VBVR-PRO: A SCALABLE AND VERIFIABLE SUITE FOR NATIVE VISUAL REASONING

Anonymous authors
Paper under double-blind review

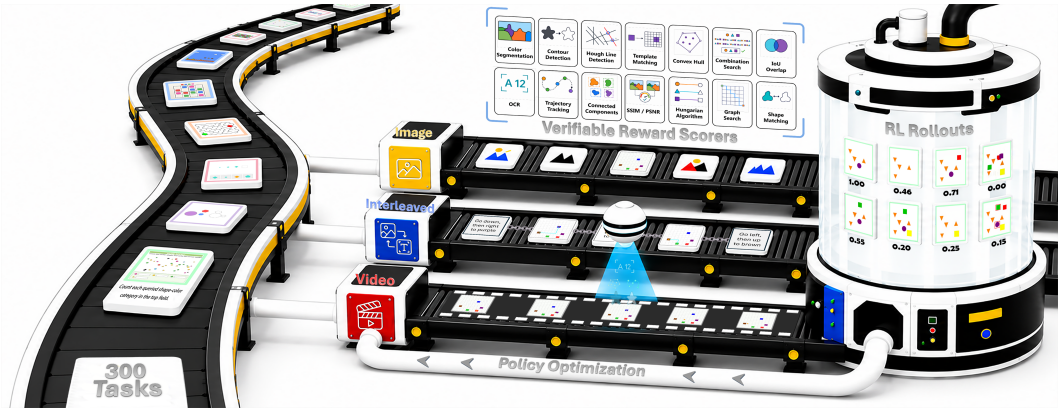

Figure 1: **Overview of VBVR-Pro.** VBVR-Pro provides *300* visual reasoning tasks that support controlled studies of image, interleaved text-image, and video generators under a unified task distribution. Its *verifiable reward scorers* provide reproducible, human-aligned evaluation and further serve as task-grounded reward functions for large-scale multi-task reinforcement learning.

ABSTRACT

Native visual reasoning treats visual states (i.e., images and videos) not as merely inputs to be understood or outputs to be rendered, but the intermediate steps through which a problem is solved. We introduce **VBVR-Pro**, a closed-loop testbed that makes native visual reasoning through generation trainable, verifiable, optimizable, and experimentally controllable. **1) Task scaling.** VBVR-Pro curates a task space of *300* procedurally generated tasks. Models trained on VBVR-Pro show strong transfer beyond the proposed suite across *seven* external video reasoning benchmarks. Analysis suggests that these gains reflect visual reasoning rather than instruction-pattern fitting. **2) Verifiable rewards.** VBVR-Pro provides verifiable reward scorers for task-grounded evaluation. While the prevalent *VLM-as-a-judge* paradigm shows recurring failures, deterministic scorers align more closely with human judgments, and can further serve as reliable reward signals for large-scale multi-task reinforcement learning. **3) Modality study.** VBVR-Pro enables comparisons across leading image, video, and interleaved generators under a shared task distribution and scoring protocol. Our analysis shows that video generation remains strongest for modeling spatiotemporal persistency, while interleaved generation provides a compute-efficient alternative where visual states are crucial. We release all data, models, scorers, and code to facilitate future research.

1 INTRODUCTION

The success of large language models has made language the dominant medium for studying machine reasoning. Yet many forms of intelligence in the physical world are inherently non-linguistic, involving spatial transformations, temporal continuity, object persistence, and dynamic interactions. Recent studies suggest that generation can better expose such reasoning (He et al., 2026b; Wang et al., 2026c), motivating a new paradigm of *native visual reasoning*: models reason by constructing, updating, and optimizing visual states (e.g., images and videos), which serve as *first-class* reasoning substrates rather than merely inputs to a language-based reasoner or final outputs to be

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

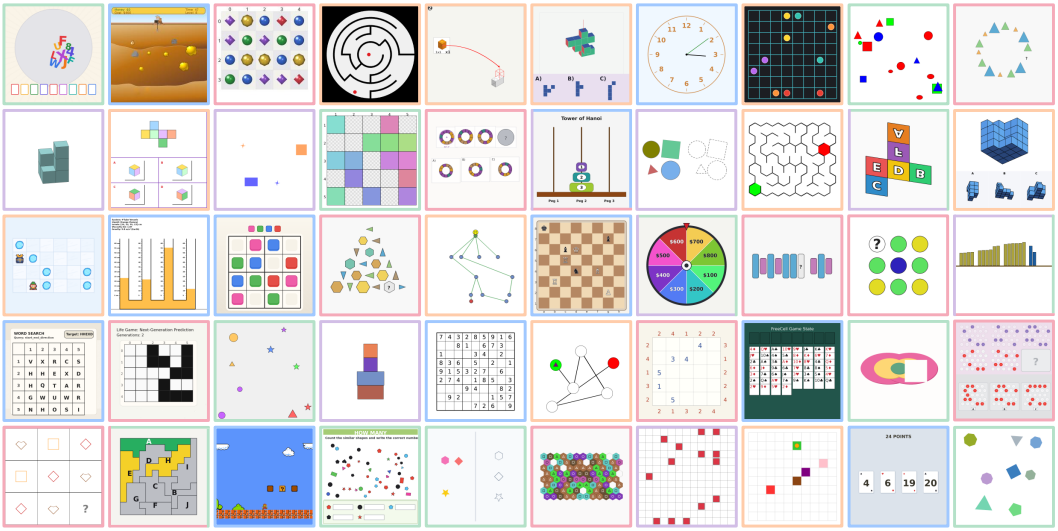

Figure 2: **Overview of VBVR-Pro tasks.** We show representative tasks spanning our five cognitive faculties, outlined according to their underlying capabilities: Perception, Transformation, Spatiality, Abstraction, and Knowledge. Tasks requiring multiple capabilities use multiple outline colors.

generated. However, native visual reasoning remains difficult to study systematically: it remains unclear whether visual reasoning through generation can be trained at scale, reliably evaluated, optimized with reward feedback, and compared across different generative substrates. In this light, we introduce VBVR-Pro, a closed-loop framework with procedurally generated tasks, outputs are verifiably scored, models are optimized with reliable rewards, and modality choices are compared under a shared protocol.

First, native visual reasoning lacks scalable training sources. Existing benchmarks cover diverse capabilities but are primarily designed for evaluation and often provide little training data. Conversely, existing large-scale synthetic sources (Wang et al., 2026a) tend to focus on simplified symbolic settings, leaving open whether they teach transferable visual reasoning skills. VBVR-Pro addresses this by defining a controlled procedural task space. Each task is implemented as a procedural generator. Importantly, multiple aligned modalities, including videos, keyframes, and interleaved textual annotations, are generated simultaneously to enable fairer comparison among image generators, video generators, and interleaved generators trained from the same task distribution. Metadata needed for verifiable evaluation are also recorded. This yields 300 tasks (Figure 2) and models trained on resulting data transfer strongly beyond the proposed suite. In our capability transfer study, video reasoners trained on VBVR-Pro obtain consistent gains (often more than 20 percentage points) across a wide range of held-out visual reasoning benchmarks, including RISE-Video (Liu et al., 2026c), V-ReasonBench (Luo et al., 2025b), RULER-Bench (He et al., 2025), MME-CoF-Pro (Qi et al., 2026), VideoThinkBench (Tong et al., 2026), BabyVision (Chen et al., 2026b), and IntelligentVBench (Pan et al., 2026). These results suggest that covering diverse reasoning task types matters more for transfer than simply scaling instances from a narrow task set.

Second, native visual reasoning requires reliable feedback. While *VLM-as-a-judge* is attractive for open-ended outputs, it can be unreliable for visual reasoning, where correctness depends on exact counts, fine-grained spatial relations, temporal consistency, and rule satisfaction. We systematically analyze leading MLLMs, for example GPT-5.5 (OpenAI, 2026a), Qwen-3.7-plus (Qwen Team, 2026b), and Gemini-3.1-Pro (Google DeepMind, 2026), as judges and identify recurring failure modes, including numerical imprecision, neglect of fine-grained evidence, and misunderstanding of task rules. To address this, VBVR-Pro pairs evaluated tasks with verifiable reward scorers based on structured semantic extraction and task-grounded verification. Unlike prior model-level validation using aggregate rank correlations, we evaluate reward quality at the instance level through an arena-style human study. Our scorers achieve strong agreement with human judgments and substantially outperform VLM-based judges in per-vote agreement on model pairs. Crucially, these scorers enable optimization rather than evaluation alone: they offer task-grounded learning signals that are unambiguous and directionally meaningful. Experiments show that using them as rewards yields

Table 1: **Comparison of VBVR-Pro with selected visual reasoning benchmarks and datasets.** This table shows a representative selection; the full comparison is provided in Section D.4. *VLM* denotes VLM-only judging, *Mixed* combines VLM judgment with limited programmatic checks, and *Verifiable* uses verifiable scorers for all benchmark tasks. VBVR-Pro offers the broadest task coverage and largest scale, supports multiple modalities, and enables fully verifiable evaluation.

Dataset	#Tasks	#Images	#Videos	Split			Modality			Eval
				#SFT	#RL	#Test	Image	Interleaved	Video	
VideoThinkBench (Tong et al., 2026)	48	6,845	0	0	0	4,149	–	–	✓	Mixed
Ruler-Bench (He et al., 2025)	40	493	0	0	0	622	–	–	✓	VLM
RISEBench (Zhao et al., 2025)	16	415	0	0	0	360	✓	–	–	VLM
Zebra-CoT (Li et al., 2026)	18	921,039	0	182,384	0	0	–	✓	–	–
RealUnify (Shi et al., 2026)	32	587	0	0	0	1,000	–	✓	–	VLM
Uni-MMMU (Zou et al., 2026)	8	3,544	0	0	0	885	–	✓	–	Mixed
VBVR (Wang et al., 2026a)	150	2,015,000	1,007,500	1,000,000	0	500	–	–	✓	Verifiable
VBVR-Pro	**300**	**3,471,558**	**1,300,500**	**1,250,000**	**50,000**	**500**	✓	✓	✓	**Verifiable**

steady improvements even on strong baselines, demonstrating that verifiable rewards can serve as a practical foundation for multi-task reinforcement learning in large-scale visual reasoning.

Third, it remains unclear which generative modality best supports visual reasoning. Video generation offers continuous temporal trajectories, image generation can provide a compact solution in a single visual output, and interleaved generation offers integrated visual-textual reasoning (Gu et al., 2026; Li et al., 2026). However, these paradigms have rarely been compared under the same task distribution and verification protocol. Using the same task suite and scorers, VBVR-Pro enables controlled comparison of 28 image, interleaved text-image, and video generators across model scale, output format, and modality configuration. Our results show that video generation performs best on tasks requiring persistent spatiotemporal state tracking, while interleaved generation offers a compute-efficient alternative by externalizing intermediate visual states. In contrast, image-only generation often lacks the capacity to express procedural or temporal reasoning. Moreover, degrading language reasoning in interleaved models has limited effect compared with removing visual states. Further middle-state intervention experiments show that intermediate visual states are causally used by the model: corrupting or replacing them can predictably degrade or redirect the final answer. These results suggest that visual trajectories, rather than language-based chains of thought, form a critical substrate for native visual reasoning.

In summary, VBVR-Pro establishes a closed-loop foundation for scaling native visual reasoning through generation. It provides: 1) a scalable training suite of procedurally generated visual reasoning environments; 2) verifiable reward scorers that are human-aligned and usable for reinforcement learning; 3) a model suite spanning image, video, and interleaved generators, trained and compared under a shared protocol to enable diagnosis and analysis. Together, these components make native visual reasoning trainable, verifiable, optimizable, and experimentally controllable.

2 VBVR-Pro-Dataset: Task Scaling for Transferability

A central challenge in visual reasoning is the scarcity of large-scale, diverse training data. Prior work (Wang et al., 2026a) shows that large-scale synthetic supervision can improve video reasoning, but its transfer to broader visual reasoning benchmarks remains limited. We hypothesize that a key bottleneck is task coverage rather than instance count alone. Guided by this hypothesis, we construct VBVR-Pro-Dataset, a large-scale dataset with 300 tasks spanning diverse visual environments and reasoning capabilities. VBVR-Pro-Dataset is built through a large-scale collaborative effort involving more than 50 researchers and engineers worldwide. Rather than enumerating isolated tasks, we seek to construct a broad and trainable task space that covers visual reasoning capabilities of practical interest to the community.

Task Construction. We adapt the five-faculty cognitive framework of VBVR (Wang et al., 2026a), allowing tasks to carry multiple faculty labels (details in Section D.1). VBVR-Pro is built from 300 task-specific generators under a unified framework, including 150 reimplemented and revised from VBVR (Wang et al., 2026a) and 150 newly designed to broaden task coverage. Each generator samples a configuration from a structured parameter space, such as grid size, object count, layout, visual appearance, and difficulty level, and then instantiates a corresponding problem. Each generated in-

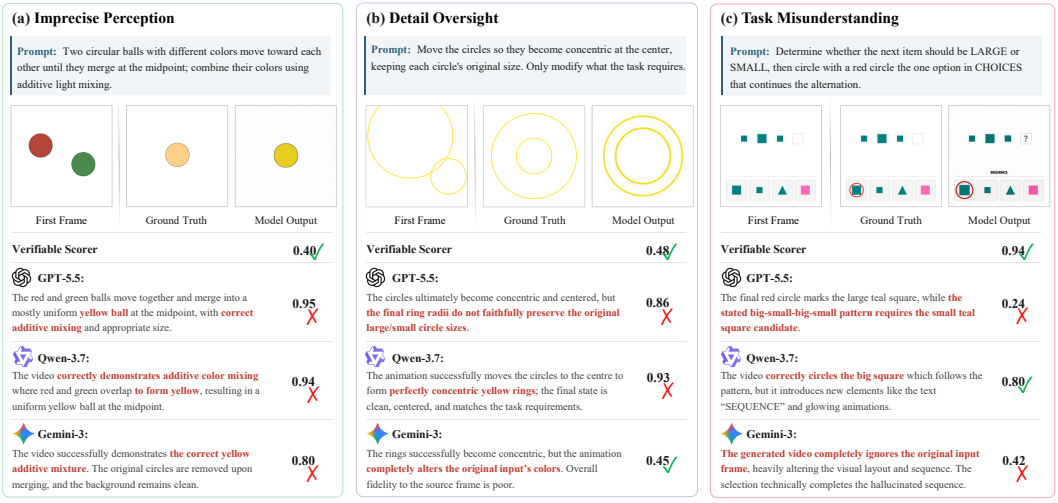

Figure 3: **Three failure modes of VLM-as-a-judge.** Each example includes the ground truth, model output, and the scores and explanations from three strong VLM judges.

stance records its random seed, problem definition, solution, and key attributes to support verifiable scoring, deduplication, and dataset split construction (Section 3).

Multi-modality. Each instance in VBVR-Pro is rendered into aligned video and image modalities. Depending on the task, the image modality uses three output regimes: *Last-Frame* shows only the final state for tasks whose solutions can be represented by a single image, *Key-Frame* presents selected states that capture the essential transitions in the solution process, and *Multi-Frame* uniformly samples intermediate states for tasks that evaluate process integrity or temporal continuity. Representative examples of these three regimes are shown in Section D.2. We highlight that videos and images are different renderings of the same underlying problem with equivalent solutions, enabling controlled comparison under identical task conditions. Moreover, for the interleaved text-image setting, we use Gemini-3.1-Pro-Preview (Google DeepMind, 2026) to generate textual descriptions of intermediate solution steps (details in Section D.3).

Statistics and Analysis. Among the 300 tasks in VBVR-Pro, 50 are held out for out-of-domain evaluation. The remaining 250 tasks each contribute 5,000 instances, yielding 1.25M training instances. VBVR-Pro-Bench comprises the 50 held-out tasks and 50 in-domain tasks selected from the 250 training tasks. Additionally, a reinforcement-learning training split covers these same 50 in-domain tasks and is used in Section 5. Table 1 compares VBVR-Pro with representative visual reasoning benchmarks and datasets. Among them, VBVR-Pro is the largest training source and uniquely supports video, image, and interleaved modalities under a unified task distribution. Figure 2 shows representative tasks from VBVR-Pro, colored according to their corresponding cognitive faculties. Compared with the 150 reworked tasks adapted from VBVR, the 150 newly developed tasks are visually more complex and require deeper reasoning; a detailed analysis is provided in Sections D.5 and D.6. Furthermore, we show that training on VBVR-Pro consequently yields stronger and more consistent performance on external video reasoning benchmarks in Table 4.

3 Verifiable Reward Scorers

A trustworthy benchmark requires accurate, inexpensive, and reproducible evaluators. However, *VLM-as-a-judge* paradigm often fails to satisfy these requirements, especially for tasks that demand precise verification. We therefore build *verifiable reward scorers* for all 100 tasks and validate them against large-scale human preferences. Importantly, our scorers enable large-scale multi-task reinforcement learning (Section 5).

Limitations of VLM-as-a-Judge. VLM-as-a-judge is widely used in visual reasoning benchmarks (as shown in Table 1) for its flexibility for open-ended outputs. However, this paradigm

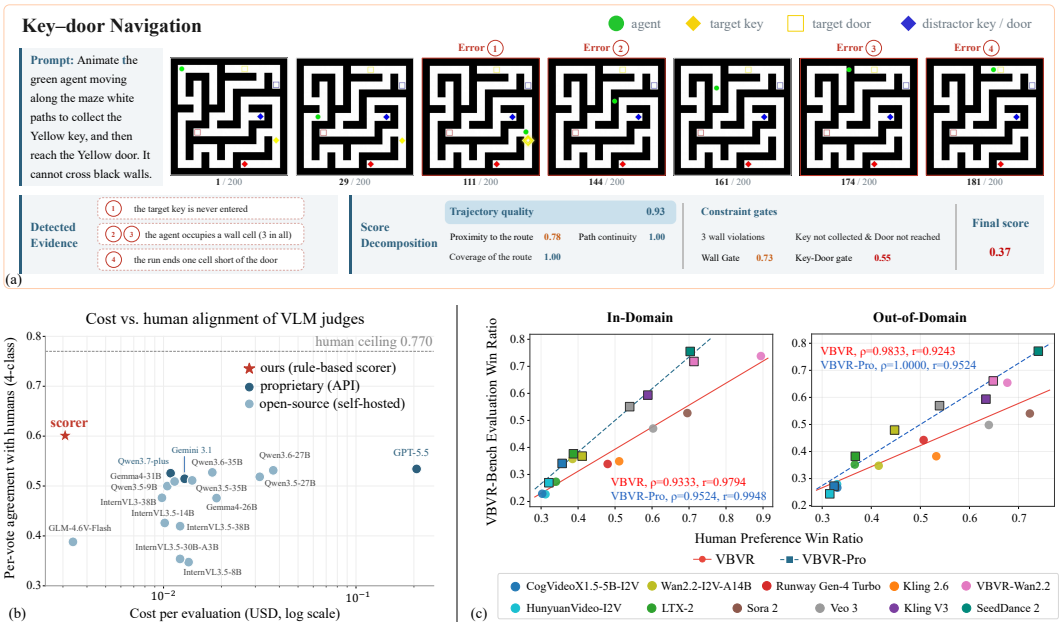

Figure 4: **Task-specific scoring and evaluation.** (a) The scorer evaluates trajectory quality and applies constraint gates to produce a final score. (b) Per-vote agreement with humans versus cost per evaluation (log scale). (c) Model-level agreement. Each point denotes one generation model, with its scorer-based win ratio on the y-axis and human win ratio on the x-axis.

is problematic when used either as an evaluation metric or as a reward function for reinforcement learning (Liu & Zhang, 2025; Motamed et al., 2025; 2026). First, VLM judges can be inaccurate. As illustrated in Figure 3, they may be imprecise in fine-grained visual perception, overlook decisive errors, or misunderstand the task, leading to overestimation of incorrect outputs and underestimation of correct ones. Second, VLM judges are expensive, especially for reinforcement learning: proprietary judges incur large API costs, while open-source judges consume GPU resources for rollouts. Our efficient scorers are significantly cheaper, as shown in Figure 4(b). Third, VLM judges are not reliably reproducible. Even at temperature 0, repeated evaluations on the same videos produce different scores for 55–93% of samples, whereas our deterministic scorers produce identical results across runs (see Table 18 in the appendix).

Design of Verifiable Reward Scorers. Our scorers operate on task-relevant semantic entities rather than pixel-level differences. Each scorer locates the objects of interest, extracts their attributes, and applies task-specific computation checks. This semantic design avoids penalizing correct outputs for minor spatial or rendering variations while focusing on task-relevant correctness. We illustrate an example in Figure 4(a), this scorer tracks the agent's trajectory, compares its path length with the shortest valid path length, and applies penalties for wall crossings, failure to collect the target key and reach the correct door to obtain the final score. Note that these lightweight computations require no API calls, use GPU computation only optionally for OCR, are fully deterministic, and provide interpretable error decomposition.

