and cast shape-aligned shadows.
- **Table**: light wood, soft directional lighting; edges may faintly show a desk corner.
- No hands, robot arms, clamps or external objects; every tile stays in its input position with magnetic edges snapped.

Input Image Construction Prompt: OBJECT_PACKING

A casual phone-style real-scene photo, no intermediate schematic.
- **Container**: in the centre of a light wooden table, place an open beige canvas tote bag (35 cm tall) with its top fully open so the empty interior cavity is clearly visible. The interior is plainly roomy: large enough to fit every sensibly-sized item with margin to spare.
- **Surrounding items**: arrange 4 items on the tabletop around the container, evenly spaced, none overlapping or occluding another: a small white folded parasol, an orange-capped sunscreen bottle, a rolled blue-and-white striped beach towel, and a clear glass of iced lemonade.
- **Lighting**: natural cool-white window light from the upper left; every item casts a crisp, real-world drop shadow and shows side highlights, with the contrast and sharpness of a casual snapshot: **not** a soft studio-lit look.
- **Camera**: 70° top-down at 16:9 aspect ratio.
- No people, hands, text, labels, stickers, logos, arrows, numbers, tools, or extra props anywhere in the frame.

Text Prompt Construction. Each prompt is structured to describe both the task objective and the constraints under which the video should be generated. It first states the goal of the task, then specifies task-specific rules over relevant objects, visual attributes, allowed actions, prohibited actions, and required state changes. Following (Wiedemer et al., 2025), we also include generation-control instructions on the scene background, spatial layout, camera motion, etc. These constraints are intended to reduce irrelevant variation and keep the generated video focused on the intended visual procedure.

Below we show the video prompts used for three representative tasks: MAZE, RECOVER_2D_NET, and OBJECT_PACKING.

Video Prompt: MAZE

Task Goal:
Create a smooth video of the toy car driving forward through the open corridors of the maze, turning at junctions, and stopping on the red goal square.
Constraints:
- The car must not climb, jump, fly over, or clip through any wall; it stays on the corridor floor at all times.
- Use the input layout exactly: no walls, corridors, start cell, or goal are altered.
- Motion is continuous along a single valid corridor path from start to goal; no teleport and no sudden cut.
- Keep the walls, floor, lighting, and viewing angle fixed throughout; no glitches or artifacts.

Video Prompt: RECOVER_2D_NET

Task Goal:
The translucent plastic tiles in the input complete the assembly of a 3D solid entirely on their own: no hands, tools, or external props appear. Each remaining flat face rotates upward about its hinge edge until the net closes into a single closed 3D solid resting on the hatched bottom face.
Constraints:
- **Polygon set preserved**: same count, shapes, and colours as the input; no face is added, removed, replaced, or recoloured.
- **Rigid faces**: each face keeps its exact polygonal shape (a triangle stays the same triangle, a 2:1 rectangle stays 2:1); no stretching, no edge-length or angle change.
- **Adjacency**: two polygons sharing a creased edge in the input share that same edge in the 3D solid; adjacent faces remain joined along their hinge edge throughout; no two faces overlap in 3D, no face is missing on the surface.
- **Bottom face anchored**: the hatched face stays in contact with the table throughout: it does not lift, rotate, or change shape; faces already in their 3D position at frame 0 stay there.
- **Continuous fold**: faces rotate gradually and continuously, no sudden cut from flat to solid, and stay perfectly rigid throughout.
- **Scene fidelity**: no hands, fingers, arms, sleeves, or other objects enter the frame; background, lighting, and table stay consistent with the input; the camera may tilt from top-down up to a $3/4$ view as the solid forms.

Video Prompt: OBJECT_PACKING

Task Goal:
Pack every item that is BOTH size-appropriate AND semantically appropriate for the open container in the centre of the frame into that container; items that do not fit, would spill, contaminate, or damage the container or its contents must stay still on the table. By the end of the video every appropriate item is inside the container and every inappropriate item is still in its original position.
Constraints:
- **Sole container**: the open container in the centre of the frame is the ONLY container: any box, carton, bag, packaging, plate, or other container-like object among the scattered candidate items is itself a candidate item to be judged, never the container.
- **Capacity is not the constraint**: the container is generously sized; rejection comes only from per-item size mismatch or semantic appropriateness, not from running out of room.
- **Container stays put**: the container itself stays in its starting position and orientation, with its top opening visibly upward so packed items can be seen inside; only its top is open.
- **Continuous physical motion**: one continuous take that ends with the correct items deposited inside the container interior; no jump cuts, teleporting, morphing, fades, or glitches.
- **Scene fidelity**: background, table surface, lighting, and camera viewpoint stay fixed throughout; no labels, text, arrows, debugging overlays, or captions appear.

A.2 Evaluation Criteria

Completeness (Comp.) Completeness is the *global* metric of our two-metric design. As defined in Section 3.4, it captures how far the model carried the task goal across the whole clip: a task-specific tiered standard maps each run to one of three levels: `<complete>`, `<partial>`, or `<failed>`. The VLM-based judge, shown a small set of uniformly sampled frames together with that standard, returns the corresponding tier, which is then mapped to $\{0, 0.5, 1\}$. The tiered standard is short by design, so the judge only has to discriminate between the three tiers; the per-task tier definitions used in our experiments are listed in the boxes below.

Completeness Standard: MAZE

0: The car doesn't move, or it immediately clips through / climbs / flies over walls, or it heads the wrong way.
1: The car drives correctly through part of the maze (staying in corridors) but does not reach the red goal square.
2: The car reaches and stops on the red goal square, staying on the corridor floor throughout (a tiny clip is tolerable if it clearly solves the maze).

Completeness Standard: RECOVER 2D NET

0: Stays flat / faces don't fold, or faces are added / removed, or it folds into a wrong shape.
1: Some faces rotate up about their hinges but the solid is not closed / only partially assembled.
2: The net closes into the complete 3D solid resting on its bottom face. One face not perfectly seated is fine if the solid is essentially closed.

Completeness Standard: OBJECT PACKING

0: No item is put into the container at all (packing not attempted), or the scene deviates drastically from the input.
1: Items are actually placed into the container (procedure carried out) but the selection is wrong: appropriate items missed and / or inappropriate items packed.
2: Essentially all size- and semantically-appropriate items end up inside the container and the inappropriate ones stay on the table (at most one mistake tolerated).

Rubric Score (Rub.) Rubric Score is the *local* metric of our two-metric design. As defined in Section 3.4, it scores a generated video against a fine-grained per-task checklist of process and final-state constraints. The judge runs an adaptive coarse-to-fine pass over the clip (sample at 4 fps, refine flagged intervals up to 8 fps) with 10-frame sliding windows so each call stays focused; per rubric item we convert the violation count x across all windows to an item-level score $1/(x+1)$ and average over items. Unlike Completeness, this gives credit for partial process adherence and penalises localised constraint violations even when the global outcome looks correct. The per-task checklists used in our experiments are listed in the boxes below.

Evaluation Rubric: MAZE

- The maze layout: every wall, corridor, the start cell, and the red goal square: is preserved exactly as in the input image; no walls are added, removed, moved, recoloured, or distorted at any point.
- Throughout the video the toy car stays entirely inside the corridors; it never clips through, drives over, climbs on top of, or flies above a wall. A violation requires the car to visibly pass THROUGH a wall to reach the other side (a wall-crossing event); merely touching, grazing, or briefly overlapping a wall outline without crossing to the other side does NOT count as a violation.
- The car's motion is continuous along a single valid corridor path: no teleportation, no sudden jumps, no extra duplicate cars spawning.
- The car visibly turns at junctions to navigate the maze (rather than going straight through walls or taking an impossible shortcut).
- By the end of the video the car has come to a full stop fully on the red goal square, having actually travelled there from its starting cell during the clip.

Evaluation Rubric: RECOVER 2D NET

- **Overall task**: starting from the input (a flat 2D net or a partially-folded shape), the remaining flat faces hinge upward about their fold edges on their own until the net closes into a single closed 3D solid resting on the hatched bottom face. The completed solid is exactly: 2 regular triangle bases + 3 rectangles (right regular 3-gonal prism). No other solid or unrelated object besides the target solid appears anywhere in the scene.
- **Bottom face anchored**: the hatched face stays flat on the table throughout and is the only face in contact with the table; every other face has rotated upward so the solid is closed.
- **Face inventory and identity preserved**: same number of faces with their original shapes, sizes, proportions, and pastel colours: no face added, removed, recoloured, stretched, or scaled.
- **Continuous physical fold**: the unfolding-to-3D plays out as a continuous physical fold: faces hinge upward gradually, stay rigid,

Evaluation Rubric: RECOVER 2D NET *(continued)*

and stay joined at their fold edges throughout; no sudden teleport from flat to solid.

Evaluation Rubric: OBJECT PACKING

- All starting elements from the input remain present and visually identifiable throughout the output: the beige canvas cloth tote bag (about 30 cm tall, top open, interior visible) and every candidate item (open white parasol / umbrella; glass of lemonade (full, open); bottle of sunscreen; rolled-up striped beach towel).
- By the final frame the container holds exactly the items that SHOULD be packed: bottle of sunscreen, rolled-up striped beach towel.
- The items that should NOT be packed remain on the table in their original positions and orientations and are never circled, never moved, never tilted, never picked up: open white parasol / umbrella (reason: an opened parasol is far too large to fit in the bag); glass of lemonade (full, open) (reason: open liquid would spill in a soft bag).
- Only the candidate items end up moved; the container itself stays in its starting position and orientation, with its top opening visibly oriented upward so packed items can be seen inside.
- Motion is one continuous physical take that ends with the correct items deposited inside the container interior. No jump cuts, teleporting, morphing, fades, or glitches.
- Background, table surface, lighting, and camera viewpoint are unchanged from the input throughout the output. No labels, text, arrows, debugging overlays, or captions.
- Total count of items inside the container at the final frame equals the number of correct items (2); total count of items still on the table equals the number of wrong items (2).

A.3 Benchmark Augmentation

Image Output Subset We adapt a subset (totally 16 tasks) of **VGI-BENCH**tasks into single-image output format to support auxiliary evaluation of image generation and unified multimodal models. A task is adapted only when its goal can be meaningfully represented by a single image, such as a final configuration, a selected object, a completed structure, or a visual annotation. Tasks whose correctness inherently depends on continuous temporal interaction, such as hand-object manipulation or contact-rich physical processes, are not included in this branch.

The adaptation preserves the original task goal but changes the expected output form and is used only as a supplementary evaluation. It allows us to compare static target-state inference with procedural video generation under related goals, and helps separate failures caused by goal-state inference from those caused by maintaining a valid process over time. Figure 8 shows representative input–output examples on three adapted tasks.

	Structured Puzzles	Visual Organization	Spatiotemporal Dynamics	Physical Manipulation	Total
# tasks	8	5	7	7	27

Table 8: Number of tasks per domain. Domains are mutually exclusive; each task belongs to exactly one domain.

	Planning	Physics	Spatial	Attribute Grounding	Affordance	Topology	Temporal
# tasks	12	8	7	6	5	3	3

Table 9: Number of tasks per skill tag. Skill tags are non-exclusive: each task carries one to three tags, so column counts sum to more than the total number of tasks.

