VACT: A VIDEO AUTOMATIC CAUSAL TESTING SYSTEM AND A BENCHMARK

Anonymous authors
Paper under double-blind review

ABSTRACT

With the rapid advancement of text-conditioned Video Generation Models (VGMs), the quality of generated videos has significantly improved, bringing these models closer to functioning as "world simulators" and making real-world-level video generation more accessible and cost-effective. However, the generated videos often contain factual inaccuracies and lack understanding of fundamental physical laws. While some previous studies have highlighted this issue in limited domains through manual analysis, a comprehensive solution has not yet been established, primarily due to the absence of a generalized, automated approach for modeling and assessing the causal reasoning of these models across diverse scenarios. To address this gap, we propose an automated framework for modeling, evaluating, and measuring the causal understanding of VGMs in real-world scenarios. By combining causal analysis techniques with a carefully designed large language model assistant, our system can assess the causal behavior of models in various contexts without human annotation, which offers strong generalization and scalability. Additionally, we introduce multi-level causal evaluation metrics to provide a detailed analysis of the causal performance of VGMs. As a demonstration, we use our framework to benchmark several prevailing VGMs, offering insight into their causal reasoning capabilities. Our work lays the foundation for systematically addressing the causal understanding deficiencies in VGMs and contributes to advancing their reliability and real-world applicability.

1 INTRODUCTION

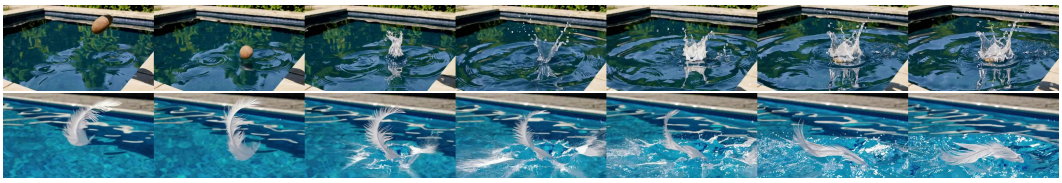


Figure 1: Videos generated by OpenAI Sora, shown as frames. The text prompt of the **Above** is: *a stone is thrown into a swimming pool*; **Below** is: *a feather is thrown into a swimming pool*. Both the generations show *noticeable splashes*, which is correct for the above (stone) scene but **incorrect** for the **below** (feather) scene.

With the rapid development of Video Generation Models (VGMs), generated videos are becoming increasingly indistinguishable from real recordings. VGMs, particularly text-to-video (T2V) models[1], are expected to serve as "world models" or "world simulators", allowing users to generate scenes from text descriptions of real-world events or environments. This approach is cheaper, faster, and more scalable than arranging and recording real-world scenes and is expected to benefit fields like robotics, autonomous driving, and video understanding.

[1]In this paper, the term VGM is referred specifically to T2V models. Text-conditioned generation is the most versatile and user-friendly method in world simulation, whereas image-conditioned models can enable T2V generation by combining with a text-to-image model.

However, the "*hallucination*" problem hinders the progress, which refers to a generation that seems correct but contains factual errors or fabrications. While VGMs have made significant strides in video quality—such as clarity, dynamic range, and continuity—they still struggle with issues like cause-and-effect confusion, detail errors, and incorrect object relationships, making the videos appear misleading upon closer inspection.

In Figure 1, OpenAI Sora (OpenAI, 2024b) is required to generate videos for two scenarios: "*a stone is thrown into a swimming pool*" and "*a feather is thrown into a swimming pool*". In both cases, an obvious splash and ripples occur around the object. While in the stone scenario the splash is accurate, the feather scenario fails to follow the correct physics principles, as the feather is too light to create a noticeable splash or ripples in reality. Here, the model seems to learn a **spurious correlation** between "*object hitting water*" and "*splash*", without understanding the actual causal factors, such as *mass* and *velocity*. We provide similar results with other VGMs in Appendix A.

Although some work has acknowledged the hallucination problem in VGMs and proposed preliminary benchmarks to identify commonsense violations (Bansal et al., 2024; Meng et al., 2024), most of them rely on manual rule design and focus on limited fields. However, real-world causal relationships are highly complex, with different scenarios involving different physical laws. Furthermore, even a simple scenario can involve various causal relationships. For instance, in the case of "*two objects collision*", dynamics might focus on mass, velocity, and elasticity to determine the object motion after collision, while material properties like hardness or brittleness might determine whether the objects deform or break. More complex relationships, like sparks from a flint or splashes from wet objects, further highlight this complexity, making it difficult to systematically address the hallucination problem through manual labeling.

To address this challenge, we propose an **automatic** method for identifying causal rules in specific scenarios and evaluating models' **causal understanding**. Our process, utilizing an LLM, generates possible causal rules (referred to as the causal system) for a given scenario. **Intervention experiments** (Pearl, 2009) are then used to assess causal behaviors in VGMs by varying the text prompts with different factor values. For example, as shown in Figure 1, replacing a heavy stone with a light feather revealed that the VGM had not correctly learned the causal rules related to density.

To analyze causal learning in VGMs, we define three levels of consistency: **text consistency**, **generation consistency** and **rule consistency**. These metrics assess the model's ability to follow explicit causes (and results), maintain consistent generation under the same conditions, and learn correct causal rules, with progressively higher levels of difficulty.