Human Preference Alignment. To evaluate accuracy efficiently, we compare our scorers with multiple VLM judges against human preferences over all 28 pairs of outputs from eight video generation models, using one instance from each of the 100 tasks in VBVR-Pro-Bench and ten annotations per pair. We measure per-vote agreement under a universal Arena protocol (Zheng et al., 2023) and estimate evaluation cost from API usage or GPU runtime. Full details are provided in Section E.2. As shown in Figure 4(b), our scorers achieve the highest agreement with human preferences while remaining the least expensive evaluator. Its per-vote agreement exceeds 0.60, outperforming GPT-5.5 (OpenAI, 2026a) (0.54) and Gemini-3.1-Pro (Google DeepMind, 2026) (0.52), and reaches 78% of the human agreement ceiling of 0.77. We note that perfect agreement is not expected, since human annotators can disagree on close comparisons. Beyond agreement on individual comparisons, a benchmark should reliably rank models. At the model level, scorer-based and human win ratios

Table 2: Benchmark results on **VBVR-Pro-Bench**. Overall, In-Domain (ID), Out-of-Domain (OOD), and category-wise breakdowns are reported. Samples with multiple category tags contribute to each corresponding category, but are counted only once when computing ID/OOD averages and the Overall score. **Bold**: best in subgroup; underline: second best.

		In-Domain by Category						Out-of-Domain by Category					
Models	**Overall**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**
Human	0.974	0.960	0.990	0.968	0.889	0.960	0.988	0.988	1.000	1.000	0.983	1.000	0.968
Image Generation Models													
Qwen-Image-2.0 (Zhao et al., 2026)	0.313	0.248	0.350	0.220	0.300	0.187	0.186	0.378	0.463	0.292	0.440	0.384	0.088
Seedream-5.0-Pro (ByteDance Seed, 2026)	**0.557**	**0.485**	**0.673**	**0.349**	**0.679**	**0.441**	**0.307**	**0.629**	**0.689**	**0.565**	**0.744**	**0.559**	**0.223**
BAGEL-7B-MoT (Deng et al., 2025)	0.089	0.066	0.050	0.095	0.089	0.050	0.038	0.111	**0.272**	0.039	0.082	0.028	**0.133**
FLUX.2-dev (Black Forest Labs, 2025)	**0.157**	**0.108**	0.114	**0.122**	0.096	0.110	**0.093**	**0.206**	0.268	**0.205**	**0.207**	**0.241**	0.085
Qwen-Image-Edit (Wu et al., 2025a)	0.134	**0.108**	**0.120**	0.092	**0.134**	**0.120**	0.079	0.159	0.239	0.078	0.158	0.182	0.091
VBVR-Pro-BAGEL	0.172	0.168	0.259	0.117	0.146	0.234	0.078	0.176	0.345	0.130	0.167	0.015	0.160
VBVR-Pro-FLUX.2	**0.407**	**0.484**	**0.628**	**0.361**	**0.490**	**0.494**	**0.475**	**0.330**	**0.490**	**0.338**	**0.287**	**0.454**	0.141
VBVR-Pro-Qwen-Image-Edit	0.322	0.332	0.387	0.243	0.257	0.474	0.313	0.311	0.464	0.297	0.262	0.413	**0.200**
Interleaved Text-image Generation Models													
GPT-Image-2 (OpenAI, 2026b)	0.507	0.428	0.593	0.356	0.571	0.227	**0.424**	0.587	0.540	0.513	0.712	0.480	**0.335**
Nano Banana Pro (Raisinghani, 2025)	**0.564**	**0.480**	**0.674**	**0.473**	**0.682**	**0.313**	0.246	**0.648**	**0.751**	**0.619**	**0.739**	**0.585**	0.243
ThinkMorph-7B (Gu et al., 2026)	0.154	0.113	0.130	0.092	0.134	0.162	0.044	0.195	0.238	0.206	0.183	0.253	0.114
VBVR-SenseNova-U1 (Wang et al., 2026a)	**0.408**	**0.469**	**0.463**	**0.351**	**0.497**	**0.425**	**0.674**	**0.347**	**0.395**	**0.393**	**0.309**	**0.480**	**0.263**
VBVR-Pro-ThinkMorph	0.373	0.402	0.524	0.385	0.317	0.500	0.259	0.344	0.499	0.278	0.268	0.535	0.284
VBVR-Pro-SenseNova-U1	**0.638**	**0.811**	**0.843**	**0.778**	**0.828**	**0.847**	**0.764**	**0.464**	**0.651**	**0.407**	**0.387**	**0.558**	**0.451**
Video Generation Models													
Veo 3.1 (Google DeepMind, 2025)	0.309	0.312	0.358	0.335	0.335	0.294	0.222	0.305	0.414	0.290	0.283	0.312	0.242
Kling VIDEO 3.0 (Kuaishou Technology, 2026)	0.392	0.356	0.277	0.365	0.426	0.390	0.323	0.427	0.399	**0.700**	0.421	0.242	0.455
Seedance 2.0 (Team Seedance et al., 2026)	**0.499**	**0.451**	**0.439**	**0.405**	**0.471**	**0.515**	**0.435**	**0.547**	**0.501**	0.634	**0.538**	**0.538**	**0.588**
HunyuanVideo-I2V (Kong et al., 2024)	0.054	0.054	0.030	0.072	0.020	0.092	0.046	0.053	0.120	0.018	0.032	0.062	0.061
CogVideoX1.5-5B-I2V (Yang et al., 2025)	0.085	0.100	0.079	0.132	0.092	0.101	0.085	0.070	0.169	0.047	0.058	0.040	0.026
Wan2.1-I2V-14B-720P (Team Wan et al., 2025)	0.100	0.105	0.068	0.141	0.121	0.113	0.073	0.095	0.152	0.090	0.080	0.123	0.049
Wan2.2-TI2V-5B (Team Wan et al., 2025)	0.094	0.066	0.037	0.082	0.067	0.091	0.044	0.122	0.212	0.064	0.119	0.063	0.110
Wan2.2-I2V-A14B (Team Wan et al., 2025)	0.182	0.157	0.106	0.147	0.147	0.178	0.221	0.207	0.304	0.172	0.157	0.195	0.301
LTX-2.3-I2AV (HaCohen et al., 2026)	0.112	0.106	0.080	0.122	0.093	0.146	0.077	0.119	0.219	0.168	0.096	0.091	0.056
VBVR-Wan2.2 (Wang et al., 2026a)	**0.517**	**0.548**	**0.308**	**0.558**	**0.446**	**0.623**	**0.834**	**0.486**	**0.421**	**0.426**	**0.389**	**0.732**	**0.756**
VBVR-Pro-LTX2.3	0.425	0.527	0.531	0.571	0.461	0.506	0.551	0.324	0.518	0.134	0.201	0.477	0.427
VBVR-Pro-Wan2.1-I2V-14B	0.562	0.730	0.802	0.649	0.603	0.744	0.880	0.395	0.557	0.379	0.259	0.617	0.486
VBVR-Pro-Wan2.2-TI2V-5B	0.470	0.641	0.687	0.623	0.498	0.621	0.786	0.300	0.453	0.158	0.181	0.505	0.452
VBVR-Pro-Wan2.2-I2V-A14B	**0.670**	**0.808**	**0.821**	**0.768**	**0.742**	**0.826**	**0.898**	**0.532**	**0.651**	**0.519**	**0.393**	**0.679**	**0.763**

are highly correlated, as shown in Figure 4(c). Compared with the VBVR scorers (Wang et al., 2026a), VBVR-Pro improves correlation on every split, reaching Spearman's $\rho = 0.95$ in-domain and $\rho = 1.00$ out-of-domain, with Pearson's $r \geq 0.95$ on both splits.

4 Benchmarking Native Visual Reasoning

This section benchmarks leading proprietary and open-source generative models on VBVR-Pro-Bench. We evaluate overall performance, compare behavioral differences across image, interleaved, and video modalities, and examine transferability to unseen tasks and benchmarks. Additional appendix analyses include Chain-of-Step (Wang et al., 2026c) visualizations of native visual reasoning (Section B.3), counterfactual diagnostics assessing whether trained models acquire reusable reasoning trajectories rather than relying on superficial shortcuts (Section B.4), and nearest-neighbor analyses that distinguish transferable reasoning from surface-level memorization(Section B.5).

TAKEAWAY 1: Open-source models can rival proprietary models with sufficient training in reasoning, yet both remain far below human-level performance.

Table 2 compares 28 proprietary and open-source models on VBVR-Pro-Bench. Top-tier proprietary models, such as Seedance 2.0 and GPT-Image-2, perform strongly, indicating that current generative models already possess certain nontrivial visual reasoning capabilities. While most academic open-source models lag behind, suggesting that limited availability of task-relevant training data may be a key bottleneck. Consistent with this observation, training on VBVR-Pro-Dataset improves all nine trained models across image, interleaved text-image, and video generation, with an average overall gain of 0.290 and an average in-domain (ID) gain of $+0.401$ on unseen instances from training task distributions. Nevertheless, even the strongest trained model remains far below human performance, leaving considerable room for improvement.

Table 3: Ablation on **interleaved generation models**. **Text** indicates whether the intermediate trajectory contains meaningful text (*Full*) or semantically uninformative placeholders (*Placeholder*), while **Images** indicates whether it contains multiple visual states (*Multi*) or a single image (*Single*).

		Modalities			In-Domain by Category						Out-of-Domain by Category					
#	Models	Text	Images	Overall	Avg.	Abst.	Know.	Perc.	Spat.	Trans.	Avg.	Abst.	Know.	Perc.	Spat.	Trans.
1-1	ThinkMorph	Full	Multi	0.373	**0.402**	**0.524**	0.385	0.317	**0.500**	0.259	0.344	0.499	**0.278**	0.268	0.535	0.284
1-2		Full	Single	0.349	0.382	0.522	0.324	0.325	0.430	**0.309**	0.316	0.452	0.177	0.254	**0.538**	0.259
1-3		Placeholder	Multi	**0.374**	0.399	0.500	**0.392**	**0.333**	0.484	0.259	**0.348**	**0.506**	0.233	**0.295**	0.479	**0.294**
2-1	SenseNova-U1	Full	Multi	**0.638**	0.811	0.843	**0.778**	0.828	**0.847**	0.764	**0.464**	**0.651**	0.407	**0.387**	0.558	**0.451**
2-2		Full	Single	0.527	0.643	0.829	0.520	0.811	0.616	0.456	0.412	0.621	0.343	0.359	**0.578**	0.220
2-3		Placeholder	Multi	0.629	**0.821**	**0.859**	0.765	**0.880**	0.807	**0.815**	0.436	0.637	**0.436**	0.370	0.468	0.375

TAKEAWAY 2: Video leads in performance, but dense visual states are not cost-efficient.

The unified task setting across image, interleaved text-image, and video generation provides a controlled setting for studying how generation modality affects visual reasoning. Although video generation naturally benefits from dense temporal continuity, after reasoning-intensive training, the strongest interleaved model, VBVR-Pro-SenseNova-U1, performs comparably to the strongest video model, VBVR-Pro-Wan2.2-I2V-A14B, on many ID tasks. This suggests that, once models learn an effective visual representation of the task-solving process from data, sparse but task-relevant visual states can suffice for effective reasoning. Together with lower generation and inference costs, interleaved generation therefore offers a favorable accuracy–efficiency trade-off.

Video generation retains an advantage when reasoning depends on accurately modeling state transitions. Video models perform better on ID transformation tasks and across most OOD categories, indicating that dense temporal representations better support fine-grained state tracking and transfer of learned transition dynamics. As illustrated in Figure 5, the video model produces coherent intermediate states and correct solutions for both shortest-path object manipulation and 180° block rotation, whereas the interleaved and single-image models fail. Single-image generation is weakest overall, particularly on transformation tasks. This is likely because a single endpoint image lacks explicit representations of intermediate states needed for multi-step transitions.

TAKEAWAY 3: Intermediate visual states matter more than language-based reasoning.

To disentangle the contributions of textual and visual trajectory supervision, we conduct an ablation study on interleaved text-image generation models in Table 3. Exp. 1-1 and Exp. 2-1 are the reference VBVR-Pro-ThinkMorph and VBVR-Pro-SenseNova-U1 models, trained on complete trajectories with meaningful reasoning text and multiple intermediate images. Exp. 1-2 and Exp. 2-2 retain the reasoning text but compress each visual trajectory into a single image, thereby isolating the contribution of intermediate visual supervision. This reduces the overall score by 0.024 and 0.111, respectively. In contrast, Exp. 1-3 and Exp. 2-3 preserve the multi-image trajectory while replacing intermediate text with semantically uninformative placeholders, causing much smaller degradation.

These findings suggest that, for visual reasoning tasks in VBVR-Pro, explicit visual state transitions contribute more than intermediate textual reasoning. Many tasks, such as maze navigation and object translation or rotation, inherently require tracking spatial states, paths, directions, and orientations. A sequence of intermediate visual states exposes these transitions directly and therefore provides a natural representation of the underlying reasoning process.

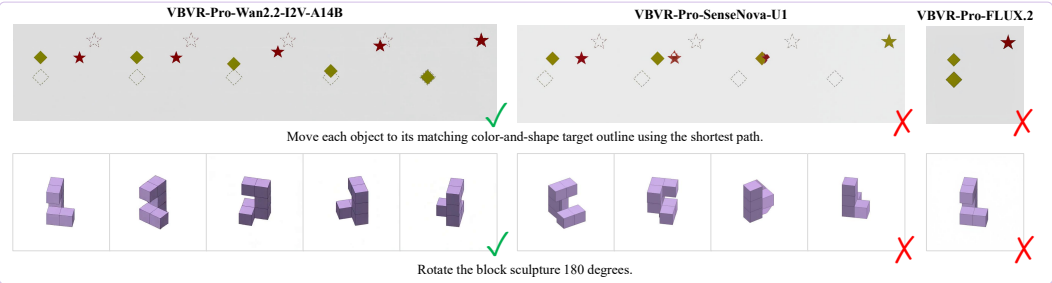

Figure 5: Qualitative comparison across visual reasoning modalities on transformation tasks.

Table 4: **Results on other video reasoning benchmarks.** Training on VBVR-Pro-Dataset transfers across diverse visual reasoning tasks. **Bold**: best overall; underline: second best overall.

Models	RISE-Video (Liu et al., 2026c)	V-ReasonBench (Luo et al., 2025b)	RULER-Bench (He et al., 2025)	MME-COF-Pro (Qi et al., 2026)	VideoThinkBench(mini) (Tong et al., 2026)	BabyVision-Gen (Chen et al., 2026b)	IntelligentVBench (Pan et al., 2026)
Veo-3.1	**76.40**	24.25	65.19	**55.90**	27.69	–	3.74/5
CogVideoX1.5-5B-I2V	46.99	7.87	42.27	28.46	22.83	1.07	3.50/5
LTX-2.3-I2AV	55.23	4.51	49.14	34.05	28.17	2.14	3.88/5
Wan2.2-I2V-A14B	62.83	10.21	57.89	24.76	26.21	0.36	3.93/5
VBVR-Wan2.2	65.94	18.05	56.28	34.13	29.22	4.64	4.08/5
VBVR-Pro-Wan2.2-I2V-A14B	66.18	**38.22**	**66.96**	48.39	**52.86**	**12.50**	**4.11/5**

TAKEAWAY 4: Reasoning capabilities emerged can transfer to unseen, real-life scenarios.

Training also improves transferability beyond the training tasks. We first evaluate out-of-domain (OOD) transfer within VBVR-Pro-Bench, where test tasks are entirely unseen during training but retain similar visual styles. Trained models improve by $+0.179$ on average, showing that learned reasoning skills transfer beyond the training task distributions. We then test broader generalization to diverse real-life reasoning problems on *seven* external video benchmarks using the strongest model, VBVR-Pro-Wan2.2-I2V-A14B. As shown in Table 4, it consistently outperforms its Wan2.2-I2V-A14B base model, with gains of up to 28 points across diverse domains, including shape fitting and analogy solving on V-ReasonBench and embodied reasoning on MME-CoF-Pro. Nearest-neighbor analysis further suggests that these gains reflect transferable visual reasoning rather than surface-level memorization; detailed results and analyses are provided in Section B.5.

5 Reinforcement Learning with Verifiable Rewards

A key goal of this work is to examine the feasibility of large-scale multi-task reinforcement learning (RL) for visual reasoning. Notably, this is not a straightforward extension of RL for general visual generation: while conventional generative tasks often reward pixel-level fidelity or aesthetics, native visual reasoning requires satisfying high-level goals expressed through semantic transformations and task-specific constraints. In this section, we present an end-to-end RL baseline for visual reasoning, introducing the training strategies (Section 5.1) and analyzing their performance (Section 5.2).

5.1 Algorithms

TAKEAWAY 5: Visual reasoning RL requires stronger rollout noise for semantic exploration.

RL for visual reasoning requires exploration over diverse *semantic* decisions. However, increasing stochasticity in conventional SDE sampling (Liu et al., 2025) can introduce excessive noise and visual artifacts, limiting the exploration achievable without degrading rollout quality. We therefore adopt Coefficients-Preserving Sampling (CPS) (Wang & Yu, 2025), which balances predicted and

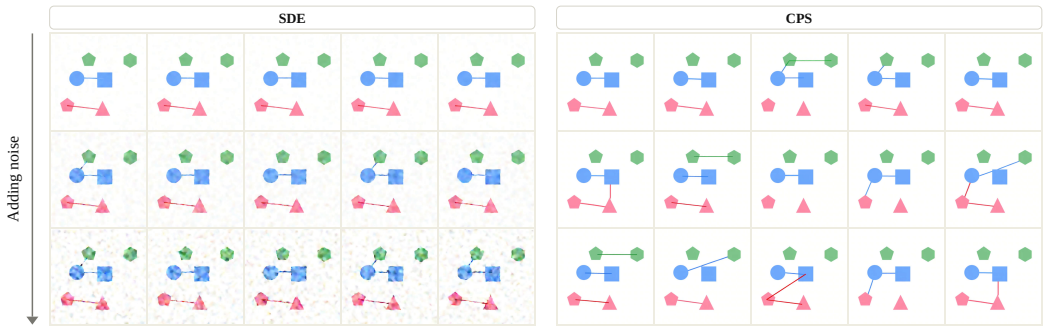

Figure 6: **Semantic exploration with visual fidelity.** SDE and CPS generate four rollouts from the same prompt at each of three increasing noise settings. Stronger CPS noise produces diverse connection decisions while preserving clean shapes and backgrounds, whereas SDE introduces increasing visual corruption. Noise settings are not directly calibrated across the two samplers.

Table 5: **Reinforcement learning results**. Overall, In-Domain, Out-of-Domain, and category-wise breakdowns are reported. *Human* denotes the human-preference win rate over six models. **Bold**: best; underline: second best. The *baseline* is VBVR-Pro-Wan2.2-TI2V-5B, already SFT-trained on the VBVR-Pro-Dataset; *+SFT** applies SFT on the RL training set. Inference details are described in Section C.2.3, with additional sampler results in Section C.4.

		In-Domain					Out-of-Domain						
Model	**Overall**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**	**Human**
Baseline	0.470	0.641	0.687	0.623	0.498	0.621	0.786	0.300	0.453	0.158	0.181	0.505	0.452	0.473
+ SFT*	0.502	0.677	**0.731**	0.629	0.545	0.662	0.840	0.327	0.514	0.163	0.190	0.555	**0.497**	0.522
+ RLVR	**0.544**	**0.711**	0.724	**0.665**	**0.648**	**0.689**	**0.857**	**0.377**	**0.574**	**0.283**	**0.266**	0.551	0.428	**0.541**
+ RLVR-SDE	0.499	0.658	0.695	0.635	0.529	0.622	0.832	0.340	0.518	0.216	0.217	**0.579**	0.425	0.519
+ RLVLM	0.503	0.660	0.661	0.640	0.579	0.624	0.818	0.345	0.534	0.263	0.239	0.467	0.429	0.509
+ RLMSE	0.419	0.561	0.562	0.561	0.539	0.423	0.744	0.278	0.461	0.139	0.195	0.288	0.430	0.437

newly sampled noise to preserve the scheduler-prescribed noise scale. As illustrated in Figure 6, this enables more diverse task-level decisions while largely preserving visual fidelity. Further discussion is provided in Section C.1 of the appendix.

We compare five training strategies against the original model, referred to as the baseline: **Supervised fine-tuning (SFT)** trains with a supervised objective; **Reinforcement learning with VLM rewards (RLVLM)** uses Qwen3.6-27B (Qwen Team, 2026a) to judge generated samples and provide rewards for policy optimization; **Reinforcement learning with verifiable rewards (RLVR)** uses the proposed VBVR-Pro scorers to compute task-specific rewards deterministically from ground-truth metadata; **RLVR-SDE** uses the same verifiable reward and RL objective as RLVR but replaces CPS with conventional Flow-SDE sampling to probe the role of the rollout sampler; and **Reinforcement learning with MSE rewards (RLMSE)** uses a mean-squared-error-based reward as a low-level reconstruction signal, serving as a control to test whether pixel-level similarity provides a useful objective for native visual reasoning. RLVLM and RLVR differ only in reward source. Data and training details are provided in Section C.2 of the appendix.