A.4 More Details

Table 8 and Table 9 report the number of tasks per domain and per skill tag, respectively. Figure 7 shows the word cloud of task prompts. Frequent tokens reflect the process-sensitive and state-evolution nature of the benchmark (e.g., *every*, *flat*, *fixed*, *stays*, *continuous*).

Figure 7: Wordcloud of Task Prompts

Acknowledgement on Task Inspirations While designing our tasks, we surveyed a wide range of benchmarks and studies on reasoning in generative models. A number of our tasks were informed by the task formats, visual settings, or evaluation perspectives introduced in prior efforts. We gratefully acknowledge the following works for inspiring our task collection:

RISE-Bench (Zhao et al., 2026b), MORSE-500 (Cai et al., 2025b), KRIS-Bench (Wu et al., 2026b), MIRA (Zhou et al., 2025), Study on Veo3's emergent capability (Wiedemer et al., 2025), Video-ThinkBench (Tong et al., 2025), Gen-ViRe (Liu et al., 2025b), TiVi-Bench (Chen et al., 2025), MMGR (Cai et al., 2025a), BabyVision (Chen et al., 2026a), MentisOculi (Zeller et al., 2026), VBVR-Bench (Wang et al., 2026b), MolmoAct2 (Fang et al., 2026), and others (Yang et al., 2025a; Luo et al., 2025b; Wei et al., 2025; Li et al., 2025; He et al., 2025a; Qi et al., 2026; Wang et al., 2026c; Newman et al., 2026; Dai et al., 2026; Li et al., 2026; He et al., 2025b; Wu et al., 2026a).

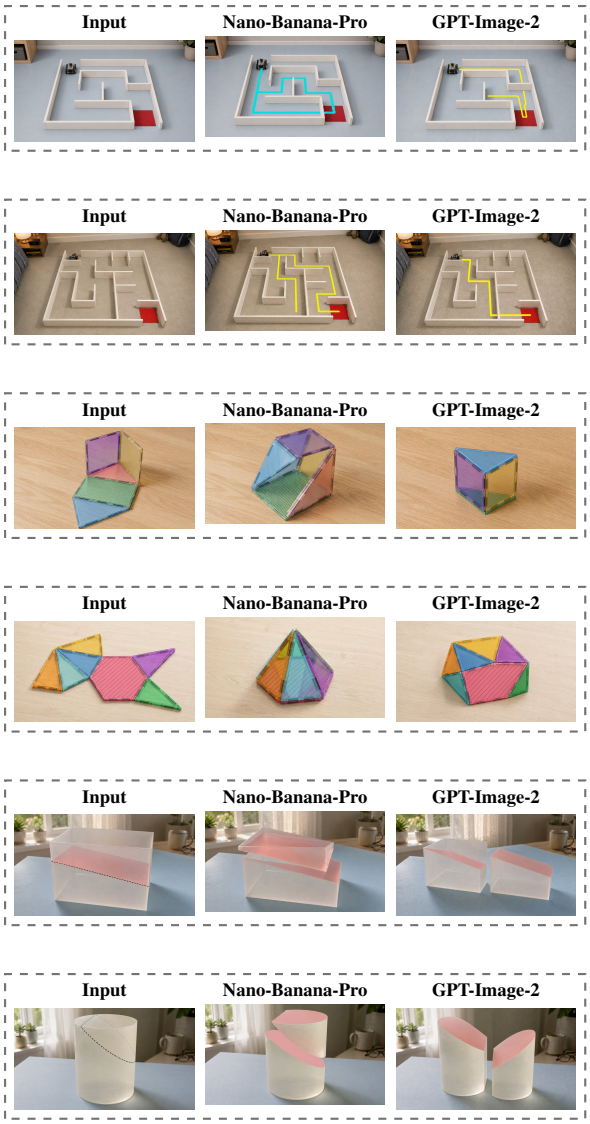


Figure 8: Image-output examples on the adapted single-image subset (top to bottom: MAZE, RECOVER_2D_TO_3D, SECTION_3D_FIGURE). Each row shows the input image, the `Nano-Banana-Pro` output, and the `GPT-Image-2` output.

B Evaluation Details

B.1 Model Generation Configuration

Video Evaluation (Primary). We evaluate our benchmark on a broad set of SOTA video generation models, including:

`Seedance2.0` (Seedance et al., 2026),
`MiniMax-H3` (MiniMax, 2026),
`Sora2` (OpenAI, 2025),
`Veo3.1` (Google, 2025b),
`Kling 3.0` (Kling AI, 2026),
`Wan2.7` (Wan AI, 2026),
`Gen 4.5` (Runway, 2025),
`Wan2.2-I2V-A14B` (Wan et al., 2025),
`HunyuanVideo-1.5` (Wu et al., 2025a).

Due to evaluation cost, for each task we evaluate half of its instances. All tasks follow a unified input format, consisting of a text prompt and an input image. Table 10 shows the specific generation configuration.

Model	Resolution	Duration (s)	FPS	Price (USD / s)
Open Source				
HY1.5	1280×720	5/10	24	–
Wan2.2	1280×720	5/10	16	–
Mnx-H3	960×544	5/10	24	–
Commercial				
Sora2	1280×720	8	30	0.10
Veo3.1	1280×720	8	24	0.20
Sdce2.0	1280×720	5/10	24	0.15
Kling3.0	1280×720	5/10	24	0.08
Wan2.7	1280×720	5/10	30	0.10
Gen4.5	1280×720	5/10	24	0.12

Table 10: Generation configuration of the video models we evaluate, resolution defaults to 1280×720. Price is the per-second generation cost. HY1.5 for `HunyuanVideo-1.5`, Sdce2.0 for `Seedance2.0`, Mnx-H3 for `MiniMax-H3`.

Image Evaluation (Auxiliary). In addition to video generation models, we adapt a subset of benchmark tasks that can be naturally reformulated as single-image output tasks, and evaluate several advanced image generation or unified multi-modal models, including

`Nano Banana Pro` (Google, 2025c),
`Qwen-Image-3-Pro` (Alibaba Cloud, 2026),
`GPT-Image-2` (OpenAI, 2026),
`MAI-Image-2.5-Pro` (Microsoft AI Superintelligence Team, 2026),
`Seedream4.5` (ByteDance Seed, 2026),
`Flux.2-Max Edit` (Black Forest Labs, 2025),
`SenseNova-U1` (Diao et al., 2026),
`JoyAI-Image` (Song et al., 2026),
`Qwen-Image-Edit` (Wu et al., 2025b),
`Step1X-Edit` (Liu et al., 2025a),
`BAGEL` (Deng et al., 2025).

Unless otherwise specified, all image generation models are evaluated at a resolution of 1280×720.

Access and Budget. Commercial models are accessed through their APIs and open-source models are run locally on NVIDIA H20 GPUs.

B.2 Full Results of Evaluation

Per-metric full results Table 11 reports the full per-metric breakdown (Completeness, Checklist Score, and Final) for all video generation models, complementing the aggregated score summary (Table 2) in the main part. The three columns aggregate over instances independently: Comp. and Rub. are the per-cell means of Completeness and Rubric scores, while Final is the per-cell mean of the per-instance product Comp. $\times$ Rub.. Because the mean of products is not equal to the product of means, Comp. $\times$ Rub. $\neq$ Final within a cell in general; this difference reflects the alignment between completeness and rubric performance at the instance level rather than any inconsistency.

Difficulty-level fluctuations. The three difficulty levels are designed to reflect increasing task complexity, but per-domain and per-model scores need not be strictly monotonic. Current video models remain unstable on many tasks, and small changes in layout, object configuration, or required action pattern can interact differently with each model's generative priors. Fine-grained averages may show local inversions, such as a Mid score exceeding an Easy score. We therefore interpret difficulty trends mainly at the aggregate level rather than requiring monotonicity in every model–domain cell.

Strict success rate Table 12 additionally reports the **strict success rate**, where an instance is counted as a success only if the judge marks it as fully completing the task (perfect Completeness) without violating any rules (perfect Rubric Score). The numbers are sharply lower than the aggregated scores in Table 2, reflecting how rare end-to-end success still is for current video models under this strict criterion. We also include the human ceiling performance for reference, demonstrating the gap of visual intelligence between current models and average humans, as detailed in Appendix B.3.

Category	Level	*Commercial*																		*Open Source*											
		Sdce2.0			**Sora2**			**Veo3.1**			**Kling3.0**			**Wan2.7**			**Gen4.5**			**Mnx-H3**			**HY1.5**			**Wan2.2**			**VBVR-Wan2.2**		
		Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final	Comp.	Rub.	Final
Visual Organization	Easy	82.0	84.8	72.6	70.0	76.1	55.6	66.0	72.5	50.3	82.0	80.4	67.6	52.0	68.6	37.0	76.0	77.5	60.5	56.0	73.4	42.8	36.0	66.7	26.6	46.0	60.2	30.4	58.0	73.6	46.1
	Mid	70.0	77.3	56.4	58.0	62.7	37.0	66.0	64.4	43.3	62.0	73.2	47.5	44.0	67.1	32.2	54.0	68.1	38.3	56.0	72.6	42.3	28.0	61.0	17.9	24.0	59.9	15.3	46.0	65.7	32.4
	Hard	70.0	74.1	53.5	56.0	66.7	39.1	62.0	69.6	44.0	58.0	72.8	43.7	46.0	66.7	32.5	58.0	70.6	41.9	52.0	74.8	39.8	38.0	62.6	24.7	26.0	64.1	18.5	48.0	68.6	34.3
	Avg.	74.0	78.8	60.8	61.3	68.5	43.9	64.7	68.8	45.9	67.3	75.4	52.9	47.3	67.5	33.9	62.7	72.0	46.9	54.7	73.6	41.6	34.0	63.4	23.1	32.0	61.4	21.4	50.7	69.3	37.6
Spatiotemporal Dynamics	Easy	81.1	67.3	54.6	62.9	63.9	45.3	46.2	58.7	28.1	62.5	70.2	45.2	57.1	64.3	38.9	56.2	66.9	39.6	52.9	76.0	44.3	50.0	64.7	34.7	60.0	62.1	38.7	57.1	66.6	37.0
	Mid	68.6	66.6	47.4	52.9	60.6	34.6	37.2	54.7	21.2	56.4	62.2	35.2	61.4	66.0	42.6	54.3	60.0	34.6	51.4	74.3	40.7	41.4	61.7	27.2	52.9	56.9	32.0	30.0	71.0	21.6
	Hard	50.0	62.3	33.4	41.4	57.3	24.4	28.6	52.4	16.9	47.2	58.2	28.2	42.9	61.1	28.9	50.0	54.6	29.5	47.1	71.8	35.9	38.2	58.2	23.9	31.4	55.5	19.9	40.0	68.2	28.2
	Avg.	66.8	65.4	45.3	52.4	60.6	34.8	37.7	55.4	22.3	55.7	63.7	36.5	53.8	63.8	36.8	53.6	60.8	34.8	50.5	74.0	40.3	43.3	61.5	28.7	48.1	58.2	30.2	42.4	68.6	29.0
Structured Puzzles	Easy	64.3	67.1	46.9	51.4	72.1	40.5	41.4	64.3	31.8	62.9	68.4	45.3	30.0	66.8	25.6	41.4	59.6	29.2	58.6	80.5	51.9	14.3	64.5	9.9	22.9	61.2	17.1	68.6	82.0	57.9
	Mid	70.0	63.8	45.1	34.3	63.8	25.8	32.9	59.7	21.8	61.4	61.7	38.3	31.4	60.9	23.3	35.7	51.7	20.7	60.0	81.8	52.3	11.4	67.4	8.8	12.9	55.7	8.4	54.3	77.0	45.0
	Hard	60.0	63.6	41.8	34.3	60.5	22.4	22.9	53.4	13.5	48.6	57.1	28.9	34.3	61.0	24.7	34.3	54.5	18.7	58.6	80.2	47.6	10.0	63.4	7.9	10.0	51.9	5.6	60.0	79.6	50.0
	Avg.	64.8	64.8	44.6	40.0	65.5	29.5	32.4	59.1	22.4	57.6	62.4	37.5	31.9	62.9	24.5	37.1	55.3	22.9	59.0	80.8	50.6	11.9	65.1	8.9	15.2	56.3	10.4	61.0	79.5	51.0
Physical Manipulation	Easy	78.6	76.8	64.4	61.4	66.8	47.9	62.9	67.2	49.6	78.6	71.9	60.6	71.4	70.6	55.8	67.1	69.3	52.1	72.9	76.7	58.6	35.7	58.5	22.5	44.3	66.2	33.7	55.7	78.6	46.7
	Mid	81.4	69.8	59.2	60.0	62.7	41.8	57.4	59.2	40.4	68.6	68.2	51.2	75.7	61.7	48.2	67.1	58.9	43.4	58.6	67.3	41.5	28.6	57.0	16.7	32.9	64.8	22.1	57.1	72.9	43.1
	Hard	72.9	58.0	44.4	51.4	58.9	32.6	55.7	58.8	37.3	72.9	58.9	45.6	60.0	59.4	37.1	61.4	58.5	39.5	52.9	60.5	33.0	21.4	57.1	11.9	30.0	60.8	17.5	48.6	72.1	37.9
	Avg.	77.6	68.2	56.0	57.6	62.8	40.8	58.7	61.8	42.5	73.3	66.3	52.5	69.0	63.9	47.1	65.2	62.3	45.0	61.4	68.2	44.4	28.6	57.5	17.0	35.7	63.9	24.4	53.8	74.6	42.6
Overall		70.5	68.6	51.0	52.2	64.0	36.7	46.9	60.6	32.0	63.0	66.3	44.0	50.8	64.3	35.7	54.1	61.9	36.6	56.5	74.2	44.4	29.0	61.8	19.1	32.8	59.8	21.6	52.1	73.3	40.2