5.2 Results and Analysis

Table 5 shows that RLVR achieves the highest overall score, while RLVLM offers only a marginal gain over SFT. The gap between RLVR and RLVLM supports the benefit of task-grounded verifiable rewards over VLM feedback. RLVR's advantage over RLVR-SDE is consistent with the benefit of preserving visual fidelity during rollout exploration. Its advantage over RLMSE supports using task-grounded rewards rather than pixel-level similarity.

RLVR and RLVLM generalize to unseen tasks despite using no out-of-domain data during RL. The larger RLVR gain suggests that verifiable feedback better supports transferable visual reasoning. We additionally report human preference win rates, computed as in Section 3 over all the six models, supporting the benefit of verifiable rewards over VLM-based rewards in this large-scale RL training. Further analysis of training dynamics and sampler sensitivity, together with qualitative case studies, is provided in Section C of the appendix.

6 Conclusion

By unifying procedurally generated visual reasoning environments, task-grounded verifiable rewards, reinforcement learning, and modality comparisons, VBVR-Pro establishes native visual reasoning through generation as a trainable, verifiable, and experimentally controllable research problem. Our transfer studies, counterfactual diagnostics, visual-state interventions, trajectory visualizations, and reward-driven optimization collectively suggest that visual generative models can construct, maintain, and refine intermediate visual trajectories, suggesting that visual generation can serve as a medium for reasoning, not merely as an output format. These findings call for greater attention to native visual reasoning as a complementary paradigm to language-centric reasoning, and we hope VBVR-Pro provides a useful foundation for the future research.

7 AI USE STATEMENT

The research design and methodological decisions were made by the authors, including the reasoning tasks, task variables, dataset generation procedures, and the evaluation criteria, individual checks, and tools used by each verifiable reward scorer. Generative AI tools assisted with writing portions of the generator and scorer code according to these author-defined specifications. We also used generative AI tools to improve the grammar, clarity, and readability of text originally written by the authors.

All AI-assisted work was conducted under human supervision. The authors manually reviewed the AI-assisted code, inspected its execution results, and checked all revised text for accuracy and consistency with the intended meaning. The reliability of the scorers was additionally assessed through human annotation. We take full responsibility for the final content of this work, including the manuscript, code, datasets, and all AI-assisted contributions.

8 ETHICS STATEMENT

This work introduces a dataset for visual reasoning and an accompanying verifiable reward scorer, which has been tested in large-scale reinforcement learning. The dataset is constructed by the authors and consists of logic-based reasoning tasks. It contains no personal or sensitive information. Human annotators were engaged to assess the reliability of the proposed verifiable scorers; their annotation tasks involved only the reasoning content and did not require the disclosure of personal or sensitive information. We release the dataset under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license.

9 REPRODUCIBILITY STATEMENT

The dataset construction procedures and task specifications are described in Section 2, and the verifiable reward scorers are detailed in Section 3. The experimental setup, including the models trained on the proposed dataset, their training procedures and configurations, the subsequent reinforcement learning experiments, and the evaluation protocols, is detailed in Section 4 and Section 5. To preserve anonymity, we omit links to the dataset, verifiable reward scorer code, trained model checkpoints, and reinforcement learning code. We will provide public links to all these resources following the notification of decisions.

REFERENCES

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Tianyi Wei, Xiaohang Zhan, Jiaqi Wang, Tong Wu, Xingang Pan, and Dahua Lin. AdaGRPO: A capability-aware adaptive enhancement for flow-based GRPO. *arXiv preprint arXiv:2606.06828*, 2026.

ByteDance Seed. Beyond generation, it understands design: Introducing Seedream 5.0 Pro, July 2026. URL https://seed.bytedance.com/en/blog/beyond-generation-it-understands-design-introducing-seedream-5-0-pro.

Chao Chen, Zhixin Ma, Yongqi Li, Yupeng Hu, Yinwei Wei, Wenjie Li, and Liqiang Nie. Reasoning in the dark: Interleaved vision-text reasoning in latent space. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39117–39129, 2026a.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Haozhe Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. Charles, et al. BabyVision: Visual reasoning beyond language. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b. URL https://arxiv.org/abs/2601.06521.

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

Dongjie Cheng, Yongqi Li, Zhixin Ma, Hongru Cai, Yupeng Hu, Wenjie Wang, Liqiang Nie, and Wenjie Li. Omni-R1: Towards the unified generative paradigm for multimodal reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38907–38924, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1937. URL https://aclanthology.org/2026.findings-acl.1937/.

Guo Cheng, Danni Yang, Ziqi Huang, Jianlou Si, Chenyang Si, and Ziwei Liu. RealDPO: Real or not real, that is the preference. *arXiv preprint arXiv:2510.14955*, 2025.

Junhao Cheng, Liang Hou, Tianxiong Zhong, Xin Tao, Pengfei Wan, Kun Gai, and Jing Liao. VLMs are good teachers for video reasoning via adaptive test-time optimization. *arXiv preprint arXiv:2606.02564*, 2026b. URL https://arxiv.org/abs/2606.02564.

Xuanlang Dai, Yujie Zhou, Long Xing, Jiazi Bu, Xilin Wei, Yuhong Liu, Beichen Zhang, Kai Chen, and Yuhang Zang. EndoCoT: Scaling endogenous chain-of-thought reasoning in diffusion models. In *Computer Vision – ECCV 2026*, volume 17009 of *Lecture Notes in Computer Science*, pp. 621–638. Springer, 2026. doi: 10.1007/978-3-032-37035-8_34.

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

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. SenseNova-U1: Unifying multimodal understanding and generation with NEO-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Qihua Dong, Yitian Zhang, Huimin Zeng, Yizhou Wang, Jianglin Lu, Kuo Yang, and Yun Fu. Revealing the seen, imagining the beyond: A survey of image-grounded chain-of-thought reasoning in multimodal LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45055–45070, 2026a. doi: 10.18653/v1/2026.acl-long.2087.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29316–29335, 2026b.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40748–40774, 2025. doi: 10.52202/085713-1218.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

GLM-V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Google DeepMind. Introducing Veo 3.1 and advanced capabilities in Flow. Google blog, https://blog.google/innovation-and-ai/products/veo-updates-flow/, October 2025.

Google DeepMind. Gemini 3.1 Pro: Best for complex tasks and bringing creative concepts to life, February 2026. URL https://deepmind.google/models/gemini/pro/.

Jiawei Gu, Yunzhuo Hao, Huichen Will Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. ThinkMorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=mB3vxfrQZM.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient joint audio-visual foundation model, 2026. URL https://arxiv.org/abs/2601.03233.

Xuan He, Cong Wei, Yuhao Cheng, Linrui Ma, Yuxuan Zhang, Zuojun Li, Yuhao Wen, Jize Jiang, Zeyi Liu, Yuren Hao, Songcheng Cai, Keming Wu, Penghui Du, Kai Zou, Rui Yang, Chenkai Sun, Ke Yang, Ping Nie, Kelsey R. Allen, Chenglong Wang, Michel Galley, Jianfeng Gao, and ChengXiang Zhai. VGI-Bench: Probing visual intelligence in video generation models. *arXiv preprint arXiv:2608.19583*, 2026a.

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

Zefeng He, Xiaoye Qu, Yafu Li, Tong Zhu, Qipeng Guo, Muxin Fu, Siyuan Huang, and Yu Cheng. DiffThinker: Towards generative multimodal reasoning with diffusion models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b. URL https://openreview.net/forum?id=jQ2DRkSir0.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. Liger Kernel: Efficient Triton kernels for LLM training. *arXiv preprint arXiv:2410.10989*, 2024.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379, 2024. doi: 10.52202/079017-4423. URL https://arxiv.org/abs/2406.09403.

Zican Hu, Xuyang Hu, Yiming Liu, Zuwei Long, Wei Liu, Yunzhuo Hao, Jiawei Gu, Linjie Li, Yu Cheng, Zhenhong Sun, Weibo Gu, Xing Sun, and Zhi Wang. Bridging interleaved multimodal reasoning as a unified decision process. *arXiv preprint arXiv:2607.03748*, 2026. URL https://arxiv.org/abs/2607.03748.

Ziqi Huang, Ning Yu, Gordon Chen, Haonan Qiu, Paul Debevec, and Ziwei Liu. VChain: Chain-of-visual-thought for reasoning in video generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 226–250, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.12. URL https://aclanthology.org/2026.findings-acl.12/.

ITU-T. Subjective video quality assessment methods for multimedia applications. ITU-T Recommendation P.910, International Telecommunication Union, Geneva, Switzerland, October 2023. URL https://www.itu.int/rec/T-REC-P.910-202310-S/en.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

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

Minh-Quan Le, Armand Comas, Alexandros Lattas, Stylianos Moschoglou, Pedro Vélez, Amit Raj, Aaron Germuth, Thabo Beeler, Dimitris Samaras, and Di Qiu. Concurrent image understanding and generation: Self-correcting coupled markov jump processes. *arXiv preprint arXiv:2607.13188*, 2026.

Ang Li, Charles L. Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. Zebra-CoT: A dataset for interleaved vision-language reasoning. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=c6XIVI3TiQ.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36340–36364, 2025. URL https://proceedings.mlr.press/v267/li25cz.html.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 45650–45685. Curran Associates, Inc., 2025. doi: 10.52202/085713-1362. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3a10c46572628d58cb44fb705f25cbbf-Paper-Conference.pdf.

Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, and Wanli Ouyang. UniGRPO: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026a. URL https://arxiv.org/abs/2603.23500.

Ming Liu and Wensheng Zhang. Is your video language model a reliable judge? In *International Conference on Learning Representations*, volume 2025, pp. 88570–88589, 2025.

Mingxin Liu, Ziqian Fan, Zhaokai Wang, Leyao Gu, Zirun Zhu, Yiguo He, Yuchen Yang, Changyao Tian, Xiangyu Zhao, Ning Liao, et al. GRADE: Benchmarking discipline-informed reasoning in image editing. In *Computer Vision – ECCV 2026*, volume 17039 of *Lecture Notes in Computer Science*, pp. 536–553. Springer, 2026b. doi: 10.1007/978-3-032-37435-6_29.

Mingxin Liu, Shuran Ma, Shibei Meng, Xiangyu Zhao, Zicheng Zhang, Shaofeng Zhang, Zhihang Zhong, Peixian Chen, Haoyu Cao, Xing Sun, et al. RISE-Video: Can video generators decode implicit world rules? *arXiv preprint arXiv:2602.05986*, 2026c.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepCoder: A fully open-source 14B coder at o3-mini level, April 2025a. URL https://www.together.ai/blog/deepcoder.

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

Saman Motamed, Minghao Chen, Luc Van Gool, and Iro Laina. TRAVL: A recipe for making video-language models better judges of physics implausibility. *arXiv preprint arXiv:2510.07550*, 2025. URL https://arxiv.org/abs/2510.07550.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958. IEEE, 2026.

Kaleb Newman, Tyler Zhu, and Olga Russakovsky. Video models reason early: Exploiting plan commitment for maze solving, 2026. URL https://arxiv.org/abs/2603.30043.

OpenAI. Introducing GPT-5.5, April 2026a. URL https://openai.com/index/introducing-gpt-5-5/.

OpenAI. Introducing ChatGPT Images 2.0, April 2026b. URL https://openai.com/index/introducing-chatgpt-images-2-0/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Kaihang Pan, Qi Tian, Jianwei Zhang, Weijie Kong, Jiangfeng Xiong, Yanxin Long, Shixue Zhang, Haiyi Qiu, Tan Wang, Zheqi Lv, et al. OmniWeaving: Towards unified video generation with free-form composition and reasoning. *arXiv preprint arXiv:2603.24458*, 2026.

Yu Qi, Xinyi Xu, Ziyu Guo, Siyuan Ma, Renrui Zhang, Xinyan Chen, Ruichuan An, Ruofan Xing, Jiayi Zhang, Haojie Huang, et al. MME-CoF-Pro: Evaluating reasoning coherence in video generative models with text and visual hints. In *European Conference on Computer Vision (ECCV)*, 2026.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026a. URL https://qwen.ai/blog?id=qwen3.6-27b.

Qwen Team. Qwen3.7-Plus: Multimodal agent intelligence, June 2026b. URL https://qwen.ai/blog?id=qwen3.7-plus.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Naina Raisinghani. Introducing Nano Banana Pro, November 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 68658–68685. Curran Associates, Inc., 2024. doi: 10.52202/079017-2193. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7ede97c3e082c6df10a8d6103a2eebd2-Paper-Conference.pdf.

Yang Shi, Yuhao Dong, Yue Ding, Yuran Wang, Xuanyu Zhu, Sheng Zhou, Wenting Liu, Haochen Tian, Rundong Wang, Huanqian Wang, et al. RealUnify: Do unified models truly benefit from unification? a comprehensive benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22488–22497, 2026.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026. URL https://arxiv.org/abs/2604.14148.

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, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, et al. Thinking with video: Video generation as a promising multimodal reasoning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41121–41129, 2026.

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

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, Fengyuan Yu, Weihang Xiao, Yizheng Jiao, Jianheng Hou, Danyang Zhang, Pengcheng Xu, Boyang Zhong, Zehong Zhao, Gaoyun Fang, John Kitaoka, Yile Xu, Hua Xu, Kenton Blacutt, Tin Nguyen, Siyuan Song, Haoran Sun, Shaoyue Wen, Linyang He, Runming Wang, Yanzhi Wang, Mengyue Yang, Ziqiao Ma, Raphaël Millière, Freda Shi, Nuno Vasconcelos, Daniel Khashabi, Alan Yuille, Yilun Du, Ziming Liu, Dahua Lin, Ziwei Liu, Vikash Kumar, Yijiang Li, Lei Yang, Zhongang Cai, and Hokin Deng. A very big video reasoning suite. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026a. URL https://arxiv.org/abs/2602.20159.

Pengjie Wang, Linger Deng, Zujia Zhang, Shaojie Zhang, Zhenbo Luo, Pei Fu, Jian Luan, Xiang Bai, and Yuliang Liu. DeltaV: Thinking with visual state updates in unified large multimodal models. *arXiv preprint arXiv:2607.08434*, 2026b.

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. Demystifying video reasoning. In *European Conference on Computer Vision (ECCV)*, 2026c.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

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

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, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. KRIS-Bench: Benchmarking next-level intelligent image editing models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38, 2025b. doi: 10.52202/085713-5242.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649. ACM, 2024. doi: 10.1145/3626772.3657878.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2505.11409.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026. URL https://arxiv.org/abs/2506.17218.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=LQzN6TRFg9.

Beichen Zhang, Yuhong Liu, Jinsong Li, Yuhang Zang, Jiaqi Wang, and Dahua Lin. ETCHR: Editing to clarify and harness reasoning. *arXiv preprint arXiv:2605.23897*, 2026.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-Image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026. URL https://arxiv.org/abs/2605.10730.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49842–49869. Curran Associates, Inc., 2023a. doi: 10.52202/075280-2170. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9c2aa1e456ea543997f6927295196381-Paper-Conference.pdf.

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. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38, 2025. doi: 10.52202/085713-5791.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12): 3848–3860, 2023b. doi: 10.14778/3611540.3611569.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=VJZ477R89F.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Pengfei Zhou, Xiaopeng Peng, Jiajun Song, Chuanhao Li, Zhaopan Xu, Yue Yang, Ziyao Guo, Hao Zhang, Yuqi Lin, Yefei He, et al. OpenING: A comprehensive benchmark for judging open-ended interleaved image-text generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 56–66, 2025. doi: 10.1109/CVPR52734.2025.00015.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Chaorui Deng, Shen Yan, Haoqi Fan, Yejin Choi, et al. When visualizing is the first step to reasoning: MIRA, a benchmark for visual chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26154–26164, 2026.

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

Kai Zou, Ziqi Huang, Yuhao Dong, Shulin Tian, Dian Zheng, Hongbo Liu, Jingwen He, Bin Liu, Yu Qiao, and Ziwei Liu. Uni-MMMU: A massive multi-discipline multimodal unified benchmark. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 908–924. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.40.

A Related Work

Native visual reasoning through visual generation. Native visual reasoning, broadly defined as reasoning through visual generation, has recently received increasing attention. Wiedemer et al. (Wiedemer et al., 2025) observed that video models are reasoners, and identified the *Chain-of-Frames* (CoF) paradigm. Following this direction, OpenCoF (Chen et al., 2026c) introduces learnable reasoning tokens in the video generators. Wang et al. (Wang et al., 2026c) further identify a different mechanism, *Chain-of-Steps* (CoS), where reasoning unfolds through diffusion denoising steps; Newman et al. Newman et al. (2026) likewise locate the decision on the denoising axis, but find that plans commit within the first few steps. Despite the vision-centric nature of this paradigm, MLLMs have also been incorporated to assist video reasoners. Cheng et al. (Cheng et al., 2026b) use MLLMs for test-time optimization of video reasoners, while VChain (Huang et al., 2026) injects MLLM-derived reasoning signals, i.e., Chain-of-Visual-Thought, to guide video generation at inference time. Beyond video, image generation has also been explored as a medium for native visual reasoning. DiffThinker (He et al., 2026b) reformulates visual reasoning as a native image-to-image generative task across four visual reasoning tasks, while EndoCoT (Dai et al., 2026) refines latent thoughts to guide image generation. However, existing studies are typically validated on a limited set of visual tasks, largely due to the lack of unified infrastructure that supports large-scale diverse training, verifiable evaluation, and systematic comparison across generative modalities. This is the gap we aim to address in this work.

Interleaved visual-textual reasoning. The success of language-based reasoning has naturally motivated interleaving intermediate visual states with textual rationales (Dong et al., 2026a). Within the broader paradigm of reasoning through generation, recent methods enable models to sketch with external tools (Hu et al., 2024), visualize reasoning traces (Li et al., 2025), incorporate visual editing into a reasoning harness (Zhang et al., 2026), or learn from large-scale interleaved corpora (Li et al., 2026; Wang et al., 2026b; Gu et al., 2026). A parallel line of work investigates how much explicit visual state is needed, compressing visual reasoning into update tokens (Wang et al., 2026b) or latent tokens (Yang et al., 2026; Dong et al., 2026b; Chen et al., 2026a). Beyond interleaved text-image modeling, SC-CMJP (Le et al., 2026) leverages masked diffusion models to jointly generate text and images for editing and visual reasoning tasks. While these approaches significantly expand the design space of visual reasoning, many of them treat visual states as auxiliary supports for textual reasoning, although some, such as ThinkMorph Gu et al. (2026), treat text and images as complementary. In contrast, our work systematically investigates visual trajectories themselves as first-class reasoning substrates.

Reinforcement learning for visual reasoning. Reinforcement learning for general visual generation has been extensively studied (Liu et al., 2025; Xue et al., 2025; Zheng et al., 2026; Wang & Yu, 2025; Cheng et al., 2025; Bu et al., 2026), primarily to improve instruction following and aesthetic quality. However, visual reasoning tasks require models to reach semantically valid solutions; RL for such visual reasoning objectives remains comparatively underexplored. For interleaved generation, recent approaches jointly optimize text and image policies under unified RL objectives (Liu et al., 2026a; Hu et al., 2026; Cheng et al., 2026a), but their visual states are still commonly evaluated by VLM-based judges. VPRL (Xu et al., 2026) improves large vision models on three visual navigation tasks without language, while VideoRLVR (Zhu et al., 2026) demonstrates the promise of verifiable rewards on three visual tasks. Nevertheless, existing studies remain limited in scale and task diversity. In this work, we perform joint RL training across 50 tasks with verifiable rewards and evaluate the resulting models on 100 tasks, enabling a substantially broader study of multi-task native visual reasoning.

Visual reasoning benchmarks and training sources. Existing visual reasoning benchmarks span video generation (Tong et al., 2026; Luo et al., 2025b; He et al., 2025; Liu et al., 2026c; Pan et al., 2026), image editing (Zhao et al., 2025; Wu et al., 2025b; Chen et al., 2026b; Liu et al., 2026b), and unified multimodal generation (Zhou et al., 2025; Zou et al., 2026; Zhou et al., 2026; Shi et al., 2026). However, most of these benchmarks are evaluation-only (Table 1): they offer little supervision for training and often rely on VLM-based judges, whose evaluations can be difficult to verify and reproduce. A few interleaved corpora reach training scale (Li et al., 2026; Wang et al., 2026b), but they remain limited in task coverage and, more importantly, are not procedurally generated,

making it difficult to obtain verifiable evaluation signals or task-grounded rewards. VBVR (Wang et al., 2026a) provides both large-scale supervision and verifiable evaluation, but its task space can be further expanded toward broader, more challenging, and more transferable visual reasoning capabilities. VBVR-Pro builds on this foundation by providing a larger controlled task space, verifiable reward scorers, multi-task RL training, and systematic modality comparisons across image, video, and interleaved generative models.

B Details of baseline model architectures

B.1 Image, Video, and Interleaved Generators

To evaluate the effectiveness of training on the VBVR-Pro-Dataset, we fine-tune a diverse set of generative foundation models spanning image generation, video generation, and interleaved text-image generation. Their architectural properties are summarized in Table 6, with additional model-specific details provided in Section B. They vary substantially in model scale, cross-modal interaction mechanisms, and output representations. This architectural diversity enables us to examine whether the benefits of fine-tuning on VBVR-Pro-Dataset generalize across different generative modalities and model designs, including sparse and dense architectures as well as latent- and pixel-space generation.

- **Image generation models.**
  - **BAGEL-7B-MoT** (Deng et al., 2025) is a unified multimodal model that supports both multimodal understanding and image generation within a Mixture-of-Transformers (MoT) architecture. Its language-understanding and visual-generation pathways use separate Transformer parameters while participating in a common multimodal sequence, allowing the model to specialize computation by token type. The full checkpoint contains approximately 14B parameters, but only about 7B are active for a given token pathway.
  - **Qwen-Image-Edit** (Wu et al., 2025a) is built on the 20B-parameter Qwen-Image foundation model and uses a multimodal diffusion Transformer (DiT) to perform both image generation and instruction-guided editing. Its dual-stream blocks initially maintain distinct representations for visual latents and multimodal conditioning while enabling interaction between them throughout denoising.
  - **FLUX.2-dev** (Black Forest Labs, 2025)'s Transformer begins with dual-stream blocks that process visual and conditioning tokens separately and then transitions to single-stream blocks that jointly process the concatenated token sequence. Compared with earlier FLUX architectures, FLUX.2 allocates a larger fraction of its depth and parameters to the joint single-stream stage, providing greater capacity for cross-modal composition and reference-image integration.
- **Interleaved text-image generation models.**
  - **ThinkMorph-7B** (Gu et al., 2026) is initialized from BAGEL-7B-MoT and fine-tuned on text-image reasoning trajectories. It therefore retains BAGEL's 14B-total, 7B-active Mixture-of-Transformers architecture and VAE-latent image-generation pathway.
  - **SenseNova-U1-8B-MoT** (Diao et al., 2026) is a native unified model that treats multimodal understanding and generation as operations within a common Transformer rather than as separate encoder and decoder pipelines. Its NEO-unify architecture removes both the conventional visual encoder and the generative VAE. Images are instead converted directly into RGB pixel patches, processed alongside language tokens using modality-specific Transformer parameters, and reconstructed through a lightweight pixel-output head.
- **Video generation models.**
  - **Wan2.2-I2V-A14B** (Team Wan et al., 2025) is a sparse Mixture-of-Experts model with separate high-noise and low-noise denoisers. Each expert is a full 14B-class video DiT, but only one expert is active at each point in the denoising trajectory.

Table 6: Architectural comparison of the base models used for fine-tuning. Parameter counts are reported as total, active parameters where applicable.

Model	Parameters	Generative backbone	Transformer configuration	Generation representation
Image Generation Models				
BAGEL-7B-MoT (Deng et al., 2025)	14B total; 7B active	MoT with separate understanding and generation branches	28 blocks; width 3,584; 28 heads × 128 dims/head	VAE latents
FLUX.2-dev (Black Forest Labs, 2025)	32B	Hybrid dual-stream and single-stream DiT	8 dual-stream + 48 single-stream blocks; width 6,144; 48 heads × 128 dims/head	VAE latents
Qwen-Image-Edit (Wu et al., 2025a)	20B	Dual-stream multimodal diffusion Transformer (MMDiT)	60 dual-stream blocks; width 3,072; 24 heads × 128 dims/head	VAE latents
Interleaved Text-image Generation Models				
ThinkMorph-7B (Gu et al., 2026)	14B total; 7B active	MoT with separate understanding and generation branches	28 blocks; width 3,584; 28 heads × 128 dims/head	VAE latents
SenseNova-U1-8B-MoT (Diao et al., 2026)	16B total; 8B active	MoT with separate understanding and generation branches	42 blocks; width 4,096; 32 heads × 128 dims/head	Raw RGB patches
Video Generation Models				
Wan2.2-I2V-A14B (Team Wan et al., 2025)	27B total; 14B active	Two-expert MoE video DiT with separate high-/low-noise experts	40 blocks per expert; width 5,120; 40 heads × 128 dims/head	VAE latents
Wan2.2-TI2V-5B (Team Wan et al., 2025)	5B	Dense video DiT	30 blocks; width 3,072; 24 heads × 128 dims/head	VAE latents
Wan2.1-I2V-14B-720P (Team Wan et al., 2025)	14B	Dense video DiT	40 blocks; width 5,120; 40 heads × 128 dims/head	VAE latents
LTX-2.3-I2AV (HaCohen et al., 2026)	22B	Asymmetric dual-stream audio–video DiT	48 blocks; width 4,096; 32 heads × 128 dimensions	VAE latents

- **Wan2.2-TI2V-5B** (Team Wan et al., 2025) is a dense video DiT. In contrast to the A14B variant, it uses the same 5B-parameter backbone throughout the denoising trajectory rather than routing computation between noise-specialized experts.
- **Wan2.1-I2V-14B-720P** (Team Wan et al., 2025) is a dense 14B-parameter video DiT that applies a single 40-block Transformer backbone at all denoising timesteps and generates videos as compressed spatiotemporal VAE latents.
- **LTX-2.3-I2AV** (HaCohen et al., 2026) is a unified audiovisual diffusion model that jointly generates synchronized video and audio. Its backbone is an asymmetric dual-stream Transformer, assigning substantially more capacity to the video stream than to the audio stream to reflect their differing information densities. The streams operate on separate video and audio VAE latents and exchange information through bidirectional audiovisual cross-attention, temporal positional encoding, and cross-modal adaptive normalization.

B.2 Training Recipes

All models are fine-tuned for one epoch at a resolution of 512×512. We adopt a unified fine-tuning strategy based on model scale and modality. For image generation and interleaved text-image generation models, models with fewer than 10B active parameters are fully fine-tuned, while larger models are adapted with LoRA. For all video generation models, we consistently use LoRA fine-tuning.

For image generation, Qwen-Image-Edit and FLUX.2-dev are adapted using rank-32 LoRA with a learning rate of 1×10^{-4}, whereas BAGEL is fully fine-tuned with a smaller learning rate of 5×10^{-6}. For Qwen-Image-Edit and FLUX.2-dev, the LoRA adapters are applied to the query, key, value, and output projections, as well as selected multimodal projection, feed-forward, and modulation layers. For interleaved text-image generation, ThinkMorph and SenseNova-U1 are fully fine-tuned using learning rates of 1×10^{-5} and 2×10^{-5}, respectively. For video generation, we apply rank-32 LoRA to all evaluated models, including Wan2.2-I2V-A14B, Wan2.2-TI2V-5B, Wan2.1-I2V-14B-720P, and LTX-2.3-I2AV. The adapters are applied to the query, key, value, and output projections, with the feed-forward projections additionally adapted for the Wan models. All video models are trained at 16 fps with a learning rate of 1×10^{-4}.

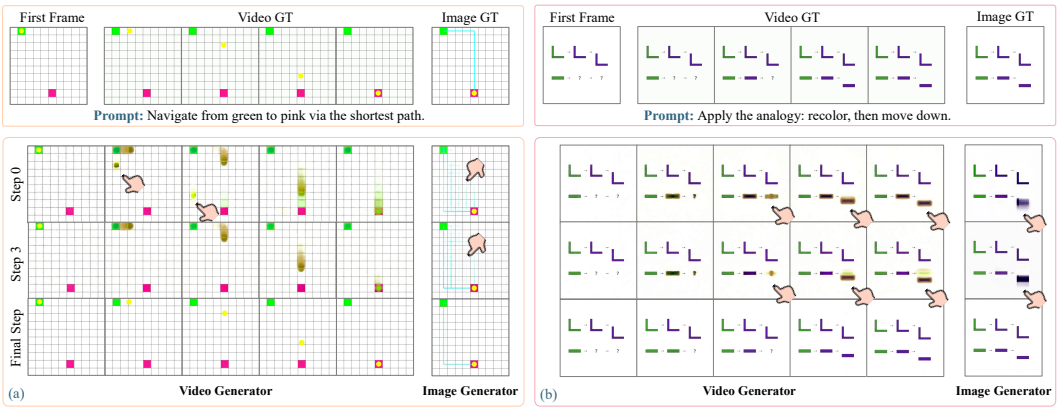

Figure 7: **Chain-of-Step visualization.** Both video and image generators exhibit Chain-of-Step behaviors (see pointed), such as *Multi-Path Exploration* (a) and *Superposition-Based Exploration* (b). Multiple solution hypotheses emerge in early diffusion steps and gradually converge to one.

Specifically, to isolate and assess the contribution of the VBVR-Pro-Dataset training data, VBVR-SenseNova-U1 and VBVR-Pro-SenseNova-U1 are initialized from an unreleased pretrained checkpoint that has never been exposed to VBVR data during pretraining. This setup allows us to more cleanly attribute the resulting performance gains to VBVR/VBVR-Pro fine-tuning.

B.3 Chain-of-Step: Visualization of Native Visual Reasoning

Following Wang et al. (2026c), who introduce tools for visualizing reasoning behaviors in video diffusion models, we apply the same analysis to both video and image generation models. Specifically, we decode noisy diffusion latents from different denoising steps to inspect the intermediate reasoning behaviors of models trained on VBVR-Pro-Dataset. We analyze two representative models: the strongest video generation model, VBVR-Pro-Wan2.2-I2V-A14B, and the interleaved text-image generation model, VBVR-Pro-SenseNova-U1. Across both modalities, we observe the *Chain-of-Step* patterns described by Wang et al. (2026c). As shown in Figure 7(a), both models explore multiple candidate paths across denoising steps before converging to the shortest feasible route, corresponding to *Multi-Path Exploration*. The video model represents this process through continuous object motion, whereas the interleaved text-image model directly draws the candidate paths in image space. In Figure 7(b), both models further exhibit *Superposition-Based Exploration*, where multiple plausible visual states are overlaid during intermediate denoising steps before the model converges to a final solution.

B.4 Counterfactual Diagnostics

Table 7: Counterfactual diagnostics on a *trained* **VBVR-Pro-SenseNova-U1** at inference time.

#	Intervention	Overall	ID Avg.	OOD Avg.
Experiment 1: Input Modalities				
1	Original input text and image	0.638	0.811	0.464
2	Input text rephrased, semantically unchanged	0.640	0.811	0.468
3	Input text with task semantics removed	0.317	0.435	0.199
4	Input image removed	0.064	0.077	0.051
5	Horizontally flipped or rotated by 180°	0.613	0.786	0.439
Experiment 2: Intermediate Reasoning States				
6	Intermediate text removed	0.533	0.661	0.406
7	Intermediate text contradicted	0.530	0.670	0.390
8	Intermediate image removed	0.099	0.128	0.069
9	Intermediate image corrupted with noise	0.190	0.307	0.074

The strong performance of trained models raises an important question: does task-specific training induce reusable visual reasoning, or does the model instead exploit superficial regularities in the training distribution, such as recurring instruction templates or memorized associations between prompts and outputs? We conduct two complementary counterfactual diagnostics to examine this question. The first perturbs the input text or image to test whether predictions depend on the underlying problem instance rather than its surface formulation. The second intervenes in intermediate reasoning states to determine which modality plays a more important role during the reasoning process. All experiments in this section are conducted on VBVR-Pro-SenseNova-U1.

B.4.1 INTERVENTION ON INPUT MODALITIES

As shown in Table 7, **Intervention #1** uses the original input text and image as the clean reference. **Intervention #2** semantically rephrases the input text while preserving the task requirements: the performance is essentially unchanged, indicating that successful reasoning does not depend on the particular linguistic form of the instruction. In contrast, **Intervention #3** removes the task-specific semantics from the input text, retaining only generic output requirements. This causes a substantial degradation, showing that the initial text still provides necessary task objectives rather than merely serving as a superficial trigger. **Intervention #4** removes the input image entirely while keeping the input text unchanged, which produces by far the largest degradation in this experiment, reducing performance to 0.064. The solution therefore cannot be recovered from the textual input alone. **Intervention #5** constructs geometric counterfactuals that preserve the underlying task but change its visual realization. Specifically, the input image is either horizontally flipped or rotated by $180°$, and direction-dependent language in the input text is transformed consistently with the corresponding geometric transformation. Performance remains substantial under these transformations. When the spatial arrangement changes, the generated reasoning trajectory changes accordingly and can still reach the correct final state.

Taken together, these results argue against instruction-template fitting or question memorization: performance is insensitive to a semantically equivalent reformulation of the input text, but strongly dependent on both the task semantics and, more importantly, the specific visual state on which the task must be executed. The gains from VBVR-Pro-Dataset training are therefore difficult to explain as memorized mappings from recurring question patterns to outputs; they are more consistent with learning reusable visual relations and operations.

B.4.2 INTERVENTION ON INTERMEDIATE REASONING STATES

While Section 4 studies which forms of supervision are most useful during training, here we examine which information is actually used during the interleaved reasoning trajectory of a trained model. Specifically, we intervene directly on intermediate outputs at inference time and measure their causal contribution to the subsequent rollout.

For the textual pathway, **Intervention #6** removes the intermediate text before the subsequent generation step, while **Intervention #7** replaces it with a contradictory version (e.g., changing "left" to "right"). Both interventions lead to similar, moderate degradation. This suggests that intermediate language provides useful context for the rollout, but the reasoning trajectory remains partially functional even when that information is removed or made inconsistent.

The visual pathway is considerably more sensitive. **Intervention #8** removes the intermediate image before it is fed back into the subsequent generation step, while **intervention #9** corrupts the intermediate image with strong noise. Both interventions cause substantially larger performance drops than the textual interventions, with complete removal producing the strongest degradation. Since the final output image itself is not directly modified, these effects arise specifically from disrupting the intermediate visual state available to subsequent reasoning steps.

The stronger sensitivity to intermediate visual interventions mirrors the training-time result in Section 4. Together, the two analyses provide complementary evidence: visual trajectories are not only a stronger source of supervision during training, but are also actively used during inference. Explicit textual reasoning can support the rollout, but the intermediate visual state appears to function as the primary working representation. This pattern is consistent with a reasoning process that maintains and updates visual states, with intermediate text playing a smaller role than intermediate images.

B.5 DETAILS OF TRANSFERABILITY WITHIN THE VIDEO MODALITY

B.5.1 DETAILED RESULTS ON EXTERNAL BENCHMARKS

We report fine-grained results on seven external video reasoning benchmarks covering a diverse range of video reasoning capabilities, including RISE-Video(Table 8), RULER-Bench(Table 9), VideoThinkBench(Table 10), BabyVision-Gen(Table 11), V-ReasonBench(Table 12), MME-CoF-Pro(Table 13), and IntelligentVBench(Table 14). Beyond overall benchmark performance, we provide category-level scores where available to enable more detailed comparisons across dimensions.

Table 8: Per-knowledge-type results on **RISE-Video**. WS: weighted score; Acc.: accuracy. **Bold**: best result; underline: second best.

Models	Commonsense		Subject		Perceptual		Societal		Logical		Experiential		Spatial		Temporal	
	WS	Acc.	WS	Acc.	WS	Acc.	WS	Acc.	WS	Acc.	WS	Acc.	WS	Acc.	WS	Acc.
Video Generation Models																
Open-source Models																
CogVideoX1.5-5B-I2V	53.54	1.20	53.01	1.28	57.33	2.27	47.69	0.00	33.38	2.38	40.80	1.32	34.50	2.00	47.86	6.56
LTX-2.3-I2AV	60.85	6.02	63.62	7.69	64.75	9.09	55.45	0.00	46.92	2.38	51.44	1.32	36.05	0.00	56.04	6.56
Wan2.2-I2V-A14B	**65.24**	**7.23**	64.31	3.85	70.66	11.36	69.43	3.03	46.52	4.76	64.20	6.58	54.05	10.00	64.15	6.56
VBVR-Wan2.2	61.58	4.82	66.64	7.69	**71.70**	6.82	68.94	**9.09**	65.24	**26.19**	66.32	7.89	**66.18**	**16.00**	**64.62**	**14.75**
Ours																
VBVR-Pro-Wan2.2-I2V-A14B	61.37	**7.23**	**67.90**	**11.54**	71.34	**18.18**	**71.40**	**9.09**	**73.12**	**26.19**	**68.90**	**9.21**	61.65	10.00	60.22	8.20

Table 9: Results on **RULER-Bench** (I2V setting). IF: Instruction Following; VC: Visual Consistency; VF: Visual Fidelity; RC: Rule Coherence.

Models	Science Rule					Game Rule					Humanity Rule					Vision Rule				Overall				
	IF	VC	VF	RC	Avg	IF	VC	VF	RC	Avg	IF	VC	VF	RC	Avg	VC	VF	RC	Avg	IF	VC	VF	RC	Avg
Video Generation Models																								
Proprietary Models																								
Veo-3.1	**65.05**	**83.18**	**91.37**	**50.97**	**72.64**	39.75	51.45	77.95	17.70	46.71	**79.90**	87.37	94.49	**61.23**	**80.75**	59.53	72.67	**48.94**	60.38	**61.57**	70.38	84.12	**44.71**	65.19
Open-source Models																								
CogVideoX1.5-5B-I2V	30.16	53.25	80.51	13.08	44.25	23.27	64.05	73.40	9.01	42.43	22.22	66.67	73.15	19.81	45.46	48.50	54.06	20.01	40.86	25.22	58.12	70.28	15.48	42.27
LTX-2.3-I2AV	44.20	70.09	80.06	18.10	53.11	22.80	59.23	73.58	11.90	41.88	38.89	66.67	83.33	40.19	57.27	56.84	58.97	26.14	47.32	35.30	63.21	73.99	24.08	49.14
Wan2.2-I2V-A14B	55.37	81.64	88.14	20.44	61.39	20.32	60.81	78.93	12.92	43.25	63.89	83.33	96.30	41.11	71.16	63.03	78.57	34.87	58.82	46.52	72.20	85.48	27.34	57.89
VBVR-Wan2.2	49.58	76.13	88.77	16.88	57.84	21.24	68.93	83.38	18.79	48.09	44.44	75.00	88.89	33.70	60.51	72.67	78.81	44.77	65.42	38.42	73.18	84.96	28.54	56.28
Ours																								
VBVR-Pro-Wan2.2-I2V-A14B	54.76	82.44	88.91	17.47	60.90	**51.12**	**84.39**	**89.08**	**46.89**	**67.87**	63.89	**91.67**	**98.61**	48.70	75.72	**73.08**	**80.98**	42.74	**65.60**	56.59	**82.89**	**89.40**	38.95	**66.96**

Table 10: Results on the vision-centric split of **VideoThinkBench**.

Models	Eyeballing			Visual Puzzles			ARC-AGI-2	Maze			Average
	Point	Line	Shape	Symmetry	Gradient	Compos.		Square	Hexagon	Circle	
Video Generation Models											
Proprietary Models											
Veo-3.1	34.44	24.29	30.00	77.50	40.00	**70.00**	**0.71**	0.00	0.00	0.00	27.69
Open-source Models											
CogVideoX1.5-5B-I2V	23.33	15.00	15.00	67.50	75.00	32.50	0.00	0.00	0.00	0.00	22.83
LTX-2.3-I2AV	23.33	17.50	35.00	55.00	90.00	55.00	0.00	5.88	0.00	0.00	28.17
Wan2.2-I2V-A14B	23.33	23.75	**37.50**	57.50	85.00	35.00	0.00	0.00	0.00	0.00	26.21
VBVR-Wan2.2	32.22	35.00	27.50	72.50	65.00	60.00	0.00	0.00	0.00	0.00	29.22
Ours											
VBVR-Pro-Wan2.2-I2V-A14B	**45.56**	**36.62**	20.00	**87.50**	**95.00**	67.50	0.00	**76.47**	**100.00**	0.00	**52.86**

Table 11: Results on **BabyVision-Gen**, reported over the four task types defined by the benchmark (accuracy, %).

Models	Fine-grained Discrimination	Visual Tracking	Spatial Perception	Visual Pattern Recognition	Overall
Video Generation Models					
Open-source Models					
CogVideoX1.5-5B-I2V	1.56	0.00	1.69	0.00	1.07
LTX-2.3-I2AV	2.34	0.00	1.69	5.26	2.14
Wan2.2-I2V-A14B	0.00	0.00	0.00	2.63	0.36
VBVR-Wan2.2	4.69	0.00	5.08	**10.53**	4.64
Ours					
VBVR-Pro-Wan2.2-I2V-A14B	**15.63**	0.00	**18.64**	**10.53**	**12.50**

Table 12: Per-task results on **V-ReasonBench** (pass@k, $k = 5$). Tasks are grouped by the four reasoning dimensions defined by the benchmark.