Table 11: Full per-metric evaluation results of video generation models across our reasoning categories at Easy/Mid/Hard difficulty plus a per-category average, scored by three metrics: Completeness (Comp.), Rubric Score (Rub.), and the final aggregated score (Final). The main-paper Table 2 reports the final aggregated score only. Higher is better; the best and second-best Final scores in each row are highlighted. HY1.5 for `HunyuanVideo-1.5`, Sdce2.0 for `Seedance2.0`, Mnx-H3 for `MiniMax-H3`. Note that, within each cell, generally, Comp. $\times$ Rub. $\neq$ Final: this is because Comp. and Rub. are the mean Completeness and Rubric scores aggregated independently across instances, whereas Final is the mean of the per-instance product Comp. $\times$ Rub. (i.e., the mean of products, not the product of means).

B.3 Human Ceiling Evaluation

Setup. To estimate a human performance ceiling, we conduct a human study on our benchmark. Since the tasks are designed to rely on everyday visual and physical intuition, the study aims to verify task solvability and clarity rather than measure human response speed. For each task, pilot instances are randomly sampled across different difficulty levels. We first run a pilot study and compute the average response time for each task family; the response-time limit is then set to the 90th percentile of these averaged pilot response times. This yields a time budget that covers most normal human solutions while avoiding excessive delays.

For each task and each difficulty level, we randomly sample half of the instances to form the evaluation set. Accuracy is averaged across participants and instances. Due to the heterogeneous nature of the tasks, we allow different response formats. For example, some tasks naturally require spatial annotation, such as maze solving and Euler path tracing, so participants can annotate the original image, while for tasks that are more naturally solved through verbal reasoning, participants describe their solution orally to the interviewer.

Metrics of Human Response. Video model outputs can exhibit a wide range of unintended behaviors, such as altering the task structure, changing the background, or producing physically inconsistent transitions. We therefore use fine-grained rubrics to evaluate generated videos. For human participants, however, these video-specific checklist violations almost never occur. As a result, checklist-based scoring is less meaningful for human ceiling evaluation. We instead report the strict success rate, where each task instance is judged with a binary label indicating whether the participant successfully solves the task or substantially achieves the intended objective. All participants are undergraduate or graduate students with a computer science background, and receive brief instructions and task-specific examples before the study. The human ceiling evaluation results are in Table 12, together with the strict success rate of video models.

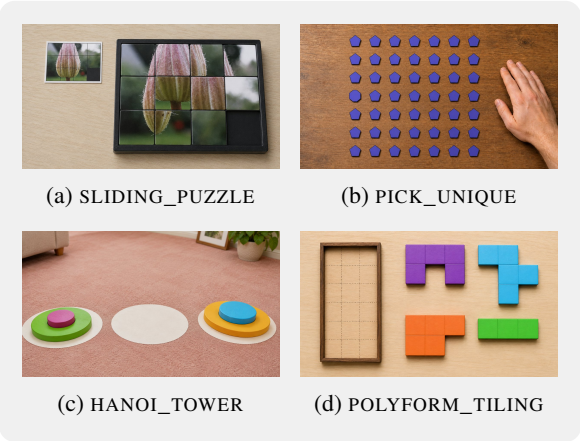
Failure Examples. The ceiling is not perfect: even at the highest difficulty of each task, a non-trivial fraction of participants fail within the response time budget. Fig. 9 collects representative failure cases of several tasks. the failures include often miscounted moves (HANOI TOWER, SLIDING PUZZLE), missed identification of the unique cell among many distractors (PICK UNIQUE), or geometric mis-fits in tiling (POLYFORM TILING).

B.4 Stability Analysis under Full-Set Evaluation

The main results in Section 4.2 are computed on a fixed subset of **VGI-BENCH**: for each (model, task) cell, we generate and evaluate videos for **half of the instances at each difficulty level**, keeping

Category	Level	Human	*Commercial*						*Open Source*			
			Sdce2.0	**Sora2**	**Veo3.1**	**Kling3.0**	**Wan2.7**	**Gen4.5**	**Mnx-H3**	**HY1.5**	**Wan2.2**	**VBVR-Wan2.2**
Visual Organization	Easy	100.0	40.0	16.0	4.0	24.0	8.0	8.0	12.0	4.0	4.0	16.0
	Mid	97.5	16.0	4.0	4.0	12.0	4.0	8.0	8.0	0.0	0.0	4.0
	Hard	92.5	12.0	0.0	4.0	12.0	4.0	8.0	8.0	0.0	4.0	4.0
	Avg.	96.7	22.7	6.7	4.0	16.0	5.3	8.0	9.3	1.3	2.7	8.0
Spatiotemporal Dynamics	Easy	100.0	5.4	5.7	0.0	10.0	5.7	5.0	8.6	0.0	2.9	5.7
	Mid	98.0	5.7	8.6	0.0	2.6	11.4	0.0	8.6	5.7	2.9	8.6
	Hard	92.0	5.7	2.9	2.9	2.8	0.0	0.0	5.7	0.0	2.9	2.9
	Avg.	96.7	5.6	5.7	0.9	5.2	5.7	1.8	7.6	1.9	2.9	5.7
Structured Puzzles	Easy	100.0	11.4	11.4	11.4	8.6	11.4	0.0	22.9	2.9	2.9	20.0
	Mid	100.0	11.4	2.9	0.0	2.9	11.4	0.0	20.0	0.0	0.0	14.3
	Hard	84.4	11.4	2.9	0.0	2.9	5.7	0.0	11.4	0.0	0.0	17.1
	Avg.	94.8	11.4	5.7	3.8	4.8	9.5	0.0	18.1	1.0	1.0	17.1
Physical Manipulation	Easy	100.0	40.0	14.3	20.0	28.6	20.0	22.9	28.6	0.0	8.6	17.1
	Mid	100.0	11.4	2.9	14.7	8.6	2.9	8.6	5.7	0.0	0.0	11.4
	Hard	98.3	5.7	2.9	2.9	5.7	8.6	5.7	2.9	0.0	0.0	8.6
	Avg.	99.4	19.0	6.7	12.5	14.3	10.5	12.4	12.4	0.0	2.9	12.4
Overall		97.1	**14.0**	6.2	5.3	9.5	7.9	5.3	12.1	1.0	2.3	11.0

Table 12: **Strict success rate** (%) of video generation models on **VGI-BENCH**. Layout mirrors Table 2. Per row, the best and second-best scores among the video models are highlighted (Human reference excluded); tied bests share the darker shade. Cells marked "-" have no judged instances yet. HY1.5 for `HunyuanVideo-1.5`, Sdce2.0 for `Seedance2.0`, Mnx-H3 for `MiniMax-H3`.



(a) SLIDING_PUZZLE (b) PICK_UNIQUE (c) HANOI_TOWER (d) POLYFORM_TILING

Figure 9: Representative human failure cases, most cases are sourced from the highest difficulty level of the task.

the video-generation cost manageable. To verify that this half-subset is a reliable proxy for the full pool, we additionally run a full-instance evaluation on a representative slice: 5 video models and 6 tasks spanning the four domains. For this slice, we generate videos for all the instances per level and evaluate the complete set with the same judge protocol. We summarize the half-subset scores with the corresponding full-set scores in Table 13. Across the 5 models, the half-subset score tracks the full-set score within 5 percentage points on average and the model ranking is preserved, suggesting that the main paper's half-subset evaluation is a faithful proxy for the full pool.

Video Model	Full (N=10)	Half (N=5, main)	Δ (Half−Full)
`Kling3.0`	34.8	33.2	−1.6
`Sora2`	30.9	27.2	−3.7
`Veo3.1`	27.8	25.1	−2.7
`Wan2.2`	14.3	14.0	−0.3
`HY1.5`	14.7	11.7	−3.0
Mean	**24.5**	**22.2**	−2.3

Table 13: Stability of half-subset evaluation against full-instance evaluation. Scores are Final Score (rubric × completeness) averaged per task then over the 6 tasks.

B.5 Evaluation Results Gallery

For each task we show three result videos from different video models, ordered low to high by final score. Each row stitches six uniformly sampled frames from that model's generation; the caption bar reports the model name and its final score. Figures 10–15 present one such gallery per task.

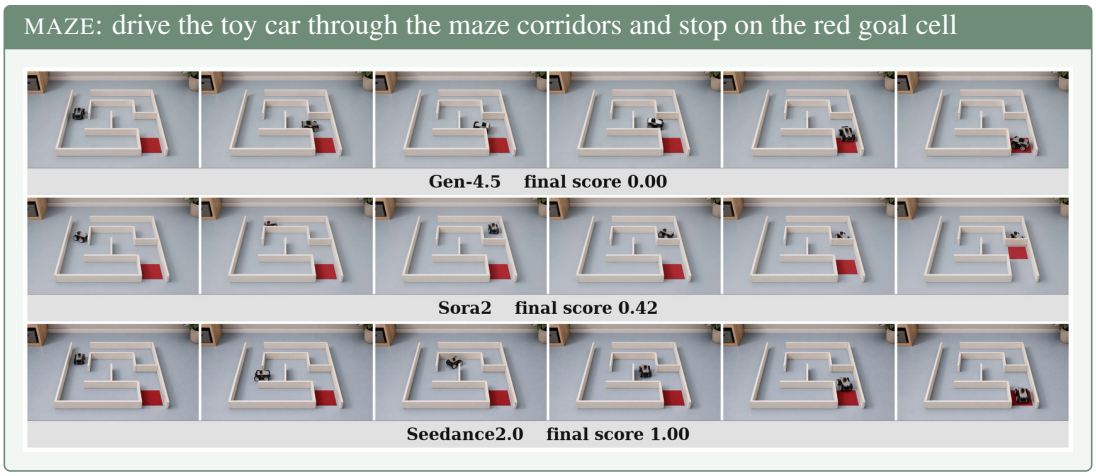


Figure 10: Generated result videos for the MAZE task, from three video models ordered low to high by final score.

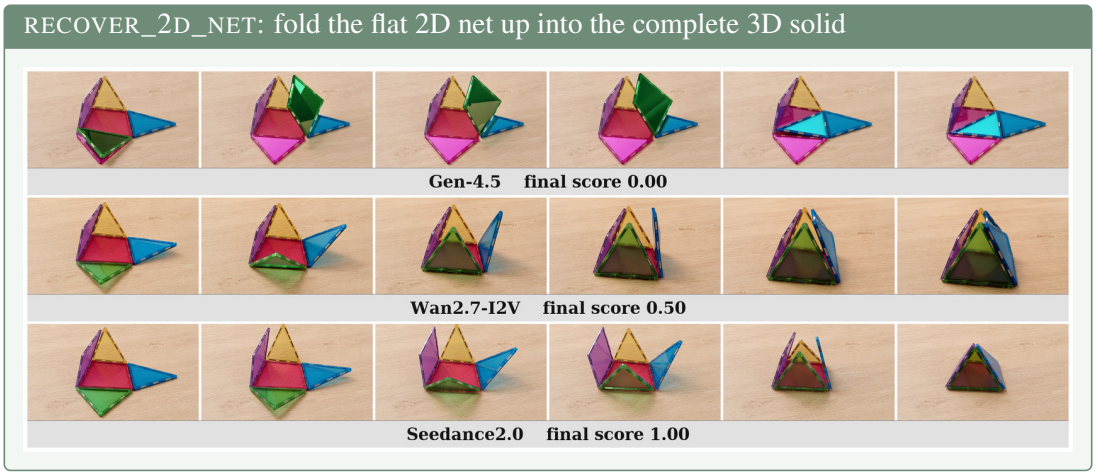


Figure 11: Generated result videos for the RECOVER_2D_NET task, from three video models ordered low to high by final score.

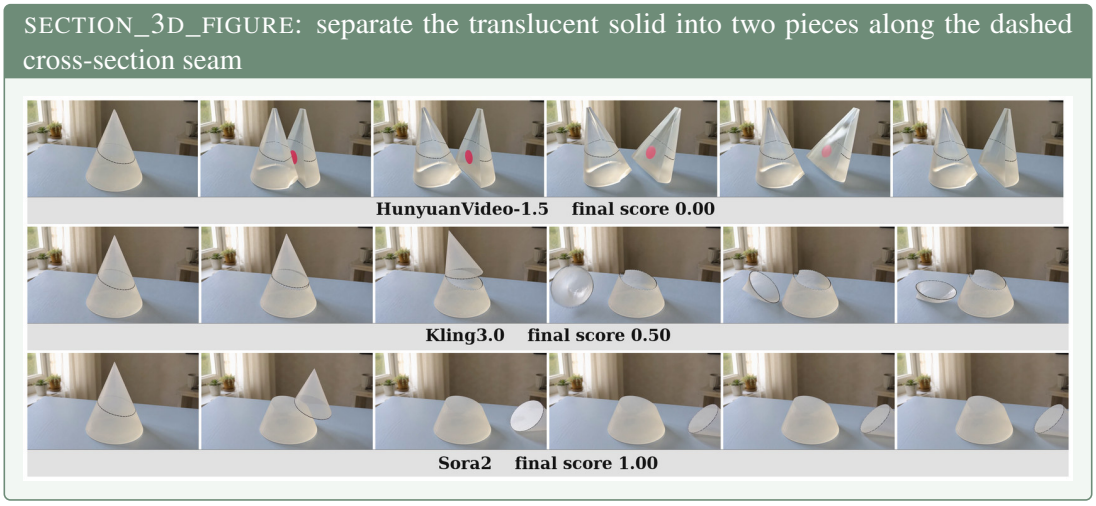


Figure 12: Generated result videos for the SECTION_3D_FIGURE task, from three video models ordered low to high by final score.

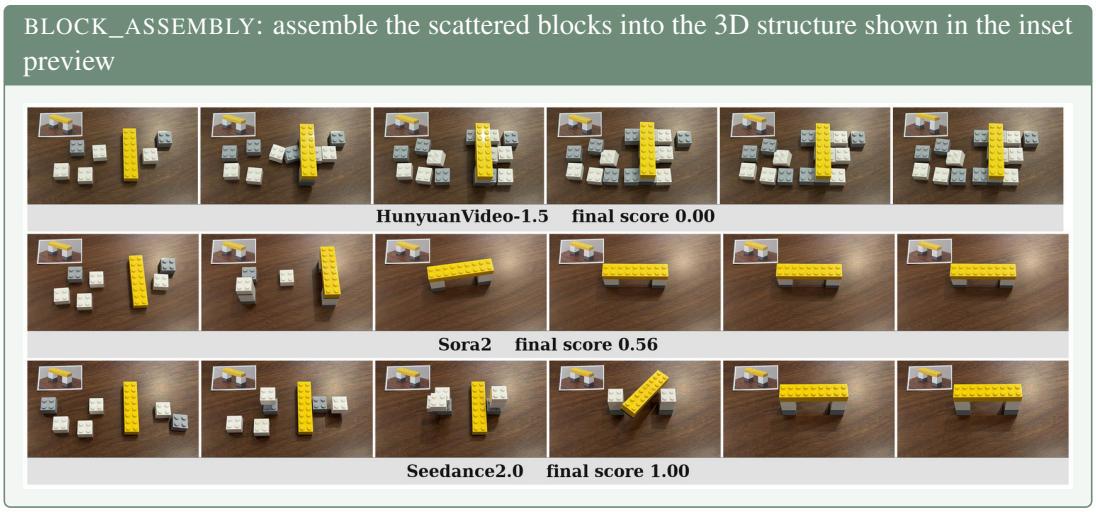


Figure 13: Generated result videos for the BLOCK_ASSEMBLY task, from three video models ordered low to high by final score.

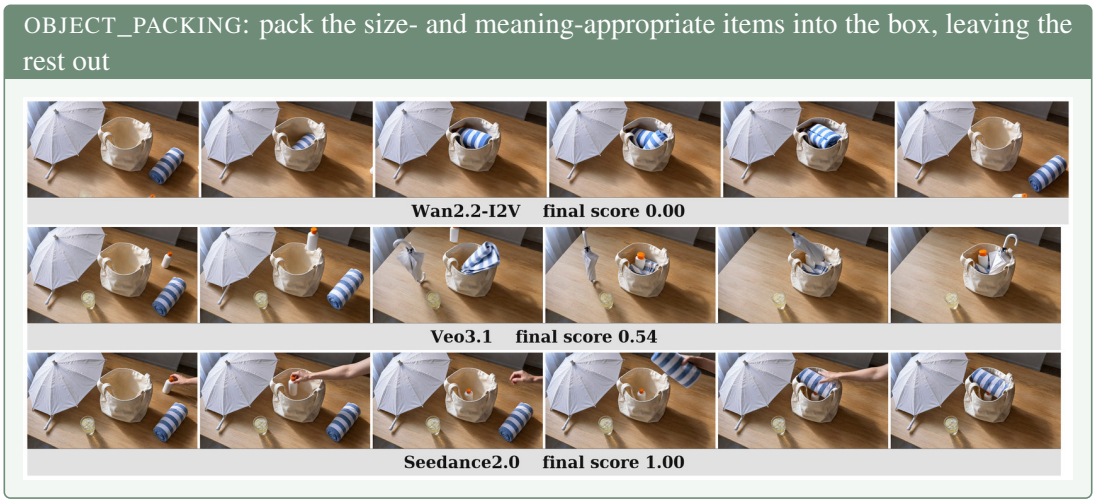


Figure 14: Generated result videos for the OBJECT_PACKING task, from three video models ordered low to high by final score.

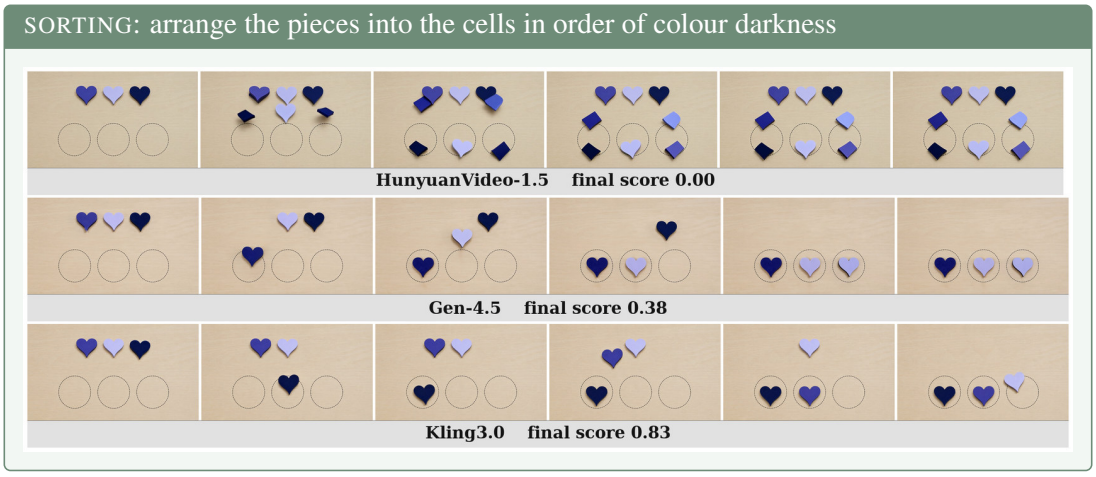


Figure 15: Generated result videos for the SORTING task, from three video models ordered low to high by final score.