Models	Structured Problem-Solving				Spatial Cognition			Pattern-based Inference			Physical Dynamics			Avg.
	Arith.	Code	Sudoku	TicTacToe	ShapeFit	Symmetry	ColorConn.	SeqComp.	Analogy	RuleFollow	BlockSlide	Vessels	Temp.	
Video Generation Models														
Open-source Models														
CogVideoX1.5-5B-I2V	0.00	**2.27**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	**90.00**	7.87
LTX-2.3-I2AV	0.00	0.00	0.00	10.00	3.57	0.00	**10.00**	5.00	0.00	0.00	**10.00**	0.00	20.00	4.51
Wan2.2-I2V-A14B	0.00	0.00	2.00	0.00	10.71	0.00	0.00	0.00	0.00	0.00	**10.00**	30.00	80.00	10.21
VBVR-Wan2.2	0.00	2.22	0.00	0.00	60.71	16.67	0.00	5.00	10.00	0.00	0.00	90.00	50.00	18.05
Ours														
VBVR-Pro-Wan2.2-I2V-A14B	0.00	0.00	**76.00**	**26.67**	**88.89**	**43.33**	**10.00**	**10.00**	**70.00**	**12.00**	0.00	**100.00**	60.00	**38.22**

Table 13: Per-category results on **MME-CoF-Pro** (no-hint setting). RS: Reasoning Score; CS: Consistency Score; Avg: mean over the five generation-quality dimensions. Following the original paper, each model is reported under all three metrics.

Models	Metric	Overall	Visual Detail	Visual Trace	Real-world Spatial	3D Geometry	2D Geometry	Physics-based	Rotation	Table/Chart	Object Count.	GUI	Embodied	Medical	4D Dynamics	Natural Sci.	Text-based	Visual Logical
Video Generation Models																		
Proprietary Models																		
Veo-3.1	RS	**55.90**	**61.50**	45.50	**54.00**	43.90	29.20	**55.90**	60.20	49.40	**51.70**	**69.30**	**81.70**	41.70	**66.20**	49.60	**62.30**	**72.00**
	CS	47.50	**51.20**	47.80	57.80	27.20	28.00	30.50	44.40	31.10	47.20	**68.90**	**72.90**	22.80	68.20	57.30	**49.60**	55.50
	Avg	49.50	56.80	53.00	62.10	32.20	32.40	27.10	45.80	33.60	46.10	**69.70**	**76.10**	27.10	65.40	56.70	**52.50**	55.50
Open-source Models																		
CogVideoX1.5-5B-I2V	RS	28.46	11.11	26.77	30.26	18.70	13.33	29.24	35.95	46.37	11.83	8.52	43.68	37.57	43.07	46.94	20.94	19.75
	CS	30.07	38.33	15.29	35.26	20.00	26.00	25.79	18.75	26.47	24.44	24.44	40.34	30.00	40.56	46.11	33.91	24.50
	Avg	30.13	39.00	13.76	30.42	18.33	27.40	23.16	20.25	25.76	29.00	23.33	37.52	32.44	38.11	46.78	39.57	27.30
LTX-2.3-I2AV	RS	34.05	42.59	38.39	39.39	25.46	3.33	46.08	27.71	42.75	14.42	27.84	38.33	43.90	56.20	42.31	19.86	25.00
	CS	42.20	21.11	40.00	40.00	26.67	31.00	36.32	21.25	**63.53**	42.22	64.12	58.97	45.00	61.11	58.89	34.35	21.90
	Avg	44.26	22.33	41.56	43.47	29.33	46.60	35.89	21.00	**65.88**	43.56	61.53	60.41	43.75	60.89	61.67	40.09	25.62
Wan2.2-I2V-A14B	RS	24.76	15.93	38.07	22.11	13.24	13.33	30.99	23.20	14.91	13.03	7.96	29.48	38.23	34.85	46.02	32.83	13.56
	CS	26.91	29.44	27.78	40.53	17.22	9.00	27.00	12.50	12.22	26.11	23.89	32.07	28.33	47.78	43.89	29.57	12.27
	Avg	29.84	32.89	31.67	41.47	18.56	15.20	25.20	16.88	27.22	31.56	22.22	36.00	31.89	43.33	44.11	31.30	18.27
VBVR-Wan2.2	RS	34.13	29.26	42.08	33.07	22.13	15.83	39.90	39.29	32.04	31.03	15.97	25.06	43.70	53.97	41.85	25.51	52.14
	CS	36.57	41.67	36.67	44.74	23.33	22.00	38.50	19.38	26.67	38.33	32.78	35.52	36.11	57.78	53.89	31.74	38.57
	Avg	39.48	45.22	41.67	48.11	24.78	35.60	36.10	21.13	42.44	48.67	38.00	37.38	35.00	56.33	47.56	35.91	36.67
Ours																		
VBVR-Pro-Wan2.2-I2V-A14B	RS	48.39	36.57	**67.67**	40.96	**57.22**	**35.83**	47.02	**66.98**	**57.50**	27.20	30.93	45.40	**50.58**	60.13	**52.50**	31.16	68.33
	CS	**61.66**	48.24	**78.33**	**75.26**	**51.11**	**57.00**	**57.89**	**58.67**	60.56	**58.33**	62.22	54.48	**70.56**	**70.00**	**74.44**	48.70	**64.50**
	Avg	**62.20**	**57.88**	**77.56**	**70.32**	**57.67**	**55.60**	**53.58**	**55.47**	63.00	**60.89**	63.33	61.66	**65.67**	**67.00**	**69.89**	50.87	**63.90**

Table 14: Results on **IntelligentVBench** (Implicit I2V). IF: Instruction Following; CP: Condition Preserving; VQ: Visual Quality. AVG = (IF+CP+VQ)/3, MIN = min(IF, CP, VQ). AVG and MIN are computed for each instance before being averaged across instances. All scores are on a 1–5 scale.

Models	IF ↑	CP ↑	VQ ↑	MIN	AVG
Video Generation Models					
Proprietary Models					
Veo-3.1	4.12	3.58	3.52	3.15	3.74
Open-source Models					
CogVideoX1.5-5B-I2V	3.70	3.74	3.07	2.82	3.50
LTX-2.3-I2AV	4.15	3.92	3.58	3.13	3.88
Wan2.2-I2V-A14B	4.30	3.68	3.80	3.30	3.93
VBVR-Wan2.2	4.36	3.97	3.91	3.52	4.08
Ours					
VBVR-Pro-Wan2.2-I2V-A14B	**4.38**	**4.02**	**3.94**	**3.55**	**4.11**

B.5.2 Transferability Examples and Nearest-Neighbor Analysis

As shown in Table 4, the strongest model VBVR-Pro-Wan2.2-I2V-A14B exhibits strong transfer across substantially different external benchmarks and these benchmarks contain many real-world visual problems beyond the training distribution. For example, compared with its Wan2.2-I2V-A14B base model, it shows substantial improvements on shape fitting and analogy solving in V-ReasonBench, and embodied reasoning in MME-CoF-Pro.

To examine whether these improvements arise from memorizing training samples, or instead from generalizable reasoning acquired from conceptually transferable synthetic examples, we conduct a nearest-neighbor analysis. For the examples in Figure 8, we retrieve nearest training samples using CLIP (Radford et al., 2021) and DINOv2 (Oquab et al., 2024) image features and BGE-base-en-v1.5 (Xiao et al., 2024) text instruction embeddings. The retrieved examples are visually distinct from the test cases, even when their task instructions are related. This suggests that the model learns transferable visual concepts and operations, such as pattern finding, ordering, and object manipulation, rather than relying on surface-level memorization.

C Reinforcement Learning Details

C.1 CPS for Semantic Exploration

Group-relative optimization methods, such as Flow-GRPO (Liu et al., 2025) and DanceGRPO (Xue et al., 2025), formulate reverse denoising as a sequential decision process and optimize it using group-relative terminal rewards and a clipped policy objective. Their effectiveness relies on sampled trajectories differing in reward-relevant decisions.

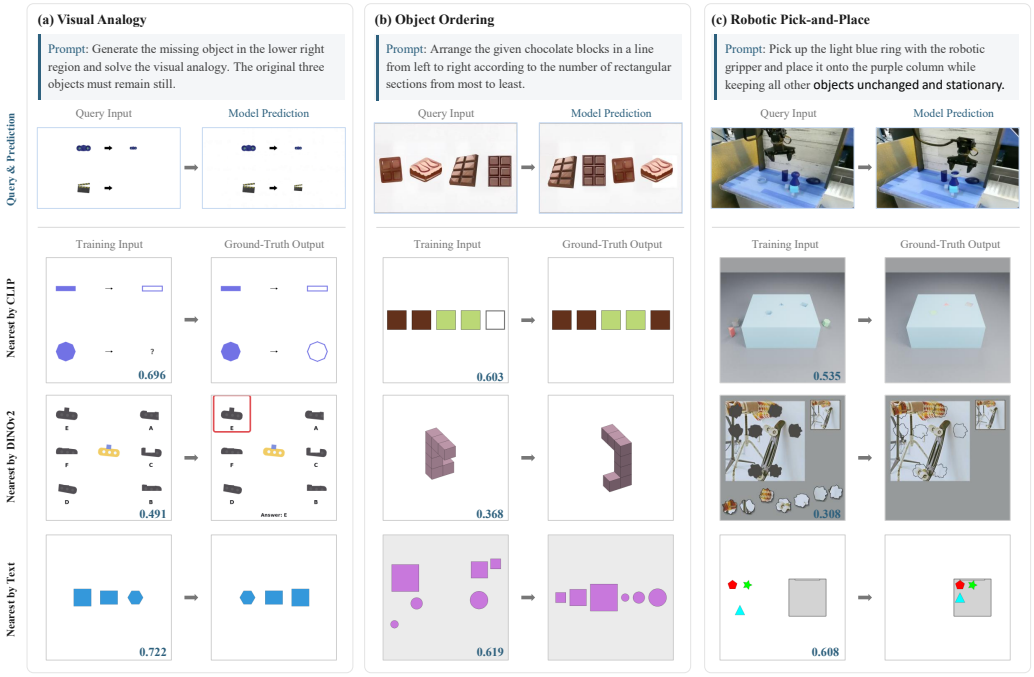

Figure 8: **Nearest-neighbor analysis against the training set.** For each query, we show the model prediction and the nearest training instances retrieved using CLIP, DINOv2, and text features. The retrieved instances are visually and semantically distinct from the queries, suggesting that the model's transferable capabilities within video reasoning cannot be attributed to direct memorization. Blue numbers indicate feature-space similarity.

Our noise hypothesis has two parts: perturbations should change task-level decisions enough to create reward variation, but preserve the objects and spatial relations needed to judge those decisions. Choosing a direction, selecting an answer, or connecting objects are examples of useful semantic variation. Perturbations that only change appearance yield weak relative-reward signals; as shown in Figure 6, low-stochasticity rollouts often converge to the same decision.

Naively increasing stochasticity, however, can degrade visual fidelity. In conventional Flow-SDE sampling, injected Gaussian noise is not exactly compensated by the scheduler-prescribed predicted-noise term, causing the effective noise magnitude to exceed its intended level. Accordingly, higher SDE stochasticity produces increasingly severe artifacts and distorted boundaries in Figure 6.

In contrast, Coefficients-Preserving Sampling (CPS) (Wang & Yu, 2025) provides a better exploration–fidelity trade-off. Stronger CPS noise induces visibly different task-level decisions and a wider range of rewards while largely preserving clean shapes and backgrounds. In Figure 6, CPS continues to produce visually clean rollouts even at high stochasticity, while supporting diverse semantic decisions.

Following Wang & Yu (2025), let $t \in [0, 1]$ denote flow time, with $t = 1$ corresponding to noise and $t = 0$ to the clean sample. Given the predicted velocity $v_\theta(x_t, t)$, the clean-sample and noise estimates are

$$\widehat{x}_0 = x_t - t\, v_\theta(x_t, t), \qquad \widehat{x}_1 = x_t + (1 - t)\, v_\theta(x_t, t). \tag{1}$$

For a denoising step from t to $t - \Delta t$, CPS updates the latent as

$$x_{t-\Delta t} = (1 - (t - \Delta t))\, \widehat{x}_0 + (t - \Delta t) \cos\left(\frac{\eta\pi}{2}\right) \widehat{x}_1 + (t - \Delta t) \sin\left(\frac{\eta\pi}{2}\right) \epsilon, \tag{2}$$

where $0 < \Delta t \leq t$, $\epsilon \sim \mathcal{N}(0, I)$ is freshly sampled Gaussian noise, and $\eta \in [0, 1]$ controls stochasticity. The squared noise coefficients sum to $(t-\Delta t)^2$ because $\cos^2(\eta\pi/2)+\sin^2(\eta\pi/2) = 1$, while the clean-sample coefficient remains $1 - (t - \Delta t)$. Thus, increasing η reallocates the noise contribution from the prediction to fresh Gaussian noise while preserving the scheduler's coefficient balance. At $\eta = 0$, Equation (2) reduces to the deterministic Euler update $x_{t-\Delta t} = x_t - \Delta t\, v_\theta(x_t, t)$.

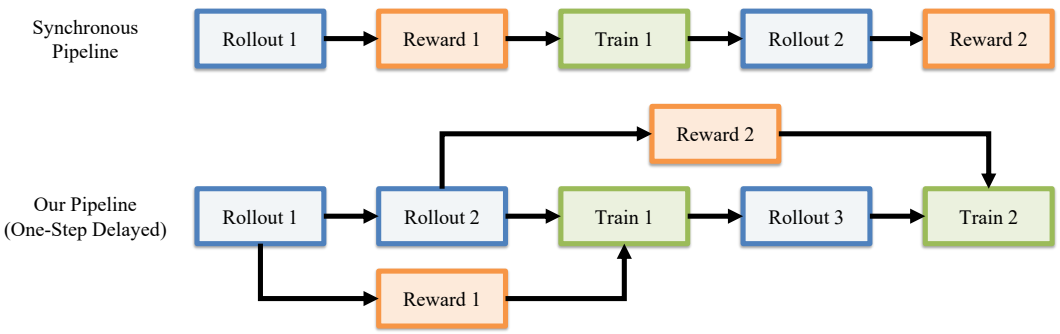

Figure 9: Comparison between synchronous training and our one-step-delayed pipeline. Our pipeline overlaps asynchronous preprocessing and reward evaluation with rollout generation and policy optimization, reducing GPU idle time while bounding policy lag to one rollout iteration.

C.2 Data and Training Recipes

All experiments in Section 5 start from VBVR-Pro-Wan2.2-TI2V-5B and use the same 50K RL subset of the VBVR-Pro Dataset, spanning 50 in-domain tasks. We refer to the original VBVR-Pro-Wan2.2-TI2V-5B checkpoint, before applying any of the training strategies in Section 5.1, as the baseline. This controlled setup isolates the effects of the training objective and reward signal.

For SFT, we fine-tune the model directly on the 50K training instances using a supervised objective, providing a non-RL baseline under the same data budget.

RLVLM and RLVR use the same RL algorithm, rollout configuration, optimization hyperparameters, training data, and model initialization; they differ only in the source of the reward signal. Therefore, their comparison directly measures the effect of replacing a general-purpose VLM judge with the proposed task-grounded verifiable reward.

C.2.1 Training Pipeline

To improve training efficiency, we adopt a one-step-delayed rollout pipeline that overlaps reward computation with GPU-side rollout generation and policy optimization. As illustrated in Figure 9, the synchronous pipeline requires policy optimization to wait for rollout preprocessing and reward evaluation to complete, resulting in substantial GPU idle time.

Inspired by DeepCoder (Luo et al., 2025a) and AReaL (Fu et al., 2025), our pipeline introduces a fixed one-step delay between rollout generation and optimization. After rollout k is generated, its preprocessing and reward computation are asynchronously dispatched to reward workers, while the GPUs continue generating rollout $k+1$. Once the rewards for rollout k become available, the GPUs perform policy optimization using rollout k while the reward workers process rollout $k+1$. This design removes reward computation from the critical execution path and improves GPU utilization.

Unlike fully asynchronous reinforcement learning systems, our approach preserves a deterministic execution order and bounds policy staleness to at most one rollout iteration. Therefore, it achieves efficient overlap between reward computation and policy execution while avoiding excessive policy lag.

To further reduce rollout overhead, we generate training samples at 512×512 resolution and only upscale them to 1024×1024 during reward computation. This strategy reduces the cost of trajectory generation while maintaining high-resolution inputs for reward evaluation.

How effectively this overlap can be exploited also depends on the systems cost of reward evaluation. The verifiable scorers used by RLVR run on CPU reward workers and can therefore overlap with GPU-side rollout generation and policy optimization without occupying accelerator memory. In contrast, RLVLM requires serving Qwen-3.6-27B as a GPU-resident reward model; in our implementation, we reserve 50% of the available GPU memory for judge inference. This difference is reflected in end-to-end training time: on 128 H800 GPUs, training for 2,200 steps takes approx-

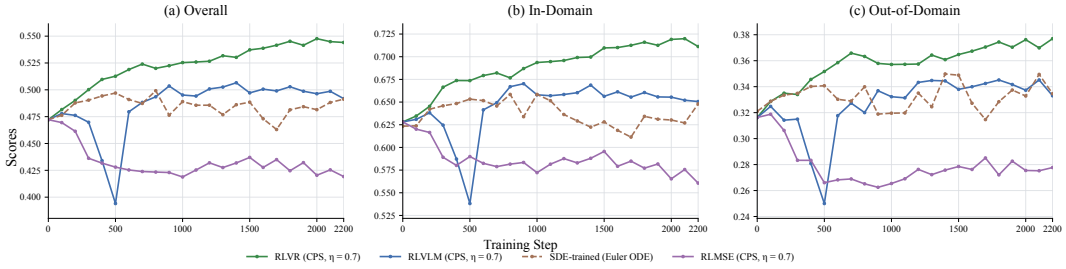

Figure 10: **Evaluation performance during RL.** Overall, in-domain, and out-of-domain scores are evaluated every 100 training steps. RLVR, RLVLM, and RLMSE use CPS inference with $\eta = 0.7$; the RLVR-SDE curve uses Euler ODE inference.

imately 5.1 days with the verifiable scorers, compared with 8.3 days with the VLM judge. RLVR therefore provides a $1.63\times$ increase in training throughput and reduces wall-clock time by 38.7%.

C.2.2 Training Optimizations

We investigate KL regularization with respect to a reference policy, which is commonly adopted in group-relative policy optimization. Maintaining an explicit reference model requires additional forward computation and memory consumption. Although lightweight alternatives such as Shadow KL avoid explicit reference-model evaluation, we find that KL regularization does not provide consistent gains in our setting. Therefore, all experiments are conducted without KL regularization.

For efficient large-scale training, we replace the native RMSNorm implementation with the fused Liger Kernel (Hsu et al., 2024), employ FlashAttention-3 (Shah et al., 2024) for memory-efficient attention computation, and enable `torch.compile` whenever compatible. We additionally use hybrid-sharded FSDP2 (Zhao et al., 2023b), which shards model states within each node while maintaining data-parallel synchronization across nodes.

C.2.3 Training and Inference Configuration

For CPS-based RL training, we use 30 sampling steps; RLVR-SDE instead uses Flow-SDE rollouts. For policy optimization, 60% of the denoising transitions are replayed, following the formulation described in Section C.1. Each optimization batch contains 32 prompts, with 32 trajectories sampled per prompt. We set the CPS stochasticity parameter to $\eta = 0.7$ and use a learning rate of 5×10^{-6}. Unless otherwise specified, all RL experiments use a CFG scale of 1.0.

For the controlled comparison in the main paper, both Verifiable-RL and VLM-Judge RL are evaluated using CPS with $\eta = 0.7$. This matched configuration isolates the effect of the reward objective from that of the inference sampler. All RL evaluations use 30 denoising steps, matching the rollout configuration used during training.

We additionally evaluate deterministic ODE sampling with Euler and UniPC in the sampler-sensitivity study. The CPS reference implementation (Wang & Yu, 2025) uses an Euler discretization as its deterministic ODE backbone, whereas the official Wan inference pipeline adopts UniPC by default. Euler is a first-order single-step solver, while UniPC (Zhao et al., 2023a) uses a higher-order predictor–corrector formulation. We refer to these configurations as ODE (Euler) and ODE (UniPC), respectively. Both use the same 30-step schedule and CFG scale as the RL evaluation, ensuring that the solver is the only inference variable being changed.

For the supervised fine-tuning (SFT) baseline, we retain the default inference configuration of the official Wan implementation: UniPC with 50 sampling steps and a CFG scale of 5. This configuration is kept unchanged from the original model inference pipeline.