C VLM-as-Judge Details

C.1 Implementation

Every generated video is scored by two complementary metrics: a global **Completeness** and a local **Rubric Score**, their product becomes the per-instance **Final Score** of Section 3.4. Both passes share the same underlying VLM (`gemini-3-flash-preview`) but differ in what they see and what they return; the per-pass prompts are in Tables 17 and 16, with `[BRACKETS]` placeholders filled per instance; the paragraphs below expand each stage.

Reference routing. Which ground-truth signal is fed to the judge is task-dependent: some tasks supply a reference image, some a textual description, some both, and a few neither (judged on the checklist alone). Each task is routed to its available signal(s), and `[BRACKETS]` for an absent signal are simply omitted from the prompt.

Completeness. The Completeness pass evaluates the global progress of a generated video toward the task goal. The full judge prompt example is provided in Table 17.

We uniformly sample the full clip at a low frame rate (2fps) and provide the sampled frames, the input image, optional ground-truth references, and the task-specific tiered standard to the VLM judge. The judge assigns one of three tiers: `complete`, `partial`, or `failed`, which are mapped to scores of 1, 0.5, and 0, respectively. A `complete` judgment allows minor visual imperfections as long as the intended procedure is substantially completed; `partial` indicates meaningful progress without reaching the correct final state; and `failed` covers cases with little progress, wrong task execution, or severe drift from the input. This metric captures the macro-level trajectory of the video, while finer-grained process violations are handled by the Rubric Score below.

Rubric pass. The Rubric pass evaluates fine-grained process validity using the task-specific checklist. The full judge prompt example is provided in Table 16.

We use an adaptive coarse-to-fine procedure. The judge first inspects the full video at a coarse sampling rate (*low fps*, 4 fps) with a sliding focus window of 10 frames and a 1-frame overlap between consecutive windows, so that each VLM call only covers a short local segment rather than the entire frame sequence. While checking the rubric items, the judge also nominates suspicious frames that require closer inspection. The time intervals around these frames are then resampled at a finer rate (*high fps*, 8 fps) and re-evaluated with the same checklist. This design allows the evaluator to capture transient violations without densely sampling the whole video.

After all inspected segments are judged, we aggregate the segment-level comments into a violation count for each rubric item. If item i is violated x_i times, its item-level score is defined as $s_i = 1/(x_i + 1)$. This inverse decay strongly penalizes repeated violations while giving diminishing marginal penalty when the same violation has already occurred many times. The final Rubric Score is the arithmetic mean over all checklist items: $\mathrm{Rub} = \frac{1}{N}\sum_{i=1}^{N} s_i$.

The per-item violation counts x_i are produced by a final, image-free *polish* call. This call sees no frames: it is given the full checklist together with all per-window evaluation comments from both the coarse and the refined passes, grouped by frame range and tagged with their time stamps. It deduplicates this evidence—a persistent violation spanning a window counts as a single event, and the same incident reported by two overlapping windows is merged—and returns the final x_i for every checklist item (plus a short overall reasoning). Only this polish call assigns the violation counts; the per-window calls only produce localized comments and nominate frames to zoom into. The polish prompt is given in Table 18.

Aggregation to Final Score and Reported Numbers. As discussed in Section 3.4, for each instance, we combine the two video-evaluation passes multiplicatively: $\mathrm{Final} = \mathrm{Comp.} \times \mathrm{Rub.}$.

Reported video results are averaged over all judged instances; a domain score pools every instance of the tasks belonging to that domain and averages over them, as shown in Tables 2 and Table 11.

C.2 Adaptation for Image Outputs

Image-output models produce a single still image, so we use a lightweight image judge. The judge receives the original input image, a reference image, the generated image, the task specification, and the textual ground-truth description when available. It then evaluates two criteria: **task correctness**, i.e., whether the task-relevant content satisfies the

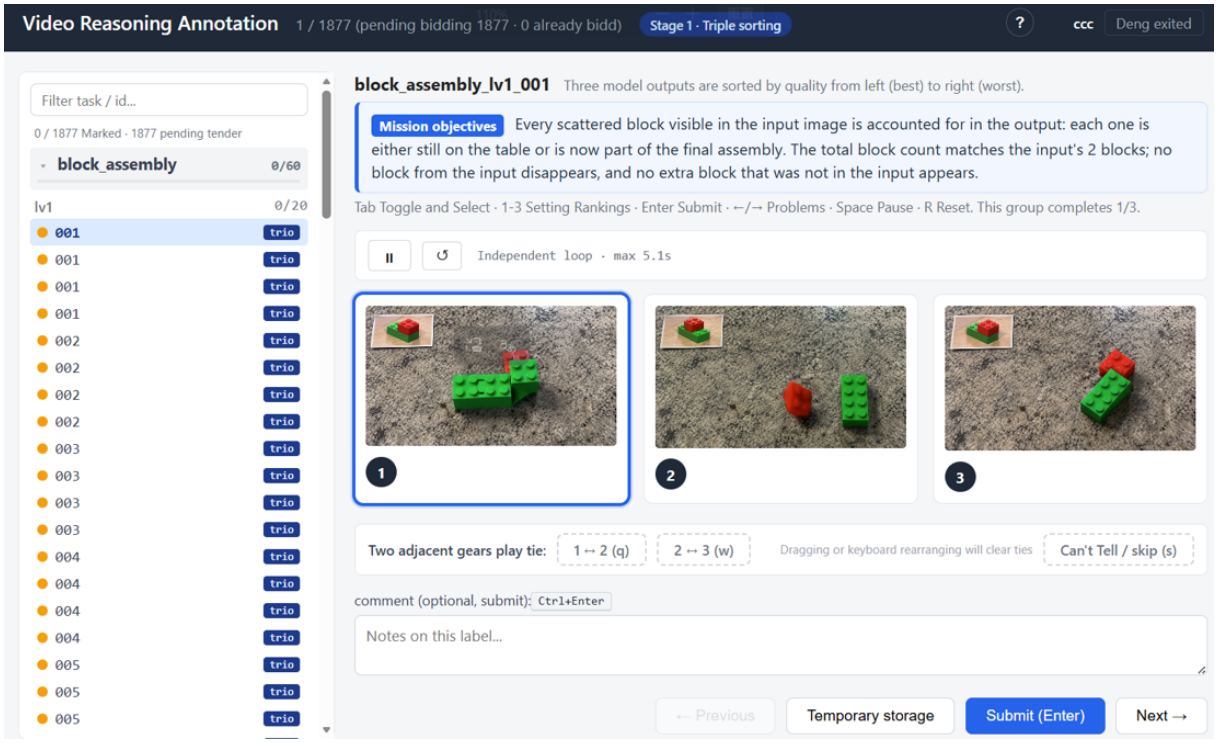


Figure 16: Screenshot of our human-annotation interface for resulting video preference. For each instance, the annotator watches the task instructions alongside three model-generated videos and either ranks them by reasoning progress from best to worst, or flags the triple as a tie / skip when no clear ordering exists.

specification, and **background preservation**, i.e., whether non-task regions remain roughly consistent with the input image. We allow minor background drift, color or layout changes, and style variation. A generated image is marked as failed when it contains task-critical errors, such as missing, wrong, unreadable, or structurally broken content. The judge returns a binary success signal. Since image outputs have no temporal axis, we do not apply Completeness or Rubric Score; the binary verdict serves as the final per-instance metric.

C.3 Human Correlation

To validate the reliability of our VLM-based evaluator, we collect human preference annotations over generated videos. For each selected task instance, we construct triplets consisting of three response videos from different models. Annotators are asked to rank the three videos by considering task progress, rule violations, and overall correctness. The interface (shown in Figure 16) also allows annotators to mark a tie between two videos, or skip a triplet when the three responses are indistinguishable; the skip rate is small in practice. We have four annotators and ensure that each triplet is annotated by two annotators.

We convert the triplet rankings into pairwise preferences and apply two filtering steps before computing human correlation. First, we remove pairs where the two annotators give contradictory preferences. Second, we remove pairs that would introduce preference cycles, such as $A > B > C > D > A$. After filtering and preprocessing, we obtain 1100+ human preference pairs, which serve as the human reference set to verify our evaluator.

Judge Design	AUC	Pairwise Acc.
Main (w/ `Gemini-3-Flash`)	**0.803**	**73.2%**
w/o adaptive fps	0.772	69.5%
w/o focus window	0.753	68.3%
w/ `GPT-5-mini`	0.690	64.3%
w/ `Claude-Haiku-4.5`	0.478	47.9%
w/ `Qwen3.6-Plus`	0.624	59.1%

Table 14: Reliability of our VLM-based evaluator compared with human annotations.

We report two correlation metrics. The first is pairwise agreement: given a human preference pair, we use the evaluator scores of the two videos to predict which one is preferred. Since the evaluator score is normalized to $[0, 1]$, we treat score differences smaller than 0.05 as ties. The second metric is ROC-AUC over strict human preference pairs. For each pair, we compute the score difference be-

tween the two videos and measure whether this continuous signal ranks human-preferred videos higher. Unlike pairwise agreement, AUC does not depend on a tie threshold and is therefore less sensitive to score calibration.

Breakdown by domain and skill tag. Beyond the aggregate numbers, we further decompose the human-correlation pool along the two levels of our taxonomy (Section 3.1), by routing every human preference pair to the domain and the skill tag(s) of its underlying task; since skill tags are non-exclusive, a task annotated with k tags contributes its pairs to k skill pools. The results are reported in Table 15.

Agreement is stable across the taxonomy: every domain and every skill tag stays within roughly 0.73–0.85 AUC and 69%–80% pairwise accuracy, so no single slice of the benchmark is driving the aggregate reliability. At the domain level, **Structured Puzzles** shows the strongest agreement with human preferences, which we attribute to its explicit rules, constrained solution space, and correspondingly verifiable outcomes. Agreement is lowest for **Physical Manipulation**, where success hinges on subtle details of real-world physical interaction that are more easily missed by a frame-sampling evaluator. The skill-level view echoes the same pattern: **Topology** and **Physics** are the weakest-aligned tags. Topology requires tracking fine-grained changes in connectivity and containment, while Physics depends on fine-grained interactions and dynamics spread across frames; both are harder to read off sampled frames than the discrete, checkable state changes that dominate the better-aligned tags such as **Affordance**.

Metric	Structured Puzzles	Visual Organization	Spatiotemporal Dynamics	Physical Manipulation
AUC	**0.851**	0.800	0.814	0.783
Pairwise Acc.	**75.0%**	69.2%	74.1%	70.1%

Metric	**Planning**	**Spatial**	**Temporal**	**Attribute Grounding**
AUC	0.815	0.802	0.819	0.800
Pairwise Acc.	73.0%	71.0%	75.6%	73.2%
Metric	**Physics**	**Affordance**	**Topology**	
AUC	0.799	**0.842**	0.731	
Pairwise Acc.	69.2%	**80.3%**	69.6%	

Table 15: Judge–human agreement broken down by domain (top) and skill tag (bottom), for the main judge configuration of Table 14. Skill tags are non-exclusive, so a task with k tags contributes its pairs to k skill pools.

Prompt: Rubric pass (one call per batch in the adaptive / sliding-window loop)

[SYSTEM]

You are a visual rubric judge for video generation outputs. You are judging a VIDEO (delivered as a small batch of frames sampled at a given fps), NOT a single image. Each frame is preceded by a metadata line giving the time in seconds and the original-video global frame index.

Rules:

- Reference frames by their global_frame number when describing what you see.
- EVERY frame in this batch must appear in at least one evaluation's frame_indices. If a frame is uneventful, group it with the nearest informative comment instead of skipping.
- Flag substantive rubric violations and ignore minor noise. Substantive means the kind of failure the rubric is actually targeting (for a maze task: the car clearly clipping through or driving over a wall, a teleport or a duplicate car, a missing required turn, etc.). IGNORE minor near-rubric deviations that are NOT the kind of failure the rubric is checking for: sub-pixel jitter, a small-part sliver of overlap during fast motion, faint compression artifacts, lighting flicker, and so on.
- DO NOT hallucinate events between adjacent sampled frames. At the current fps, consecutive frames in this batch are separated by $\sim 1/\text{fps}$ seconds of native video that you cannot see. If you SUSPECT something happened in that gap (two objects collided / phased, a piece teleported, the car briefly crossed a wall, etc.) but you cannot see it actually, you MUST list both flanking global_frame numbers in "frames_to_inspect_closely", so the next round samples higher fps in that gap. Do NOT write comments like "they collide between frame X and frame Y" or "the object phases through between X and Y" based on inference, that is hallucination. Only flag a collision / phase / teleport as a violation when you can point to one or more sampled frames that visibly show the overlap, the impossible state, or the object in two places at once.

Each turn you are also shown the original input image and the reference ground-truth image (when available) BEFORE the sampled frames; use them as references for what the input scene contains and what one acceptable output looks like.

Always reply with valid JSON matching the schema in the user message.

[USER]

Task: [TASK NAME]

Task main goal: [RUBRIC MAIN GOAL]

Detailed checklist (each item is scored independently; reference these 1-based indices when you flag violations):
1. [CHECKLIST ITEM 1]
2. [CHECKLIST ITEM 2]
...
K. [CHECKLIST ITEM K]

Ground-truth description (success / final state to reach): # Optional
"""
[GT DESCRIPTION]
"""

Below: the original input image then the reference ground-truth image (if available), followed by THIS BATCH's sampled video frames.

Input Image: ⟨INPUT IMAGE⟩

Ground Truth Image: ⟨GT IMAGE⟩

Sampled Frames:
⟨FRAME 1⟩ t=t_1 global_frame=g_1
...
⟨FRAME n⟩ t=t_n global_frame=g_n

Output STRICT JSON (no markdown, no prose outside JSON):
```
{
   "evaluations": [
     {"frame_indices": [⟨global_frame⟩, ...], "comment": "⟨one or two sentences⟩"}
   ],
   "frames_to_inspect_closely": [⟨global_frame⟩, ...] // 0..M entries; can be []
}
```

Table 16: Prompt used for the **Rubric Score** judgment. Per-batch `frames_to_inspect_closely` drives the adaptive refinement loop; the localized comments it returns are later turned into violation counts by the polish call (Table 18).

Prompt: Completeness pass (single call per video)

[SYSTEM]

You are a visual judge for task completeness in a generated VIDEO. You are shown the INPUT image (the video's first frame / starting state), optionally a GT image and/or a GT description (one acceptable final state / goal), and the whole clip sampled as still frames in time order. Using ONLY the supplied tiered standard, decide how far the task's GOAL was actually carried out and return a single integer tier. Judge the procedure and the final state you can see across the frames; do not invent events between frames. Reply ONLY with the JSON schema given in the user message.

[USER]

Task: [TASK NAME]
You will be shown: the INPUT image (starting state at t=0); a GT image (one acceptable final state); then the sampled video frames in time order.

Tiered standard (score by how far the goal was carried out):
"""
[TIERED COMPLETENESS STANDARD]
"""

Ground-truth description (the goal / acceptable final state):
"""
[GT DESCRIPTION]
"""

Input Image (Start Frame): ⟨INPUT IMAGE⟩
Ground Truth Image # Optional ⟨GT IMAGE⟩
Sampled frames from the video: ⟨FRAME 1⟩ t=t_1 global_frame=g_1
⟨FRAME 2⟩ t=t_2 global_frame=g_2
...
⟨FRAME N⟩ t=t_N global_frame=g_N

Reply ONLY with valid JSON:
{"tier": 0|1|2, "reasoning": "1-2 sentences"}
tier is the single integer from the tiered standard above (0, 1, or 2).

Table 17: Prompt used for the **Completeness** judgment. The tier is mapped to $\{0, 0.5, 1\}$ and becomes the first factor of *Final Score*.

Prompt: Polish pass (single image-free call per video)

[SYSTEM]

You are a careful video-judging assistant. You are finalising the per-rubric-item violation counts for a VIDEO based on sampled-frame evidence collected by per-batch judges.

[USER]

You are finalising the rubric violation counts for a VIDEO. The per-batch evidence below was collected from multiple LLM calls, each shown a contiguous window of sampled frames (round 1 at low fps over the whole clip; round 2 at high fps zoomed into the frames round 1 flagged). Your job is to deduplicate that evidence into a final violation count per rubric checklist item.

Task main goal:
[RUBRIC MAIN GOAL]

Detailed checklist (reference these 1-based indices):
1. [CHECKLIST ITEM 1] ... K. [CHECKLIST ITEM K]

Per-batch evidence (each batch saw a contiguous window at the given fps; consecutive batches share their boundary frame as overlap):
- round r fps=f batch b (global g_{lo}-g_{hi}, t ...s):
  comments:
    - global_frames ⟨a-b⟩: [COMMENT]
  ... (repeated for every batch of every round)

Tolerance: only count something as a violation if it is defensible to a human reviewer as a real failure; ignore minor near-rubric noise (sub-pixel jitter, a one-pixel sliver during fast motion, lighting flicker, micro-drift). A borderline / cosmetic deviation does NOT count.

Output STRICT JSON only:
```
{
  "by_range": [{"global_frames": "⟨a-b⟩", "summary": "⟨one sentence⟩"}, ...],
  "checklist": [
    {"index": ⟨1..K⟩, "n_violations": ⟨int ≥ 0⟩, "evidence": "⟨one sentence citing time ranges⟩"}
    // EXACTLY K entries, indexes 1..K in order
  ],
  "overall_reasoning": "⟨two or three sentences⟩"
}
```

Per-item rule: n_violations is the FINAL deduplicated count of DISTINCT substantive violations of that item across the whole video, after integrating evidence from ALL batches (round 1 + round 2). Deduplication: (a) a persistent violation spanning a continuous window counts as ONE; (b) two reports in adjacent batches within 0.5 s across the boundary count as ONE; (c) the same incident in two overlapping batches is ONE. Per-item score $= 1/(\text{n_violations} + 1)$; rubric_score is the arithmetic mean over items.

Table 18: Prompt used for the polishing and summarization of Rubric Score per-batch judge results.

Prompt: Image-output judge (single call per generated image)

[SYSTEM]

You are a visual judge for image-generation outputs. The model produces a SINGLE generated image; judge only that still image.

You will be shown, in order: the original INPUT image, optionally a reference GROUND-TRUTH image showing one acceptable answer, and the MODEL's generated image. A textual ground-truth description may also appear in the prompt.

Success requires BOTH criteria:
(1) **Task correctness** — the task-relevant content satisfies the spec. The GT image or JSON answer gives one acceptable solution; semantically equivalent alternatives also count.
(2) **Background preservation** — outside the task-relevant region, the generated image should remain consistent with the INPUT: same scene, objects, layout, framing, and major colors. Substantial changes such as adding/removing objects, recoloring or distorting the background, rebuilding the scene, or changing style/medium count as failures. Background is judged against INPUT, not GT.

OVERALL RULE: focus on key task-critical content. Mark success=true if the key content is correct despite minor issues, such as small background drift, slight non-task repositioning, minor color/lighting changes, small rendering artifacts, or light style shifts. Fail only if the key content is wrong, missing, unreadable, or structurally broken, or if the background is substantively rebuilt/replaced/restyled. If no INPUT is provided, judge only task correctness.

Reply ONLY with valid JSON in the schema specified in the user message.

[USER]

Task: [TASK NAME]

Task spec (what the model was asked to generate):
"""
[IMAGE PROMPT]
"""

You will see N image(s), in this order:
1. INPUT (the image the model was given)
2. GROUND-TRUTH image (one acceptable answer)
3. MODEL's GENERATED image (to be judged)

Ground-truth description (success / final state to reach):
[GT DESCRIPTION]

Input Image: ⟨INPUT IMAGE⟩
Ground Truth Image: ⟨GT IMAGE⟩
Generated Image: ⟨GENERATED IMAGE⟩

Decide if the generated image substantively satisfies the task spec. Both must hold for success: (1) task correctness and (2) background preservation (skip if no INPUT). Apply the key-thing rule: ignore minor secondary issues; only fail on substantive task-critical errors or wholesale background rebuild.

Reply ONLY with valid JSON matching this schema:
{"success": true|false, "reasoning": "2-3 sentences explaining the verdict"}

Table 19: Prompt used for the **Image-output judge**. Mirrors the video-judge two-criteria structure (task correctness ∧ background preservation), reduced to a single still-image call: the model output is one image, so no adaptive frame sampling or sliding-window loop is needed.

D Details of Discussion and Analysis

D.1 Failure Modes: More Examples

Failure Mode: Physical Collapse

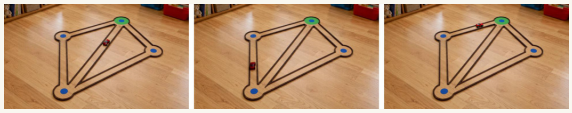
The laptop is tilted up at a noticeable angle, yet **the cup lying on its lid does not roll down**, appearing **glued to the surface** and violating gravity.

The maze layout abruptly changes mid-video: a **previously solid wall opens up into a gap**.

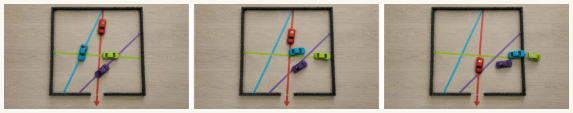
The towel **passes straight through the bottle**; in the intermediate frame the towel and the bottle are visibly **entangled rather than colliding**.

When the laser hits the mirror, **the incident ray is sometimes missing** and **the reflected ray is sometimes missing**, a clear violation of optical physics.

Failure Mode: Rule Violation



An Eulerian path requires traversing every edge exactly once, but the trajectory in the video **terminates without covering all edges**.



The LEAVE_PARKING_LOT task imposes two rules: (i) each car may only move along the track segment of its own color, and (ii) no car may cross over the walls. **The video violates both**.

The Tower of Hanoi allows only one disk to be moved at a time, yet **the video moves two disks simultaneously**.

Cars navigating the maze are not allowed to jump over the walls, yet **the car in the video does exactly that**.

Failure Mode: Object/State Inconsistency

While the car is navigating through the maze, **it suddenly splits into two cars**.

As the cup is being picked up, **a second identical cup splits off from it**.