C.3 Training Dynamics

Figure 10 compares RLVR, RLVLM, RLVR-SDE, and RLMSE throughout training. RLVR shows sustained improvements across overall, in-domain, and out-of-domain scores, whereas RLVLM

Table 15: Full comparison of inference samplers on **VBVR-Pro-Bench**. We evaluate CPS at different stochasticity levels and deterministic ODE sampling using either the Euler or UniPC solver. Baseline and SFT use the default UniPC inference configuration. Detailed inference settings are provided in Section C.2.3. We report the overall, In-Domain (ID), and Out-of-Domain (OOD) scores together with category-wise performance. Higher is better. **Bold**: best; underline: second best. Rankings are computed before rounding.

			In-Domain						Out-of-Domain					
Model	**Inference**	**Overall**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**
Baseline	UniPC	0.470	0.641	0.687	0.623	0.498	0.621	0.786	0.300	0.453	0.158	0.181	0.505	0.452
SFT	UniPC	0.502	0.677	**0.731**	0.629	0.545	0.662	0.840	0.327	0.514	0.163	0.190	0.555	**0.497**
RLVR	CPS ($\eta = 0.1$)	0.509	0.675	0.699	0.649	0.586	0.644	0.776	0.342	0.489	0.277	0.246	0.499	0.390
	CPS ($\eta = 0.3$)	0.526	0.698	0.723	0.676	0.618	0.680	0.783	0.354	0.502	0.289	0.254	0.471	0.439
	CPS ($\eta = 0.7$)	**0.544**	**0.711**	0.724	0.665	0.648	0.689	**0.857**	**0.377**	**0.574**	0.283	0.266	0.551	0.428
	CPS ($\eta = 0.9$)	0.539	0.706	0.709	**0.679**	**0.650**	**0.697**	0.799	0.372	0.501	0.273	**0.282**	**0.556**	0.382
	ODE (Euler)	0.522	0.697	0.718	0.666	0.596	0.688	0.806	0.347	0.499	0.266	0.263	0.495	0.387
	ODE (UniPC)	0.522	0.696	0.723	0.664	0.596	0.687	0.815	0.347	0.493	0.269	0.268	0.493	0.383
RLVLM	CPS ($\eta = 0.1$)	0.482	0.621	0.630	0.602	0.512	0.572	0.768	0.342	0.463	**0.302**	0.259	0.455	0.371
	CPS ($\eta = 0.3$)	0.493	0.650	0.656	0.644	0.539	0.643	0.750	0.335	0.462	0.283	0.235	0.485	0.393
	CPS ($\eta = 0.7$)	0.503	0.660	0.661	0.640	0.579	0.624	0.818	0.345	0.534	0.263	0.239	0.467	0.429
	CPS ($\eta = 0.9$)	0.509	0.665	0.668	0.654	0.573	0.673	0.773	0.352	0.469	0.269	0.266	0.481	0.385
	ODE (Euler)	0.488	0.643	0.666	0.618	0.529	0.626	0.735	0.333	0.476	0.286	0.245	0.474	0.384
	ODE (UniPC)	0.497	0.653	0.694	0.609	0.553	0.634	0.754	0.342	0.482	0.295	0.256	0.475	0.386

stagnates and briefly drops around Steps 400–500. RLVR-SDE also fluctuates, while RLMSE remains below its initial performance. The inference settings differ for RLVR-SDE, as specified in the caption. We examine sampler sensitivity in Section C.4 and analyze the temporary RLVLM drop in Section C.5.

C.4 Sampler Sensitivity

The main results use CPS with $\eta = 0.7$ for both RL models to provide a matched comparison. Table 15 extends this evaluation to CPS with $\eta \in \{0.1, 0.3, 0.7, 0.9\}$ and deterministic ODE sampling with either Euler or UniPC.

For Verifiable-RL, increasing CPS stochasticity at inference from $\eta = 0.1$ to $\eta = 0.7$ raises the overall score from 0.509 to 0.544, with corresponding gains on both the In-Domain and Out-of-Domain benchmarks. Performance decreases slightly at $\eta = 0.9$, so CPS with $\eta = 0.7$ provides the strongest aggregate result for this trained policy.

The two deterministic solvers produce nearly identical results for Verifiable-RL: ODE (Euler) and ODE (UniPC) both achieve an overall score of 0.522, with only minor category-level differences. Their close agreement indicates that the deterministic results are not sensitive to the particular ODE solver. Nevertheless, both remain below CPS with $\eta = 0.7$, showing that stochastic inference improves the measured performance of this policy.

VLM-Judge RL exhibits smaller and less consistent differences across samplers. Its strongest aggregate results are obtained with CPS at $\eta = 0.7$ or $\eta = 0.9$, while ODE (UniPC) performs slightly better than ODE (Euler). However, the gaps are substantially smaller than for Verifiable-RL. Moreover, Verifiable-RL remains stronger than VLM-Judge RL under every matched sampler, showing that the advantage of the verifiable reward is not specific to the inference configuration used in the main paper.

Figure 11 complements the aggregate results by evaluating each training checkpoint under different inference samplers. For Verifiable-RL, CPS with $\eta = 0.7$ generally yields higher measured scores than lower-stochasticity settings, while $\eta = 0.9$ provides no further improvement. The ODE (Euler) and ODE (UniPC) trajectories closely overlap and remain below the best CPS configurations across the three metrics. For VLM-Judge RL, the differences among samplers are smaller and less consistent. Higher-stochasticity CPS reaches a higher performance ceiling but also exhibits a pronounced temporary drop around Steps 400–500, whereas the deterministic ODE trajectories are smoother. These curves show sensitivity to inference noise; we examine the associated visual errors below.

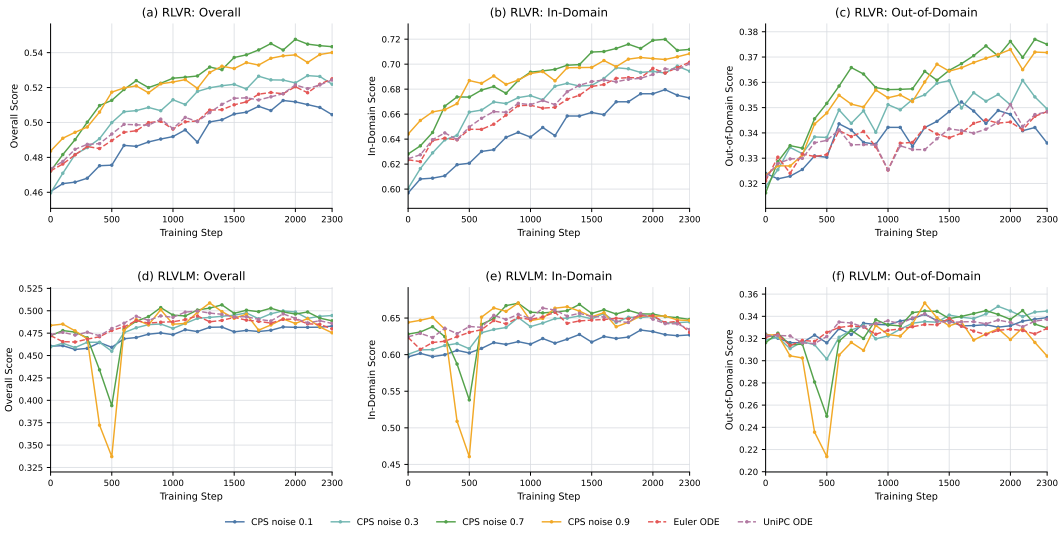

Figure 11: Evaluation performance throughout training under different inference samplers. The top and bottom rows show RLVR and RLVLM, respectively, while the columns report overall, in-domain, and out-of-domain scores. Each panel compares CPS with $\eta \in \{0.1, 0.3, 0.7, 0.9\}$, ODE (Euler), and ODE (UniPC). Checkpoints are evaluated every 100 training steps.

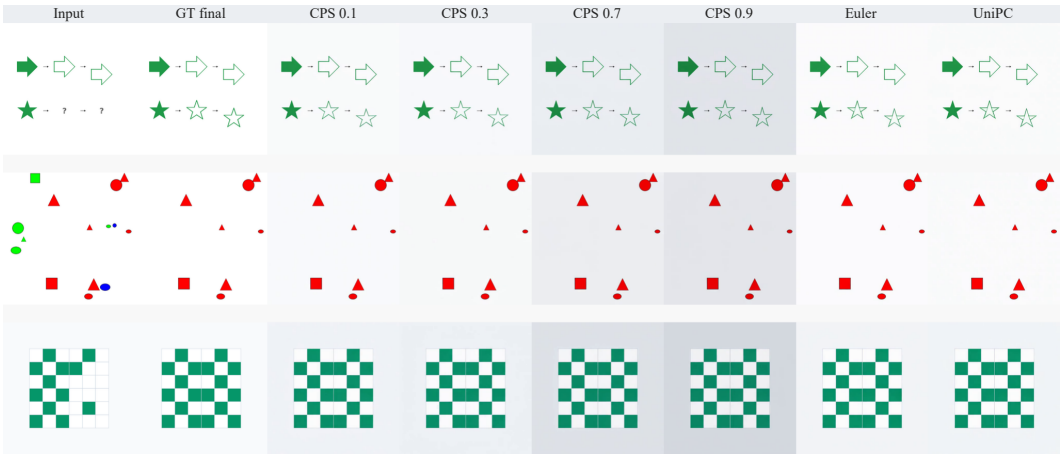

Figure 12: Sampler sensitivity of VLM-Judge RL at Step 500. Across three representative tasks, the foreground transformations remain largely correct, while the output background becomes increasingly gray as the CPS stochasticity increases. Low-stochasticity CPS and deterministic ODE sampling better preserve the target background.

C.5 Analysis of VLM-Judge Instability

As observed in Figures 10 and 11, VLM-Judge RL exhibits a temporary performance drop around Steps 400–500 under CPS with $\eta = 0.7$. The same behavior becomes more pronounced with $\eta = 0.9$. We investigate this instability by comparing inference samplers at Checkpoint 500 and tracing the corresponding failure mode across nearby training checkpoints.

Figure 12 compares inference samplers at the checkpoint exhibiting the largest performance drop. The foreground transformations remain largely correct across samplers, but the generated backgrounds become progressively grayer as the CPS stochasticity increases. In contrast, low-stochasticity CPS and deterministic ODE sampling largely preserve the target white background. The drop therefore reflects a sampler-dependent degradation in visual fidelity rather than a complete failure of the underlying reasoning capability.

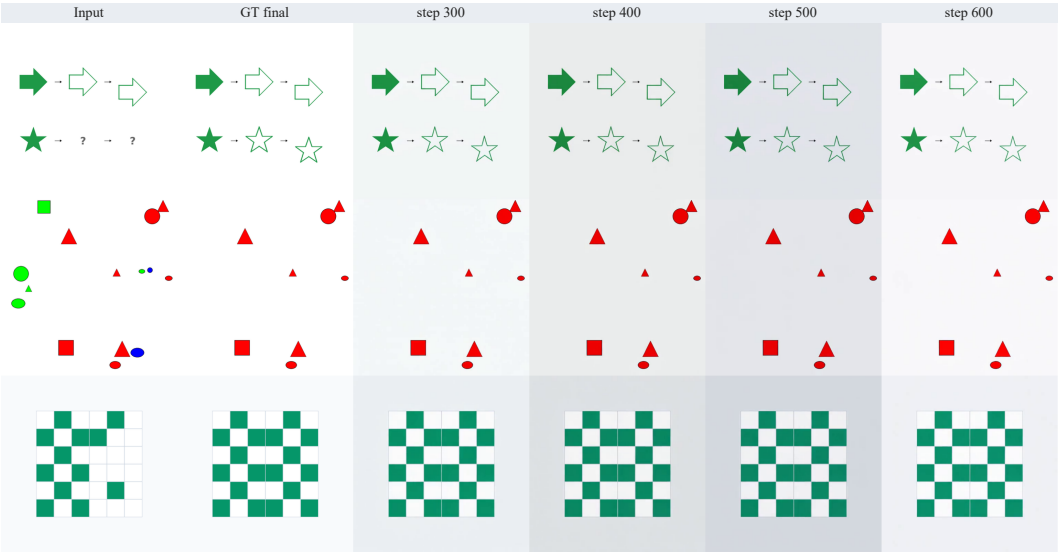

Figure 13: Evolution of the background-color failure under CPS with $\eta = 0.9$. The foreground task outcomes remain largely correct, while a gray background tint emerges at Step 400, becomes most pronounced at Step 500, and largely disappears by Step 600.

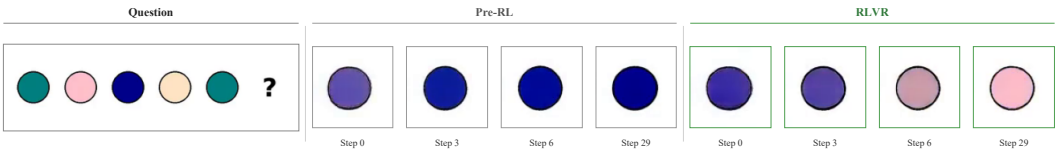

Figure 14: **Self-correction in a color-sequence completion task.** The question is shown on the left, followed by predictions from the pre-RL and RLVR models at Steps 0, 3, 6, and 29. For clarity, video frames are cropped to the predicted answer region. The pre-RL model commits to an incorrect blue answer early in denoising and retains it through the final step. In contrast, RLVR revises its initially incorrect purple prediction toward the target color at Step 6 and reaches the correct pink answer at Step 29.

Figure 13 further traces this failure mode across training checkpoints under CPS with $\eta = 0.9$. The background-color deviation begins to emerge at Step 400, is most severe at Step 500, and largely recovers by Step 600, closely matching the temporary drop in evaluation performance.

These observations suggest that the general-purpose VLM judge does not reliably penalize subtle background-color deviations. Such errors can therefore persist during optimization and become amplified under more stochastic sampling, even when the foreground task is completed correctly. In contrast, the task-grounded verifiable scorers explicitly capture these deviations and provides a more stable optimization signal. This explains why increasing stochasticity benefits VLM-Judge RL less consistently than Verifiable-RL, despite using the same underlying policy optimization procedure.

C.6 Case Studies

Aggregate benchmark scores show that RLVR improves final-task performance, but they do not reveal how RL changes the underlying visual reasoning trajectory. Following Wang et al. (Wang et al., 2026c), we visualize the model's instantaneous prediction of the final video at selected points along the denoising trajectory. We compare the SFT-initialized policy used as the starting point for RL, referred to as the pre-RL model, with the resulting RLVR model. The following examples illustrate two complementary changes in reasoning behavior: continued revision across denoising steps and more coherent execution of a multi-step spatial plan.

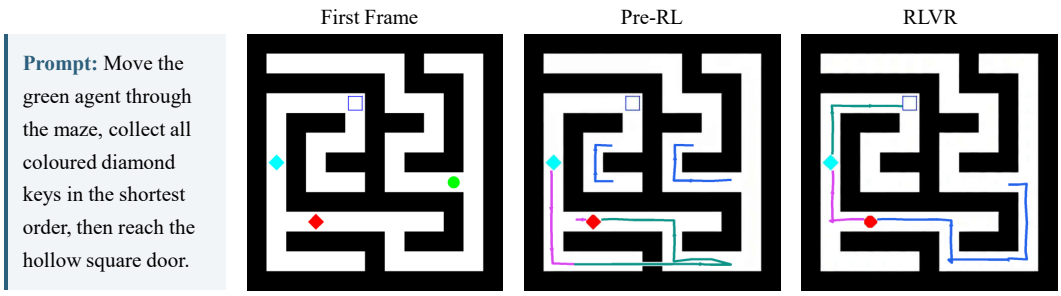

Figure 15: **More coherent maze navigation after RLVR.** Each panel summarizes all 81 frames of the corresponding generated video as a stage-wise directed trajectory. Blue, magenta, and teal denote the Start-to-Red, Red-to-Cyan, and Cyan-to-Goal stages, respectively, while arrowheads indicate motion direction. The pre-RL model produces disconnected and inconsistent movements that do not form a valid solution (score: 0.2249), whereas RLVR follows a coherent route through the maze and reaches the target (score: 0.8692). The trajectory-rendering procedure is described in Section C.6.1.

In the color-sequence task shown in Figure 14, the pre-RL model determines its answer within the first few denoising steps and does not reconsider it during the remainder of the trajectory. RLVR is also incorrect initially, but its prediction remains revisable: the predicted color changes at Step 6 and subsequently converges to the correct answer. Because both models use the same 30-step sampling schedule, the observed difference cannot be attributed to additional inference computation. Instead, RLVR continues to perform task-relevant refinement beyond the initial denoising steps, providing an example of later-step self-correction.

The maze example in Figure 15 demonstrates a different form of improvement. The pre-RL model places the agent at disconnected locations across frames and fails to maintain a valid route through the maze. In contrast, RLVR produces a more temporally consistent sequence of agent states, largely follows connected corridors, and eventually reaches the target. The improvement therefore concerns not only the final state but also the coherence of the intermediate trajectory, suggesting a stronger ability to maintain and execute a multi-step spatial plan.

Together, these examples illustrate two complementary effects of RLVR. First, it extends the effective reasoning horizon within a fixed denoising budget, allowing the model to revise decisions that would otherwise become fixed during the initial steps. Second, it improves the temporal consistency of state transitions across generated video frames. Additional qualitative examples are provided in Section C.6.2.

C.6.1 Directed Temporal Trajectory Visualization

To summarize the complete navigation videos without displaying a sparse subset of frames, we render each 81-frame video as a stage-wise directed centerline trajectory. Let I_t denote frame t, where $t \in \{0, \ldots, 80\}$. Because the scene layout and task markers remain static, we first estimate a clean background by taking the per-pixel temporal median. We then construct a soft foreground mask $M_t(p) \in [0, 1]$ using the known color of the moving agent—green in the maze task and orange in the waypoint task.

To recover a single agent location from each frame, we threshold the soft mask, retain its largest connected component $\mathcal{C}_t$, and use the component centroid:

$$B(p) = \operatorname*{median}_t I_t(p), \qquad \widehat{p}_t = \frac{1}{|\mathcal{C}_t|} \sum_{p \in \mathcal{C}_t} p. \tag{3}$$

Keeping only the largest component suppresses small compression artifacts and residual colors from the static task markers. We linearly interpolate isolated single-frame detection failures when the neighboring locations are sufficiently close. Longer disappearances and large inter-frame jumps remain disconnected, so failures such as abrupt relocation in the pre-RL video are not artificially repaired.

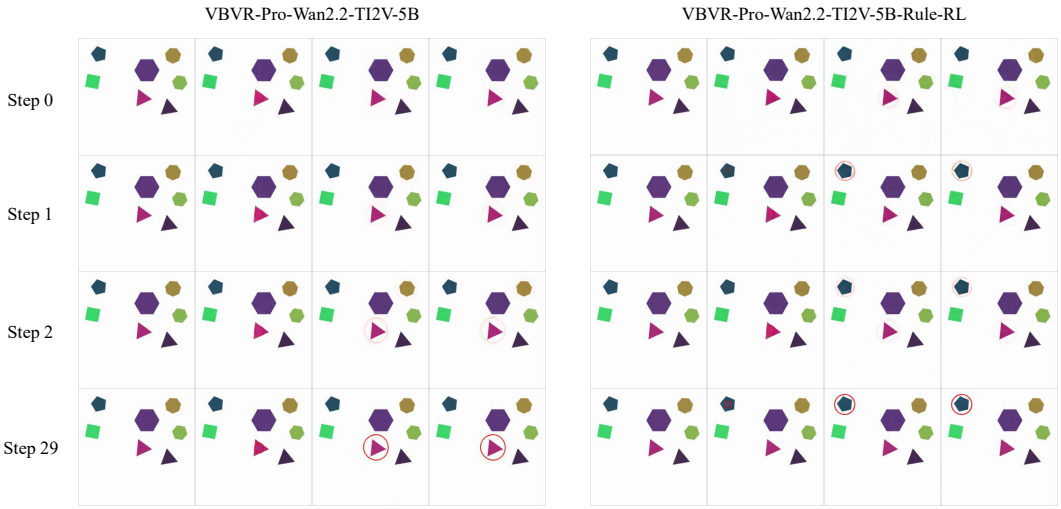

Figure 16: Revision of an object-selection hypothesis. The correct target is the blue pentagon. The pre-RL model commits to an incorrect triangle and retains this choice, whereas Verifiable-RL revises its initial prediction and ultimately selects the correct object. Rows show predictions at different denoising steps.

We divide each RLVR trajectory at the frames where the agent reaches the task-specific intermediate targets and apply the same temporal boundaries to the corresponding pre-RL video for direct comparison. For Figure 19, we divide the trajectories into four stages: Start-to-1, 1-to-2, 2-to-3, and 3-to-Goal. For Figure 15, we divide them into three stages: Start-to-Red, Red-to-Cyan, and Cyan-to-Goal. Each stage is assigned a distinct color, and arrowheads indicate the direction of motion. When two stages traverse the same spatial segment in opposite directions, their centerlines are slightly offset so that both passes remain visible.