The glasses have just been moved to the right side, yet in the very next moment **they reappear on the book in the middle**.

After the acrylic panels finish unfolding, **an extra panel suddenly appears** that was not present before.

Figure 17: Representative failure cases for the three failure modes: **Physical Collapse**, **Rule Violation**, and **Object/State Inconsistency**.

D.2 Input Condition Sensitivity

D.2.1 Oracle Prompting.

Prompt optimization is a common test-time strategy for improving video generation quality and controllability, and recent studies have explored similar directions (Tong et al., 2025; Chen et al., 2025; Cheng et al., 2026; Kim et al., 2026). VideoThinkBench (Tong et al., 2025) studies how prompt rewriting and related prompting strategies affect video-based reasoning, while TiVi-Bench (Chen et al., 2025) introduces VideoTPO as a training-free test-time optimization method for improving reasoning performance. Motivated by these works, we examine the performance ceiling when the intended solution is made explicit in the prompt.

We construct an **oracle prompt** for a subset of 7 tasks by augmenting the canonical task rules with a ground-truth step-by-step solution. Each prompt contains (i) a concrete, executable solution, such as an exact move sequence or target assignment, followed by (ii) the original task rules. This reduces the reasoning burden and tests whether the model can follow detailed instructions and render a valid visual process under all constraints. In the examples below, we show the solution explicitly and abbreviate the retained content as `[original task goal and rules]`.

We focus on open-source models and commercial systems that expose prompt enhancement as a configurable option. Since many proprietary systems may internally rewrite user prompts, isolating the effect of oracle prompting is difficult. We therefore conduct the controlled comparison using **`HunyuanVideo-1.5`**, **`Veo 3.1`**, and **`Wan 2.7`**.

Oracle Prompt: SORTING example 1

Move every top-row piece one at a time, in darkness order (darkest first, lightest last), into a bottom-row cell so the bottom row reads darkest → lightest left to right. Per-instance assignment:
step 1: top-row piece at position 1 (rank-1 darkest) → slot 1
step 2: top-row piece at position 3 (rank-2 darkest) → slot 2
step 3: top-row piece at position 2 (rank-3 darkest) → slot 3

Each move is one continuous human-hand pick-and-place from the top-row position to the target slot. Once placed, a piece does not move again; cells themselves never move. Piece colours, shapes, sizes unchanged. Exactly one bare human hand operates the pieces; no tools or other agents enter the frame. Keep background, lighting, and camera fixed. No glitches.

`[original task rules]`

Oracle Prompt: SORTING example 2

Move every top-row piece one at a time, in darkness order (darkest first, lightest last), into a bottom-row cell so the bottom row reads darkest → lightest left to right. Per-instance assignment:
step 1: top-row piece at position 5 (rank-1 darkest) → slot 1
step 2: top-row piece at position 2 (rank-2 darkest) → slot 2
step 3: top-row piece at position 3 (rank-3 darkest) → slot 3
step 4: top-row piece at position 4 (rank-4 darkest) → slot 4
step 5: top-row piece at position 1 (rank-5 darkest) → slot 5

Each move is one continuous human-hand pick-and-place from the top-row position to the target slot. Once placed, a piece does not move again; cells themselves never move. Piece colours, shapes, sizes unchanged. Exactly one bare human hand operates the pieces; no tools or other agents enter the frame. Keep background, lighting, and camera fixed. No glitches.

`[original task rules]`

Oracle Prompt: MAZE example 1

The blue dot (or toy car) drives smoothly through the 4x4 maze corridors from its starting cell to the centre of the red goal cell, taking exactly 6 single-cell steps. Each step moves to the adjacent corridor cell in the named direction:
down, down, right, right, right, down
Stay strictly inside corridors; never cross, climb, or pass through any wall. Motion is continuous from start to goal: no teleport, no jumps, no airborne motion. Maze layout, walls, dot/car, and goal square are pixel-identical to the input.
Use exactly the input scene; do not add, remove, recolour, or resize any object beyond what the solution above moves. No hands or external agents enter the frame; motion is autonomous. Keep background, lighting, and camera fixed. No glitches or artifacts.

`[original task rules]`

Oracle Prompt: MAZE example 2

The blue dot (or toy car) drives smoothly through the 5x5 maze corridors from its starting cell to the centre of the red goal cell, taking exactly 10 single-cell steps. Each step moves to the adjacent corridor cell in the named direction:
right, down, left, down, down, down, right, right, right, right
Stay strictly inside corridors; never cross, climb, or pass through any wall. Motion is continuous from start to goal: no teleport, no jumps, no airborne motion. Maze layout, walls, dot/car, and goal square are pixel-identical to the input.
Use exactly the input scene; do not add, remove, recolour, or resize any object beyond what the solution above moves. No hands or external agents enter the frame; motion is autonomous. Keep background, lighting, and camera fixed. No glitches or artifacts.

`[original task rules]`

Oracle Prompt: ROPE_UNTANGLE example 1

Two human hands enter from the sides and cooperatively untangle the cables shown in the input. By the final frame every cable lies in its own region as a single roughly-straight line with zero crossings: no cable passes over, under, or through another. Resolve each crossing by lifting one cable up and over the other (visible 3D motion), never by phasing through. Loops shrink and open as cables slide free. Same cable count, same individual colours, same connectors at both ends of every cable as the input; nothing is added, removed, recoloured, cut, spliced, or merged.
Two normal human hands only (five fingers each, natural look, no morphing, visible contact). Keep the floor / table background, light-

Oracle Prompt: ROPE_UNTANGLE example 1 *(continued)*

ing, and top-down perspective fixed. No extra cables or objects. No glitches.

`[original task rules]`

D.2.2 Visual Style.

We evaluate the effect of visual style on 5 video models: **`Sora2`**, **`Veo3.1`**, **`Kling3.0`**, **`Wan2.2`**, **`HunyuanVideo-1.5`**, and 7 selected tasks: CLOCK_RUNNING, HANOI_TOWER, JIGSAW_PUZZLE, LEAVE_PARKING_LOT, MAZE, PICK_UNIQUE, and POLYFORM_TILING. Below We show some generated videos of the same task in photorealistic style and line-art style, including MAZE (Figure 18) and HANOI TOWER (Figure 19).

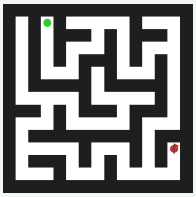
D.3 Scaling Fine-tuning on Synthetic Data

Complementing Section 5.3, we provide further details on how our **VGI-BENCH** tasks are grouped according to their structural overlap with the VBVR training set distribution (Wang et al., 2026b). We identify **overlap** tasks as those whose goals and overall task structures closely match a VBVR training task, with the main difference being the visual domain: VBVR uses abstract synthetic scenes, whereas our benchmark uses realistic inputs. Representative examples include MAZE and CLOCK_RUNNING. **Semi-overlap** tasks share only part of the task structure or require action patterns similar to those seen during VBVR training. **Non-overlap** tasks have no clear structural counterpart in VBVR; this group includes, for example, tasks requiring 3D imagination and understanding, capabilities largely absent from its training distribution. The qualitative examples below use the `VBVR-Wan2.2` / `Wan2.2-I2V` pair.

The following examples illustrate these three groups. For overlap and semi-overlap tasks, each pair places our real-scene input on the left and the most related VBVR training input on the right, each labelled with its own task name; only our input is shown for non-overlap tasks. Each example is annotated with its task-level final score averaged across difficulty levels. The value before the arrow corresponds to the base `Wan2.2-I2V`, the value after it to `VBVR-Wan2.2`, and the highlighted delta indicates a gain in green or a drop in red.

Overlap

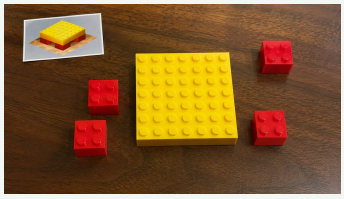
ours: MAZE_SQUARE



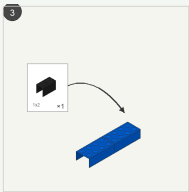
VBVR: `maze`

29.0 (base) → 87.8 (SFT) ↑58.7

Overlap



ours: BLOCK_ASSEMBLY



VBVR: `LEGO_construction_assembly`

7.1 (base) → 51.7 (SFT) ↑44.6

Overlap

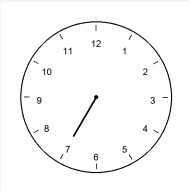
ours: HANOI_TOWER

VBVR: `construction_stack`

3.2 (base) → 52.9 (SFT) ↑49.6

Overlap

ours: CLOCK_RUNNING

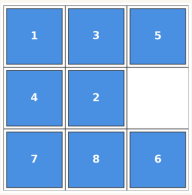


VBVR: `clock`

32.5 (base) → 10.0 (SFT) ↓22.5

Semi-overlap

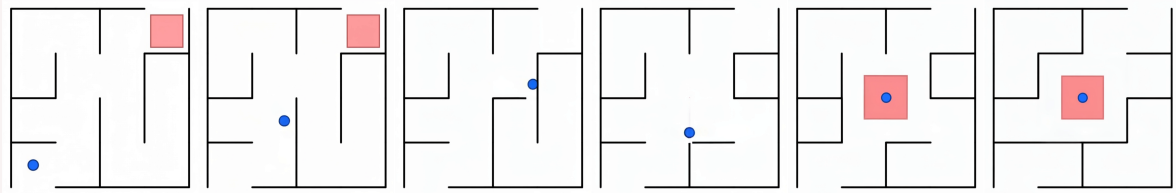
ours: SLIDING_PUZZLE



VBVR: `sliding_puzzle`

0.0 (base) → 7.6 (SFT) ↑7.6

(a) Real-scene: succeeds.



(b) Line-art: fails and the maze structure is not preserved, with prompt explicitly instructing the model to preserve the maze structure and follow the maze rules.

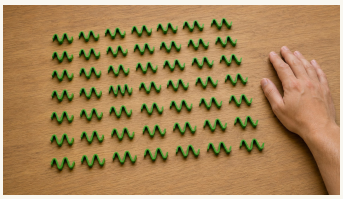
Figure 18: Effect of appearance and style difference. Example task: MAZE.

(a) Real-scene: succeeds.

(b) Line-art: fails and rules are not followed.

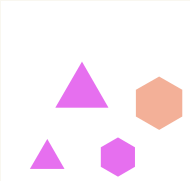
Figure 19: Effect of appearance and style difference. Example task: HANOI TOWER.

Semi-overlap



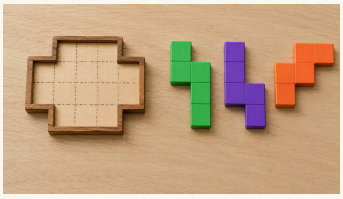
ours: PICK_UNIQUE

53.9 (base) → 55.0 (SFT) ↑1.1



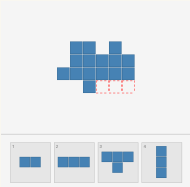
VBVR: spot_unique_non_repeated_color

Semi-overlap



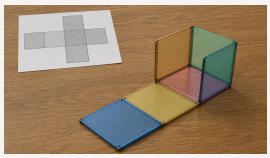
ours: POLYFORM_TILING

15.8 (base) → 49.2 (SFT) ↑33.3



VBVR: construction_blueprint

Non-overlap



EXPAND_3D_TO_2D

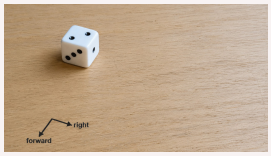
16.3 (base) → 16.7 (SFT) ↑0.4

TOOL_USE_COMMON

58.8 (base) → 44.0 (SFT) ↓14.8

UNTIE_KNOT

1.8 (base) → 34.0 (SFT) ↑32.2



ROLLING_DICE

30.6 (base) → 22.2 (SFT) ↓8.4

At the group level, the average improvement decreases from overlap to semi-overlap and then to non-overlap tasks, as shown in Table 5, consistent with the intuition that training on structurally aligned examples helps models follow task rules, reach the intended goal, and avoid rubric viola-

tions. Smaller gains or occasional degradation are expected when the required structures and skills are weakly represented in the training data.

However, this aggregate trend does not hold uniformly at the task level: some overlap tasks still degrade, while some non-overlap tasks improve. Degradation on overlap tasks may arise from the remaining synthetic-to-real domain gap or execution requirements that are not captured by structural similarity alone. Meanwhile, the improvements on non-overlap tasks often come from more controlled and less aggressive generation after fine-tuning, which reduces large-scale rule violations observed in the base model. Basic spatial and logical capabilities learned during fine-tuning may also transfer to tasks with different surface structures. A representative example here is UNTIE_KNOT: although its success rate changes little, the metric score ($final_score = completeness \times rubric_score$) increases substantially after fine-tuning (from 0.02 to 0.34) since the generated actions become more restrained and controllable.

D.4 Reasoning along Denoising Trajectory

This section gives the protocol behind Table 7 of Section 5.4, which quantifies how often a model revises and potentially corrects its solution states during the denoising process.

Operational definition. Self-correction is only meaningful if we can say what the model's current answer *is* at an intermediate step, so we define it over decoded intermediates rather than over latents. For an open-source model we decode the video at several intermediate denoising steps and compare consecutive decoded checkpoints pairwise. Each earlier→later pair is assigned exactly one label. ♠ **Unrecognizable**: at least one of the two videos is too unresolved to read off a solution state, which is concentrated in the earliest steps. ♣ **Stable**: the solution state is readable and does not change. ■ **Changed**: the solution state changes, sub-divided into (a) correct → wrong, (b) wrong → correct, and (c) wrong → another wrong solution; only (b) counts as self-correction.

Labelling the *transition* rather than the endpoint state is what separates the two failure narratives: a model that never revises and a model that revises into another wrong answer both end up incorrect, but only the latter is evidence of the search-like behavior reported for dLLMs.

Setup. We evaluate four open-source models under their default generation settings (40 denoising steps): `Wan2.2-I2V`, `VBVR-Wan2.2`, `HunyuanVideo-1.5`, and `LTX2.3`. Because early denoising carries the large structural decisions while later steps mainly refine appearance, we sample the trajectory non-uniformly and use the step pairs $1 \to 2$, $2 \to 3$, $3 \to 4$, $4 \to 10$, $10 \to 20$, and $20 \to 40$. We sample 117 task instances and decode their intermediate denoising products, giving $117 \times 6 = 702$ annotated video pairs; Table 7 reports the label distribution within each step pair, so every column sums to 100%.

Interpretation. Two observations follow from the distribution. First, the solution state changes often: outside the unresolved early steps, up to a quarter of all transitions fall into the *changed* categories. Second, these changes are almost never corrections. Wrong → correct transitions stay at or below 0.9% in every step pair and stop occurring after step 4, the first 10% of the trajectory, whereas wrong → wrong$'$ remains common well into the middle of denoising. Since these models solve few instances to begin with, a change of state is much more likely to move between two wrong solutions than to reach the right one. The answer is largely settled within the first few steps, and the remaining steps refine it.

E Other Details

E.1 Detailed Comparison with Related Works

Table 20 (same as Table 1) compares related video benchmarks along three desiderata central to our motivation: 1 **input appearance**, 2 **process-sensitivity**, and 3 **difficulty control**. For benchmarks with data released, human inspection is conducted, while for benchmarks that have not publicly released their data (e.g., TiVi-Bench), we take the statements in paper as the source of truth. The three desiderata are discussed in turn below.

Bench	Reasoning Demand	Appearance Abst., Real.	Process-sensitive No, Yes	Difficulty Control
PhysGenBench	Low/Mid	●	●	✓
WorldSimBench	Low/Mid	●	●	✓
TiVi-Bench†	High	◔	◔	✓
V-ReasonBench	High	◔	◔	✗
VBVR-Bench	High	●	◔	✗
Ours	High	●	●	✓

†Data has not been publicly released, the entries are inferred from the paper.

Table 20: Each pie encodes fraction of tasks that satisfy a desideratum versus do not. For input appearance, Real. denotes photorealistic-style inputs, while Abs. denotes non-photorealistic inputs such as line-art or schematic renderings.

Input Appearance. For each dataset, we count the ratio of input images that depict **realistic scenarios**, as opposed to line-art, schematic, or otherwise abstract styles. PhysGenBench and WorldSimBench are fully realistic; TiVi-Bench contains roughly 1/4 and 1/9 realistic or near-realistic inputs, respectively; VBVR-Bench is entirely script-generated for controllability and scalability, without photographic inputs. Overall, reasoning-heavy video generation benchmarks still largely rely on abstract or line-art inputs, which may deviate from the visual distribution of modern video models. This concern is supported by our style-sensitivity study in Section 5.2: holding the task fixed, switching an instance from a realistic scene to a line-art or low-poly rendering leads to substantially different performance, especially for models with lower overall performance, and abstract variants more often violate task constraints.

Process-sensitivity. We identify a task as process-sensitive if **it requires a non-trivial sequence of state transitions, regardless of whether the final outcome can be verified from a single frame.** For example, maze solving and Hanoi Tower have final states that are easy to inspect, but a valid video must still reach them through legal intermediate steps. This differs from one-shot visual reasoning tasks, such as Raven-style matrices, visual analogy, or even multiple-choice VQA, where the expected answer can be produced without temporal state evolution.

Under this definition, PhysGenBench and WorldModelBench are effectively process-sensitive, as they aim to evaluate physical coherence and world simulation. TiVi-Bench contains 15 process-sensitive tasks out of 24, while the remaining are closer to one-shot reasoning. For V-ReasonBench ($\sim$33%) and VBVR-Bench (47%), the portion of both parts are roughly comparable. In our benchmark, tasks are designed with explicit state transitions, and the rubrics inspect intermediates and transition validity (Section 3), making shortcut-to-final-state behavior detectable as a violation.

Difficulty Control. Task difficulty would become less interpretable when tasks are either saturated by current models or far beyond their feasible regime. For example, long-horizon tasks may require trajectories exceeding the practical generation duration, while knowledge-heavy tasks may rely on advanced scientific, cultural, or historical expertise rather than visually grounded reasoning. Although such tasks can serve as stress tests, diagnostic benchmarks should also produce informative differences among current models. We therefore consider two criteria to control difficulty: ① **feasibility calibration**, where we verifies that tasks can be meaningfully attempted by current video models without failures dominated by excessive duration or non-visual expertise; and ② **multi-level difficulty design**, where tasks are organized into explicit levels to support graded performance analysis as procedural complexity increases.

Under these criteria, PhysGenBench and WorldSimBench are treated as difficulty-calibrated since their tasks primarily target physical world simulation, with prompts manually reviewed for feasibility and clarity. PhysGenBench is marked as "partial" for not providing explicit difficulty splits. In contrast, reasoning-heavy benchmarks like TiVi-Bench, V-ReasonBench, and VBVR-Bench do not include a model-facing feasibility calibration stage; among them, only TiVi-Bench provides multi-level difficulty design. Our benchmark satisfies both criteria through pre-generation and manual review for feasibility calibration, together with explicit multi-level construction. All above are shown in the "Difficulty Ctrl." column of Table 20.

F Submission Checklist

F.1 Potential Risks

The potential risks are minimal. Our benchmark does not involve sensitive domains, personal data, or identity-related attributes. Human studies only collect anonymized task responses and preference annotations. We also screen task descriptions, prompts, and visual contents to avoid offensive or sensitive material.

F.2 Licenses and Terms of Use

We use both commercial video generation APIs and open-source model weights in our experiments. For commercial APIs, we follow the corresponding provider terms of service and use the generated outputs only for research evaluation. For open-source models and tools, we use them under their released licenses and cite the original creators. We do not redistribute third-party model weights or proprietary API outputs beyond what is permitted by their terms.

For the benchmark artifact created in this work, we will release the data and evaluation code under a research-friendly license, with the intended use limited to research evaluation and diagnostic analysis of video generation models.

F.3 Artifact Use Consistent With Intended Use

We use existing artifacts, including video/image generation models, VLMs, evaluation tools, and packages like FFmpeg, only for research evaluation and data processing, following their intended use and applicable terms. We do not use them for commercial training, individual profiling, or real-world decision-making.

Our benchmark is intended for academic research and diagnostic evaluation of procedural and visual reasoning in video generation models. Human annotations are used only for estimating human-ceiling performance and validating VLM-as-Judge, and are anonymized before analysis.

F.4 Personally Identifying Information and Offensive Content

Our benchmark is constructed from task-level metadata and synthetic visual instructions, and does not include personal data or real-world records that identify individuals. For human studies, we collect only task responses for the human-ceiling evaluation and preference annotations for validating the reliability of VLM-as-Judge. We do not collect names, contact information, demographic attributes, or other sensitive personal information. All human annotations are anonymized and reported only in aggregate. We manually inspect task descriptions, prompts, and visual contents to remove offensive, hateful, sexually explicit, or otherwise sensitive content.

F.5 Human Annotation Recruitment, Payment, and Data Consent

The human annotations were provided by the paper co-authors as part of the research process. We did not recruit external participants through crowdsourcing platforms or student pools, and no separate compensation was provided. All annotators were aware that their task responses and preference annotations would be used for benchmark validation, including estimating human-ceiling performance and validating VLM-as-Judge reliability.

F.6 Ethics Review

The human-ceiling evaluation and preference annotations were conducted by the paper co-authors as part of the research process. We did not obtain formal ethics review board approval or exemption. No external participants were recruited, and we did not collect names, contact information, demographic attributes, or other sensitive personal information. The annotations were used only for benchmark validation and reported in aggregate.

F.7 Information About Use Of AI Assistants

We used AI assistants for grammar checking, language polishing, and debugging assistance. The authors reviewed and verified all technical content, experimental design, analyses, and final manuscript decisions.

F.8 Package Usage and Parameters

We use FFmpeg for video preprocessing. Specifically, we extract frames from each generated video at a fixed frame rate using the following command:

```
ffmpeg -i input.mp4 -vf fps=n
        frames/%04d.png
```

This extracts n frames per second (we use $n = 2$, 4, or 8 depending on the judging pass, see Appendix C.1) and saves them as PNG. No additional normalization or filtering is applied.