Finally, we render the directed trajectories over the median background and restore the static task markers from a clean reference image. All 81 frames contribute to the recovered trajectories; no temporal subsampling is used. Compared with direct RGB averaging, which attenuates a moving object that occupies each location for only a few frames, the foreground-based centerline preserves its motion. Compared with a temporal foreground union, the stage-wise representation additionally exposes traversal direction, repeated visits, and discontinuities.

C.6.2 Additional Qualitative Examples

We provide additional examples to illustrate how Verifiable-RL changes the model's reasoning trajectory. The first three examples compare predictions at different denoising steps, while the final example examines how the generated behavior evolves across video frames. We use *pre-RL* to denote the SFT-initialized policy from which Verifiable-RL is trained.

Figure 16 shows that Verifiable-RL is more capable of revising an intermediate decision. Once the pre-RL model begins marking the triangle, its prediction remains unchanged. Verifiable-RL also considers the triangle initially, but subsequently redirects its prediction toward the correct pentagon. Since both models use the same 30-step sampling schedule, this difference reflects more effective use of the denoising trajectory rather than additional inference computation.

The examples in Figures 17 and 18 involve different tasks but exhibit a similar progression. In the set-intersection task, the model must identify the correct region. Verifiable-RL progressively reduces a broad prediction to the intended overlap, while the pre-RL model remains focused on irrelevant boundaries. In the object-matching task, the model must determine which objects should be connected. Verifiable-RL gradually replaces uncertain or inconsistent connections with the two correct pairings. Thus, one example concerns finding the correct region and the other concerns connecting the correct objects, but both show a more orderly transition toward a task-consistent solution after RL.

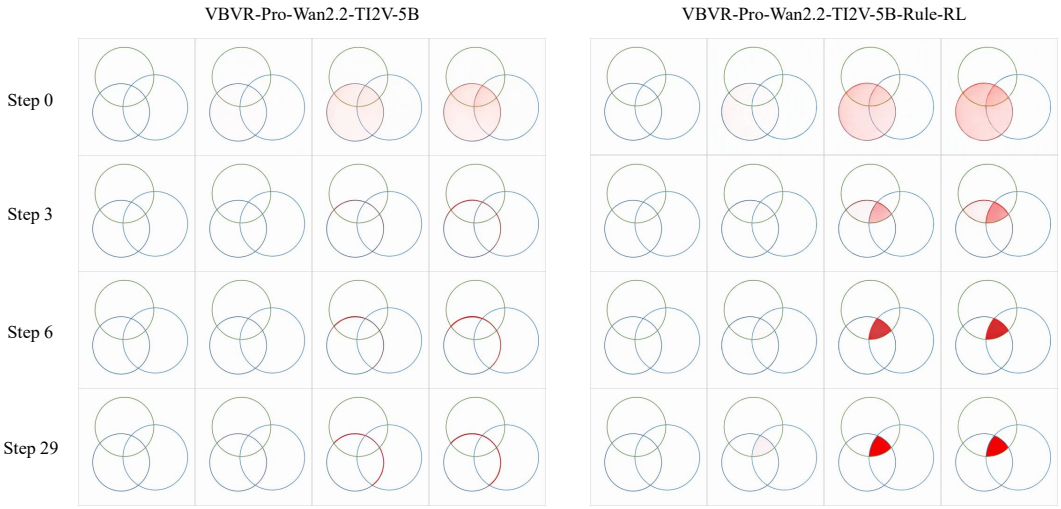

Figure 17: Progressive localization in a set-intersection task. The pre-RL model fails to isolate the target region and produces incomplete boundary traces. Verifiable-RL gradually narrows a broad initial prediction to the common intersection of the three circles.

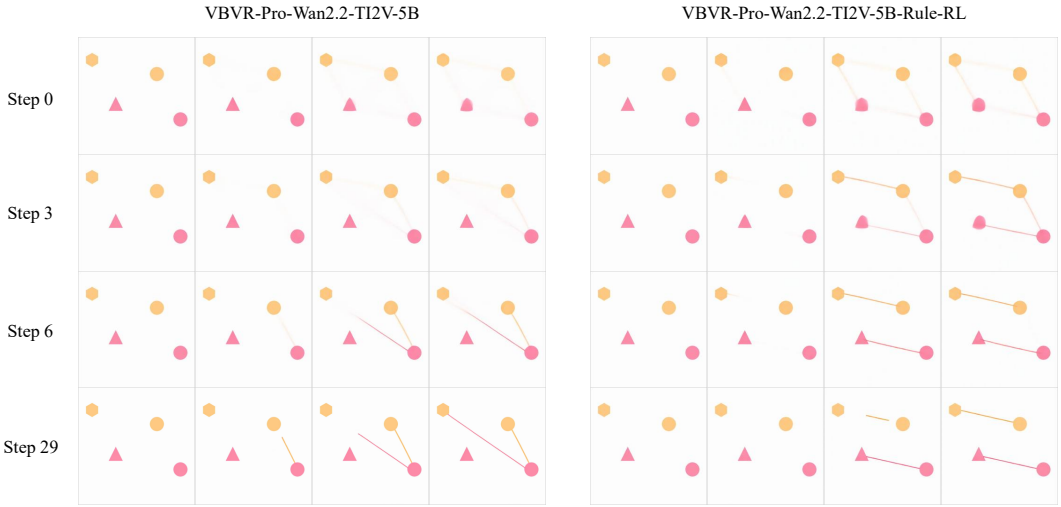

Figure 18: Progressive construction of object relations. The task requires connecting the two objects of each color. The pre-RL model produces inconsistent cross-color connections, whereas Verifiable-RL gradually forms the two correct same-color pairings.

The preceding examples compare predictions across denoising steps, whereas Figure 19 examines the temporal evolution of the final video. The pre-RL model places the agent at inconsistent locations and fails to maintain a coherent sequence of intermediate goals. Verifiable-RL produces more continuous state transitions and eventually reaches the destination. This suggests that RL improves not only how a solution is refined during denoising, but also how a multi-stage behavior is executed over time.

Overall, these examples show that Verifiable-RL improves two aspects of the visual reasoning trajectory: intermediate predictions become more effectively refined toward task-consistent outcomes, and the resulting behavior becomes more coherent across video frames.

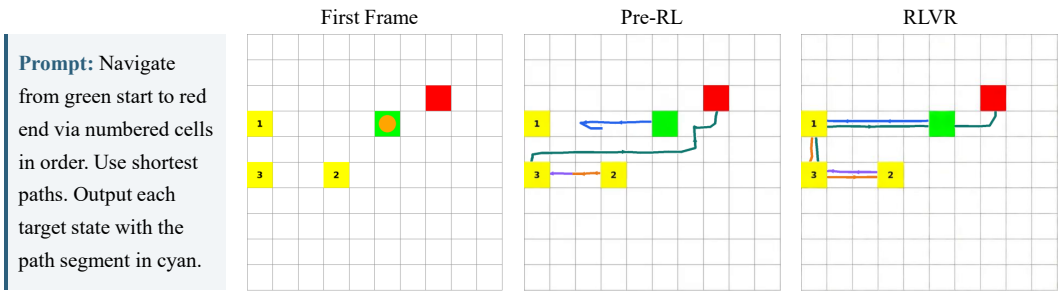

Figure 19: Stage-wise visualization of a waypoint-navigation task. The orange agent must start from the green square, visit Waypoints 1, 2, and 3 in order, and finally reach the red square. Each panel summarizes all 81 video frames as a directed centerline trajectory: blue, orange, purple, and teal denote the Start-to-1, 1-to-2, 2-to-3, and 3-to-Goal stages, respectively, while arrowheads indicate motion direction. Overlapping paths are slightly offset to reveal repeated traversal in opposite directions. The pre-RL model fails to complete the first stage and subsequently exhibits discontinuous relocation, whereas Verifiable-RL completes the required waypoint sequence and reaches the destination. The trajectory-rendering procedure is described in Section C.6.1.

D Additional Details and Analysis of the VBVR-Pro-Dataset

D.1 Details of Task Design and Implementation

Taxonomy. We adapt the cognitive framework of VBVR (Wang et al., 2026a), which organizes visual reasoning into five faculties: *perception*, *spatiality*, *transformation*, *abstraction*, and *knowledge*. *Perception* extracts information directly from visual inputs, covering OCR, symbol recognition, counting, sorting, and visual comparison. *Spatiality* reasons about locations and geometric relationships, including relative positions, 3D structures, and navigation. *Transformation* manipulates visual representations through operations such as 2D translation and rotation. *Abstraction* infers categories or rules from observations, including pattern induction, symmetry and shape completion, and reasoning under complex or newly specified rules. Finally, *knowledge* draws on intrinsic or acquired knowledge, such as conventional icon meanings, physical phenomena, common sense, rules and strategies of classic games, and the mechanics of everyday objects. Recognizing that a comprehensive task could demonstrate multiple capabilities, tasks in VBVR-Pro can carry several labels.

Design Principles. Each task is designed according to four principles. First, the core reasoning process should be vision-native: the solution should depend on visual state transformations, spatial relations, or dynamics rather than purely linguistic deduction. Second, task outcomes should be unambiguous and rigorously inspectable; prompts and visual inputs are iteratively refined through human inspection of model responses. Third, tasks should exhibit substantial visual diversity through randomized appearances, layouts, object attributes, scene configurations, and paraphrased textual prompts. Fourth, contributors are encouraged to design non-trivial tasks that require multi-step reasoning rather than single-step perception or pattern matching.

Generators and Metadata. VBVR-Pro is built from 300 task-specific generators under a unified framework: 150 are reimplemented and revised from VBVR (Wang et al., 2026a), and 150 are newly designed to broaden task coverage. Each generator is a parameterized program that samples a configuration from a structured parameter space, such as grid size, object count, layout, visual appearance, and difficulty level, and then instantiates a corresponding problem instance. A task-specific solver computes the correct solution, providing programmatically derived supervision without manual annotation. During generation, each solved instance is accompanied by a metadata file containing its random seed, problem specification, complete solution, and key element attributes. This information provides structured ground truth for the verifiable reward scorers introduced in Section 3. It is also used for de-duplication and split construction, ensuring a clear separation between training and evaluation instances.

Curation. Since configurations are sampled procedurally, each generator can produce a large number of diverse instances. To ensure reliability, we review all 300 generators by inspecting rendered samples and verifying their solver implementations. We also check the resolution, frame rate, and frame count of every task before release. For the interleaved setting, we additionally use Gemini-3.1-Pro-Preview (Google DeepMind, 2026) to generate textual descriptions of intermediate solution steps. We design a dedicated prompt for each task and apply it uniformly to all instances from that task. To assess annotation quality, we manually inspect five instances per task, checking the correctness of the generated intermediate-step descriptions. Most descriptions examined in this manual audit are judged to be correct. The annotation procedure is described in Section D.3.

D.2 Additional Examples of the Multi-modality Design

Every instance in VBVR-Pro is released in two modalities, video and image, which are two renderings of the same solved instance rather than two independently collected corpora. This appendix describes how the image modality is derived from a solved instance and provides representative examples of the three output regimes introduced in Section 2. Figure 20 shows multiple tasks from each regime.

Choice of output regime. The regime is a property of the task generator, so all instances of a task share the same regime. It is determined by how much of the solution trajectory has to be made explicit in order to specify the answer. *Last-Frame* is used when the solution is fully determined by its final state and no intermediate state carries information the model must produce; the task then has exactly one output image, as in completing a Sudoku grid or placing the winning stone in Gomoku. *Key-Frame* is used when the solution consists of semantically meaningful discrete steps whose intermediate states form part of the answer, such as one board state per move in a sliding-block puzzle. The number of output images is therefore generally instance-dependent, following the number of steps in that instance's solution. Path-based tasks, such as mazes, are treated as a special case: the solution path is overlaid on a selected frame, allowing the trajectory to be represented on the image. *Multi-Frame* is used for continuous processes that have no natural discrete decomposition, such as rotating and translating an object to a target pose. In this regime, intermediate states are sampled at uniform progress along a task-specific process parameter.

Image prompts. The video and image modalities share the same task description. The image prompt appends an explicit output specification stating how many images to produce and what each should depict, as shown in bold in Figure 20. The video prompt omits this specification because the video represents the process continuously. This keeps the task conditions identical across modalities while making the discretisation used by the image modality explicit to the model.

Distribution over tasks. Of the 300 tasks in VBVR-Pro, 209 (69.7%) use the Last-Frame regime, 58 (19.3%) use Key-Frame, and 33 (11.0%) use Multi-Frame. Last-Frame accounts for approximately seventy percent of the tasks because, for many reasoning problems, the final state alone constitutes a complete answer. The remaining thirty percent require intermediate solution states to be represented explicitly, and thus provide a controlled setting for studying a broader question: whether dynamic video or discretized image sequences offer a more effective representation for visual reasoning.

D.3 Interleaved Data Generation

The interleaved representation pairs each output image with a textual reasoning step placed immediately before it. The images are taken directly from the image modality described in Section D.2. An instance containing K output images therefore produces K reasoning-image pairs.

Reasoning text generation. For each instance, Gemini-3.1-Pro-Preview receives the task question, input image, and complete sequence of solver-generated output images, and produces one reasoning step per image. The complete trajectory is provided because many tasks are difficult to solve reliably from the input alone. Gemini therefore annotates the known state transitions rather than independently discovering the solution. Each step is phrased prospectively and placed before its corresponding image.

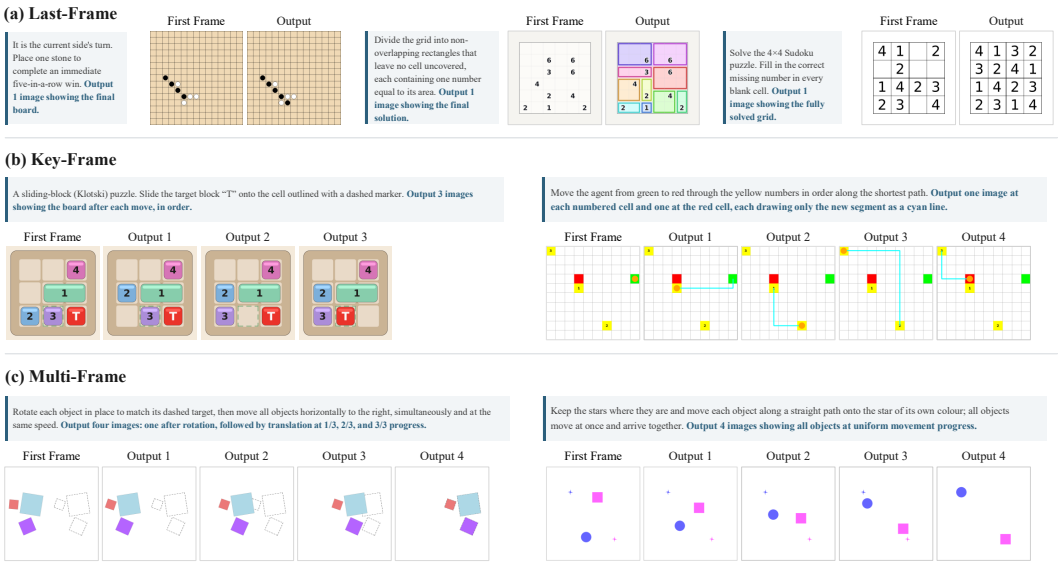

Figure 20: Representative examples illustrating the three output regimes of the VBVR-Pro image modality.

Task-specific prompts. We manually design one prompt for each task, specifying its rules, image order, required reasoning, and output format. We then inspect five generated annotations from each task to verify that they are consistent with the solver-generated transitions. In particular, we check that each step explains the decision leading to its corresponding output image rather than merely captioning the image. As a representative example, we show the annotation prompt for the Klotski sliding-block task, shown as the leftmost example in the Key-Frame row of Figure 20.

Example annotation prompt: Klotski sliding-block task

You are given the complete sequence of images for a Klotski sliding-block task, including the initial tray and the tray after each successive slide. Use the complete sequence as the ground-truth solution trajectory for generating the reasoning steps.

Scene. A tray is packed with rectangular wooden blocks and a small amount of empty space. Blocks may slide horizontally or vertically into the open space, but cannot be lifted, rotated, or overlapped. The goal is to move the highlighted target block to its designated exit. Each output image shows the tray after one slide.

Image order.

- Input image: the initial layout with the highlighted target block.
- Output images 1 to K: the layout after each successive slide.

For each output image, write 2 to 3 sentences of reasoning that can be placed before that image:

1. Identify the next block that should slide into the empty space and explain how this move contributes to the solution.
2. State the layout that this move should produce, with all other blocks unchanged.
3. Phrase the step prospectively as what should happen next, rather than as a caption describing an image that has already been produced.

Output format.

Step 1: <reasoning before output image 1>
...
Step K: <reasoning before output image K>

Table 16: **Comparison of VBVR-Pro with existing visual reasoning benchmarks and datasets.** *VLM* denotes VLM-only judging, *Mixed* combines VLM judgment with limited programmatic checks, and *Verifiable* uses verifiable scorers for all benchmark tasks. VBVR-Pro offers the broadest task coverage and largest scale, supports multiple modalities, and enables fully verifiable evaluation.

Dataset	#Tasks	#Images	#Videos	Split			Modality			Eval
				#SFT	#RL	#Test	Image	Interleaved	Video	
VideoThinkBench (Tong et al., 2026)	48	6,845	0	0	0	4,149	–	–	✓	Mixed
V-ReasonBench (Luo et al., 2025b)	13	652	0	0	0	326	–	–	✓	Mixed
Ruler-Bench (He et al., 2025)	40	493	0	0	0	622	–	–	✓	VLM
RISE-Video (Liu et al., 2026c)	8	477	0	0	0	467	–	–	✓	VLM
MME-CoF-Pro (Qi et al., 2026)	16	370	0	0	0	303	–	–	✓	VLM
IntelligentVBench (Pan et al., 2026)	4	1,694	210	0	0	1,030	–	–	✓	VLM
VGI-Bench (He et al., 2026a)	27	–	0	0	0	810	–	–	✓	VLM
RISEBench (Zhao et al., 2025)	16	415	0	0	0	360	✓	–	–	VLM
KRIS-Bench (Wu et al., 2025b)	22	1,670	0	0	0	1,267	✓	–	–	VLM
BabyVision-Gen (Chen et al., 2026b)	21	560	0	0	0	280	✓	–	–	VLM
GRADE (Liu et al., 2026b)	10	1,040	0	0	0	520	✓	–	–	VLM
OpenING (Zhou et al., 2025)	56	17,603	0	3,240	0	2,160	–	✓	–	VLM
Zebra-CoT (Li et al., 2026)	18	921,039	0	182,384	0	0	–	✓	–	–
StructCoT (Wang et al., 2026b)	44	–	0	840,000	0	5,600	–	✓	–	Mixed
RealUnify (Shi et al., 2026)	32	587	0	0	0	1,000	–	✓	–	VLM
Uni-MMMU (Zou et al., 2026)	8	3,544	0	0	0	885	–	✓	–	Mixed
ThinkMorph (Gu et al., 2026)	4	50,780	0	24,990	0	400	–	✓	–	Mixed
MIRA (Zhou et al., 2026)	20	1,482	0	0	0	546	–	✓	–	Mixed
VBVR (Wang et al., 2026a)	150	2,015,000	1,007,500	1,000,000	0	500	–	–	✓	Verifiable
VBVR-Pro	**300**	**3,471,558**	**1,300,500**	**1,250,000**	**50,000**	**500**	✓	✓	✓	**Verifiable**

D.4 Comparison with existing visual reasoning resources

Table 16 provides a full comparison of existing visual reasoning resources in terms of task coverage, dataset scale, splits, modalities, and evaluation protocols. VBVR-Pro is the largest dataset in the comparison and provides dedicated SFT, RL, and benchmark splits. It is also the only resource that supports image, video, and interleaved modalities within a unified task distribution. Like VBVR, VBVR-Pro uses verifiable scorers for all benchmark tasks, whereas the other resources with evaluation sets rely wholly or partly on VLM judgments.

D.5 Visual Complexity Measurements

Section 2 reports that the 150 tasks newly designed for VBVR-Pro are visually richer than the 150 tasks reworked from VBVR. Here, we provide the measurements supporting this claim. Because there is no single accepted definition of image complexity, we consider four complementary measures. Our conclusion is based on their consistent ordering of the two groups rather than on any individual measure.

Protocol. All measurements are computed on the question frame (`first_frame.png`), which is provided to the model as input. For each of the 300 tasks, we select 50 instances at evenly spaced positions in the instance list. We average each measure across these instances to obtain one task-level value, yielding 150 values for the reworked tasks and 150 for the newly designed tasks. We then compare the distributions of these task-level values. This treats the task, rather than the individual instance, as the unit of analysis and avoids regarding multiple instances generated by the same task as independent observations.

Measures. We use four complementary measures, each capturing a different aspect of visual complexity:

- **Distinct regions.** We quantise each frame to a 16-colour palette using median-cut quantisation without dithering. For each palette colour, we identify 8-connected components and retain those occupying at least 10^{-4} of the frame area. We remove the largest connected component, which corresponds to the background in our renderings, and count the remaining components.
- **Distinct colours.** We quantise each RGB channel to five bits and count the resulting colours that occupy at least 10^{-4} of the frame pixels.

- **Spatial information (SI).** We compute the standard deviation over pixels of the Sobel gradient magnitude of the luminance plane, corresponding to the frame-level SI measure defined in ITU-T P.910 (ITU-T, 2023).
- **Encoded size.** We re-encode each frame as JPEG at quality 90 and report the resulting file size in kilobytes. The same encoder and encoding settings are used for all frames.

For the first two measures, the minimum area thresholds are normalised by frame size, and the measures are computed at the native rendering resolution. Because SI and encoded size are sensitive to image resolution, we compute both after resizing each frame to 512×512.

Results. Table 17 reports the results. The newly designed tasks have higher means and medians under all four measures. Based on the median task-level values, they contain $6.9\times$ as many distinct regions and $2.4\times$ as many distinct colours as the reworked tasks, while exhibiting $1.7\times$ higher spatial information and a $2.0\times$ larger encoded size. The consistent ordering across these complementary measures supports the conclusion that the newly designed tasks are visually richer than the reworked tasks.

Table 17: Visual complexity of the question frames. For each task, each measure is first averaged over 50 instances; the table reports statistics over 150 task-level values per group. The ratio divides the median for the newly designed tasks by that for the reworked tasks.

	Reworked		Newly designed		
Measure	Mean	Median	Mean	Median	Ratio
Distinct regions	25.6	11.6	171.4	80.4	6.9×
Distinct colours	6.7	5.3	29.3	12.7	2.4×
Spatial information	79.8	68.8	134.3	119.0	1.7×
Encoded size (KB)	14.4	12.1	28.4	24.7	2.0×

D.6 Reasoning Depth: Blinded Pairwise Judgement

Section 2 reports that the new half of VBVR-Pro requires deeper reasoning than the reworked half. Unlike visual complexity, reasoning depth cannot be measured directly from a rendered frame. We therefore assess it through a blinded comparison from human annotation. We additionally conduct a binary per-task assessment that distinguishes direct single-rule application from multi-step reasoning.

Protocol. Each of the 150 reworked tasks is paired with one of the 150 new tasks using a fixed random one-to-one matching. Annotators receive only the starting frame and instruction for each task; task origins and group labels are withheld. Each pair is evaluated twice, with the two tasks exchanged between the first and second slots. This controls for positional preference and allows us to measure its magnitude.

For the pairwise assessment, annotator is asked to select the task that requires more reasoning before its answer can be determined. A pair is counted as consistent only when the same task is selected under both orderings. In a separate per-task assessment, annotator classifies each task according to whether its answer follows from a direct application of a single rule to the starting frame or instead requires multiple dependent inference steps, such as comparing alternatives, searching over possibilities, or eliminating candidate answers.

Results. The first slot was selected in 48.8% of pairwise judgements with a decisive outcome, indicating little systematic position bias. The two orderings selected the same task for 131 of the 150 pairs (87.3%). Among all 150 pairs, the new task was selected as requiring more reasoning under both orderings in 114 cases (76.0%), whereas the reworked task was selected under both orderings in 17 cases (11.3%); the remaining 19 pairs (12.7%) produced inconsistent outcomes. In the per-task assessment, 13 of the 150 reworked tasks (8.7%) were classified as requiring multi-step reasoning, compared with 68 of the 150 new tasks (45.3%). These results show that the new tasks are more likely to require both greater relative reasoning depth and multiple dependent inference steps than the reworked tasks.

E Details of Verifiable Scorers

E.1 Reproducibility of Scorers and VLM Judges

We re-evaluate the same videos with each evaluator, setting the VLM judges' temperature to 0. Table 18 reports the percentage of samples whose scores change across runs and the largest observed score change on the 0–1 scale. The VLM judges assign different scores to 55–93% of samples, with a maximum score change of up to 0.221. In contrast, our deterministic scorers return identical scores across runs. This consistency allows our verifiable scorers to assign the same reward to a given video across evaluations, supporting its use in large-scale RL training.

Table 18: **Reproducibility of evaluators.** We re-run each evaluator on the same videos. Our scorers are deterministic, while VLM judges often produce different scores across runs.

	Scorers	GLM-4.6V-Flash (GLM-V Team et al., 2025)	Gemma4-31B (Gemma Team et al., 2026)	Qwen3.6-27B (Qwen Team, 2026a)	Qwen3.7-plus (Qwen Team, 2026b)	InternVL3.5-38B (Wang et al., 2025)	Gemini-3.1-Pro (Google DeepMind, 2026)
Cases with Changed Score (%) ↓	**0.0**	54.6	69.5	74.5	80.8	82.9	92.8
Max. Score Change (0–1) ↓	**0.000**	0.125	0.110	0.117	0.091	0.158	0.221

E.2 Human Preference Alignment

Annotation Protocol We evaluate eight video generation models on VBVR-Pro-Bench. To keep annotation tractable, we sample one instance per task and form all $\binom{8}{2} = 28$ model pairs per task. Each pair is shown to ten independent annotators, who rate the absolute quality of both videos on a four-level scale: L_0 denotes a fully correct and faithful result, L_1 minor errors that do not affect task success, L_2 major errors while still attempting the task, and L_3 a complete failure in which the task is not accomplished. Annotators judge each video independently, so a pair receives ten level ratings per video.

From Ratings to Pairwise Preferences For each annotation vote, we convert the two level ratings into one of four outcomes: *A wins*, *B wins*, a *tie above the failure level*, or a *tie at the failure level*. If the two videos receive different levels, the one with the better level wins. If they receive the same level, the vote is a tie-above-failure when both are rated L_0–L_2, and a tie-at-failure when both are rated L_3. Aggregating the ten votes for a pair yields a human preference vector $\mathbf{v} = [\, n_A, n_{\text{tie-above}}, n_{\text{tie-fail}}, n_B \,]$ with $\sum_i v_i = 10$. Splitting ties is important because the tasks are difficult and most generated videos fall at L_3; collapsing all ties into one class would conflate pairs in which both videos fail with pairs in which both are of high quality.

Mapping Evaluator Scores to Preferences Each evaluator (our scorers or a VLM judge) produces a scalar quality score in $[0, 1]$ per video. For a pair with scores s_1, s_2, if $|s_1 - s_2| \le \theta$ (with $\theta = 0.05$) we declare a tie and assign it to tie-above- or tie-at-failure according to whether the mean $(s_1 + s_2)/2$ exceeds 0.5, matching the L_2–L_3 boundary of the human protocol; otherwise the higher-scoring video wins. The same mapping is applied to every evaluator, ensuring a consistent comparison protocol.

Agreement Metric We score an evaluator by how well its single predicted outcome agrees with the human vote vector of each pair. Let c be the outcome predicted by the evaluator; the per-pair agreement is $v_c/10$, i.e., the fraction of the ten annotators whose vote matches the prediction. We report the mean of this quantity over all evaluated pairs ("per-vote agreement").

Cost Estimation For proprietary APIs, cost per evaluation is computed from token usage and the provider's price. For open-source judges, it is the total wall-clock evaluation time on a self-hosted Lambda H100 SXM node multiplied by its rental price. Our scorers are lightweight rule-based computation requiring no API calls and only minimal compute (optionally a GPU for OCR) is required. However, for a conservative estimate, we charge its entire wall-clock runtime at the same H100 node rental rate used for the open-source judges. Its reported cost therefore represents an upper bound.

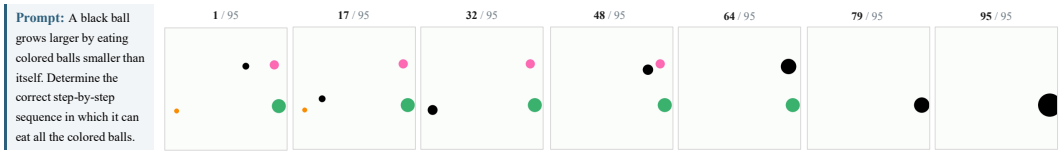

Figure 21: Greedy Ball Eating (Task O-31): prompt and ground-truth frames. The black ball absorbs the colored balls one at a time in ascending size order until only the grown black ball remains.

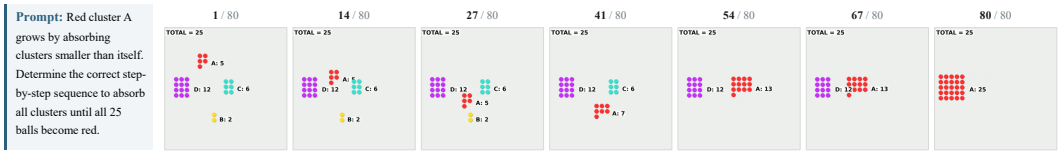

Figure 22: Ball-Cluster Merging (Task O-29): prompt and ground-truth frames. A survivor cluster sequentially absorbs the other clusters until only it remains, and its label must show the total count.

E.3 Detailed Scoring Logic for Selected Tasks

For each selected task, we present its prompt and ground-truth key frames, followed by the corresponding scoring logic. These examples illustrate how the task-specific scorers jointly assess the correctness of the final state, the validity of the intermediate process, and the preservation of task-irrelevant scene content.

Greedy Ball Eating (Task O-31) As shown in Figure 21, a black ball must absorb the smaller colored balls one at a time in ascending size order until only the grown black ball remains. From the first frame, we detect the black ball and each colored target i via HSV segmentation and contour analysis, recording its area A_i. Across frames, we track a presence ratio $\rho_i(t) \in [0, 1]$ and register an *eat* event for target i at the frame where ρ_i drops from $\approx$1 to $\approx$0 while the target overlaps the dilated black-ball mask. The final score equally weights a final-state term and an eating-process term:

$$S = 0.5 \cdot S_{\text{final}} + 0.5 \cdot S_{\text{eat}}. \tag{4}$$

Final-state score. We first compute a size-match score $b \in [0, 1]$ that measures how closely the final black ball's area matches its expected grown size, i.e., the total area of all absorbed balls. The last-frame score is then gated by b:

$$S_{\text{final}} = b \cdot \big(0.5 + 0.5 \cdot (0.5 \cdot q_{\text{tgt}} + 0.5 \cdot q_{\text{clean}})\big), \tag{5}$$

where $q_{\text{tgt}} = \max(0,\, 1 - r/N)$ penalizes the r colored balls still present among the N targets, and q_{clean} penalizes stray colored pixels. Because b multiplies the entire term, a missing or incorrectly sized black ball collapses the score. *Process score.* Each ground-truth eat event is evaluated using four checks:

$$S_{\text{eat}} = \tfrac{1}{N} \sum_{k=1}^{N} (0.25 \cdot o_k + 0.25 \cdot a_k + 0.30 \cdot g_k + 0.20 \cdot s_k), \tag{6}$$

where o_k rewards the correct size rank, a_k the black ball's arrival at and contact with the target, g_k its growth by the expected factor, and s_k the stillness of the not-yet-eaten balls. A target that is never eaten contributes 0.

Ball-Cluster Merging (Task O-29) As illustrated in Figure 22, a survivor cluster sequentially absorbs the other colored clusters until only it remains, and its label must show the total count. We track each cluster's pixel presence across frames to detect merge events and use OCR to read the final count. The score is a weighted sum of a final-state and a merge-process term, which is then controlled by a process gate $g_{\text{proc}} = 0.2 + 0.8 \cdot S_{\text{proc}}$:

$$S = (0.6 \cdot S_{\text{final}} + 0.4 \cdot S_{\text{proc}}) \cdot g_{\text{proc}}. \tag{7}$$

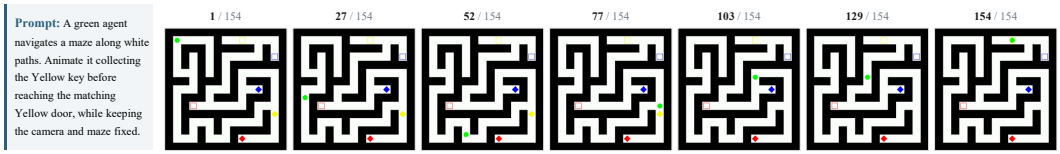

Figure 23: Key-Door Navigation (Task G-45): prompt and ground-truth frames. A colored agent must reach the key cell and then the door cell without crossing walls.

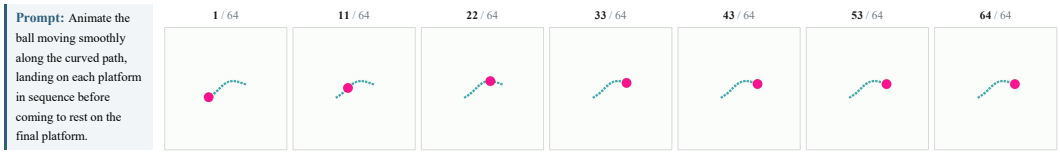

Figure 24: Rolling Ball (Task O-32): prompt and ground-truth frames. A ball rolls along the fixed curved path to the target.

so a poor merging process suppresses the overall score. *Final-state score.* The last frame is evaluated in terms of the survivor count, cluster compactness, and scene cleanliness:

$$S_{\text{final}} = (0.5 \cdot q_{\text{surv}} + 0.5 \cdot q_{\text{cluster}}) \cdot (0.6 + 0.4 \cdot q_{\text{pen}}) \cdot (0.8 + 0.2 \cdot q_{\text{text}}), \tag{8}$$

where q_{surv} measures whether the survivor count is correct, q_{cluster} whether the survivors form a single compact group, $q_{\text{pen}} = 0.5 \cdot q_{\text{gone}} + 0.5 \cdot q_{\text{clean}}$ whether the other colors have disappeared without leaving stray pixels, and q_{text} whether the label matches the total count. *Merge-process score.* The process score combines the average quality of individual merge events with a survivor-stability term:

$$S_{\text{proc}} = 0.7 \cdot \tfrac{1}{M} \sum_{k=1}^{M} (0.2 \cdot o_k + 0.3 \cdot a_k + 0.3 \cdot c_k + 0.2 \cdot s_k) + 0.3 \cdot q_{\text{stab}}, \tag{9}$$

where o_k, a_k, c_k, and s_k score the order, arrival, resulting count, and stillness of the k-th merge, respectively, and q_{stab} rewards a stable survivor count between consecutive merges.

Key-Door Navigation (Task G-45) As shown in Figure 23, a colored agent must move from its starting cell to the key cell and then to the door cell without crossing walls. We track the agent's per-frame centroid using color segmentation, reconstruct its path, detect wall crossings, and determine whether the matching door is reached. The score multiplies a trajectory term by two constraint gates:

$$S = \underbrace{\tfrac{1}{3}(\text{proximity} + \text{continuity} + \text{coverage})}_{\text{trajectory}} \cdot (0.4 + 0.6 \cdot g_{\text{wall}}) \cdot (0.4 + 0.6 \cdot g_{\text{kd}}). \tag{10}$$

Trajectory score. proximity measures closeness to the optimal path, rescaled by how near the agent's final position comes to the door, so that a stationary video cannot score highly; *coverage* is the fraction of the agent's decoded cell-to-cell trajectory that is contiguous motion rather than teleportation patched in by shortest-path fill, with the final hop counted as contiguous if it ends within one cell of the target; and *continuity* penalizes teleportation. *Constraint gates.* $g_{\text{wall}} = \max(0.4,\ 1 - 0.15 \cdot n_{\text{hits}})$ penalizes the n_{hits} wall cells entered, while the key–door gate $g_{\text{kd}} = 1$ if the agent reaches the matching door and 0.25 otherwise. Each gate is linearly mapped into $[0.4, 1]$ via $0.4 + 0.6 \cdot g$, so violating either constraint sharply reduces the score; for example, failing to reach the door caps its multiplier at $0.4 + 0.6 \cdot 0.25 = 0.55$.

Rolling Ball (Task O-32) Figure 24 shows a ball rolling along a fixed curved path toward a target. We detect the ball in each frame and measure the fraction of frames in which the ball lies on the path. The score gates a completion term by a scene-consistency term:

$$S = S_{\text{comp}} \cdot (0.6 + 0.4 \cdot q_{\text{cons}}). \tag{11}$$

Completion score. Final-position accuracy is gated by a path-following term:

$$S_{\text{comp}} = q_{\text{pos}} \cdot (0.2 + 0.8 \cdot p), \qquad p = q_{\text{on-path}} \cdot q_{\text{dir}} \cdot q_{\text{mid}}, \tag{12}$$

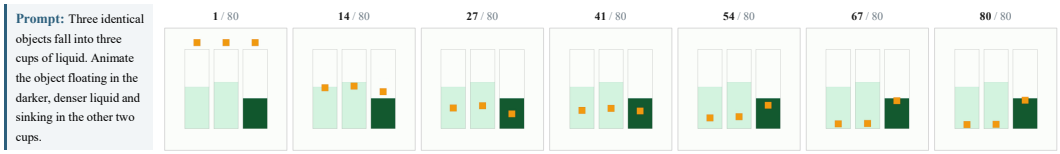

Figure 25: High-Density Liquid (Task G-273): prompt and ground-truth frames. Three objects fall into cups; the object above the denser liquid should float rather than sink.

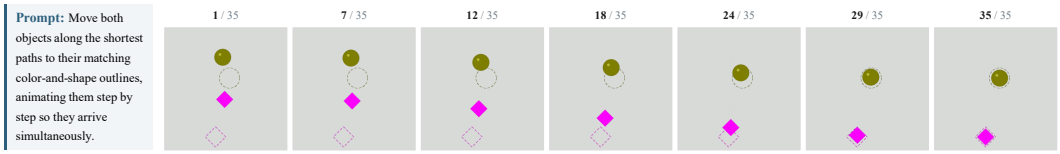

Figure 26: Move Objects to Targets (Task O-27): prompt and ground-truth frames. Two objects move to their corresponding target outlines.

where q_{pos} rewards ending near the target, and p is the product of $q_{\text{on-path}}$ (the fraction of frames in which the ball lies on the path), q_{dir} (motion in the correct forward direction), and q_{mid} (genuine intermediate motion). Because these factors are multiplied, a ball that rarely follows the path receives a low S_{comp} even if it ends near the target. *Consistency gate.* q_{cons} averages scene- and background-preservation scores, scaling the completion score by a factor in $[0.6, 1]$.

High-Density Liquid (Task G-273) As illustrated in Figure 25, three identical objects fall into three cups, one of which contains a denser liquid and should therefore cause its object to float rather than sink. We track each object's descent, classify its final state as floating or sinking based on its resting height, and check the liquid levels and background. The score is a weighted sum of final-state and process terms:

$$S = 0.6 \cdot S_{\text{final}} + 0.4 \cdot S_{\text{proc}}. \tag{13}$$

Final-state score. The final object positions are evaluated together with scene preservation:

$$S_{\text{final}} = q_{\text{pos}} \cdot (0.5 + 0.5 \cdot q_{\text{scene}}), \tag{14}$$

where q_{pos} rewards each object for resting at its correct floating or sinking height, and q_{scene} measures whether the cups, liquid, and background are preserved. *Process score.* The descent must remain physically plausible:

$$S_{\text{proc}} = q_{\text{ntp}} \cdot (0.1 + 0.3 \cdot q_{\text{opc}} + 0.3 \cdot q_{\text{stab}} + 0.3 \cdot q_{\text{area}}), \tag{15}$$

where q_{ntp} requires non-teleporting descent, q_{opc} checks that each cup column contains at most one object, q_{stab} measures the integrity of the intermediate frames, and q_{area} rewards consistent object size throughout the video.

Move Objects to Targets (Task O-27) As shown in Figure 26, two objects must be moved to their corresponding target outlines. We track each object's trajectory and evaluate its transport before checking whether the two objects move concurrently. The score gates the mean transport quality by a synchronization term:

$$S = S_{\text{mov}} \cdot (0.8 + 0.2 \cdot q_{\text{sync}}). \tag{16}$$

Transport score. For each object, transport quality is computed as a product:

$$S_{\text{mov}} = \tfrac{1}{2} \sum_{i=1}^{2} q_{\text{trans}}^{(i)} \cdot q_{\text{vis}}^{(i)} \cdot q_{\text{tp}}^{(i)} \cdot q_{\text{dur}}^{(i)}, \tag{17}$$

where q_{trans} combines path quality and endpoint accuracy, q_{vis} is the fraction of frames in which the object is successfully tracked, q_{tp} penalizes teleportation, and q_{dur} rewards a realistic motion duration. *Synchronization.* $q_{\text{sync}} = \sqrt{q_{\text{start}} \cdot q_{\text{end}}}$ rewards the two objects for starting and finishing at similar times. Moving them sequentially drives $q_{\text{sync}} \to 0$ and limits the score to $0.8 \cdot$ movement.