In summary, we introduce the **V**ideo **A**utomatic **C**ausal **T**esting (**VACT**) system, which requires no human annotation, scoring, or intervention. To our knowledge, this is **the first approach to automatically apply causal analysis tools for testing causal understanding in VGMs**. It is scalable, generalizable, and can be applied across various fields without additional manual effort, while also providing a detailed causal analysis of model behavior. To validate its effectiveness and generalizability, we conducted crowd experiments, where 60 causal systems under 20 different scenarios by our system (involving various scenarios such as motion, force, light, heat, fluid, material, etc.) are compared to human annotation, showing that automatic annotations achieve comparable (and even better) performance with human annotation. We also use these 60 systems to construct a benchmark to assess current video generation models, revealing that no existing model achieves satisfactory causal learning. This system offers a powerful tool to enhance our understanding of VGM reliability and lays the groundwork for a systematic solution to the hallucination problem, like dataset supplementation or alignment by reinforcement learning.

2 Related work

Text-to-Video(T2V) generation models. T2V models generate videos from textual descriptions. Early methods using generative adversarial networks (GANs)(Wang et al., 2020) and variational autoencoders (VAEs)(Li et al., 2018; Pan et al., 2017) faced limitations like low resolution and diversity. Starting with Video Diffusion Models (Ho et al., 2022), recent advances in diffusion models have significantly improved T2V generation. CogVideo (Hong et al., 2022) combines a pre-trained text-to-image model with a text-to-video framework, facilitating more effective learning. LaVie (Wang et al., 2024) enhances video quality with interpolation and super-resolution techniques. VideoCrafter2 (Chen et al., 2024) leverages Diffusion Transformers(DiT) (Peebles & Xie, 2023) to

synthesize high-quality videos by refining generated sequences with high-resolution images. Models like Gen-3 Alpha (Esser et al., 2023), HunyuanVideo (Tencent, 2025), and Sora (OpenAI, 2024b) further push the boundaries with advanced architectures and processing techniques. Comprehensive reviews on these developments are available in Xing et al. (2024) and Sun et al. (2024).

Evaluation for video generation models. The rapid advancement of VGMs has underscored the need for accurate quality evaluation. Traditional metrics like IS (Salimans et al., 2016), FVD (Unterthiner et al., 2019), and CLIP (Hessel et al., 2022; Liu et al., 2023) assess only limited aspects like frame quality, and often fail to align with human judgment. To address this, benchmarks like V-Bench (Huang et al., 2024) and EvalCrafter (Liu et al., 2024) provide more comprehensive evaluations, considering factors like subject consistency, spatial relationships, and action continuity. However, these metrics still focus on visual quality while overlooking the logical coherence of events and scenes in videos.

Evaluation for world simulators. As video quality further improves and the concept of a "*world simulator*" becomes an expectation, the focus has shifted from *aesthetics* to *authenticity* — ensuring generated content follows real-world physics rules. Recent benchmarks including VideoPhy (Bansal et al., 2024) and PhyGenBench (Meng et al., 2024) have made initial attempts to address this. VideoPhy uses human annotations to verify commonsense violations, making it labor-intensive and difficult to generalize. Their attempts to fine-tune a vision-text model for automatic ranking have yet to align well with human assessments, limiting its scalability. PhyGenBench (Meng et al., 2024) tests on 27 *human-designed* physics laws, using LLM-generated questions to check rule fidelity in videos by a video language model. Our work further expands this series of work in two aspects: 1) **Full automation**: our approach eliminates manual rule design, allowing physical rules to be automatically inferred from a short textual descriptions, enhancing scalability. 2) **Causal evaluation**: We introduce *intervention experiments* to test whether models truly understand physics rather than relying on shortcuts, ensuring a more robust assessment. Additionally, other works like Kang et al. (2024) explore 2D physics simulation in VGMs, while WorldSimBench(Qin et al., 2024) assesses world simulators from an embodied perspective. These works, along with ours, collectively contribute to a multi-faceted understanding of world simulators,

3 VACT: THE PIPELINE OF AUTOMATIC CAUSAL RULE TESTING

3.1 SCENARIO-BASED CAUSAL RULE TESTING

Our tests begin with **scenario**s, short text descriptions of a event, such as "something is thrown into a swimming pool" (Figure 1). Each scenario involves variables representing object or event properties, linked by causal relationships modeled using a causal graph and a causal system.

Definition 1 (Causal graph and system (Pearl, 2009))**.** A deterministic causal system over a set of variables $\mathbf{V}$ is a directed acyclic graph G with node set $\mathbf{V}$ and edge set $\mathbf{E}$, and a series of structural equations $V_j = f_j(pa(V_j))$ for every $V_j \in \mathbf{V}$, where $pa(V_j) = \{V_k \in \mathbf{V} : V_k \to V_j \in \mathbf{E}\}$. Furthermore, let $\mathbf{X} = \{V_j \in \mathbf{V} : pa(V_j) = \emptyset\}$ be the **root** (**cause**) variables and $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ be the **non-root** (**outcome**) variables.

We provide an example in Figure 2. The system describes some physics commonsense that density affects whether the object will sink and speed, size and density affect the splash. Directed edges in the graph represent causal relationships between variables, like the edge "high density" $\to$ "object sink" indicating causation while no causal effect existing between "large size" and "object sink", as a dense object will sink regardless of its size. The basic unit of our VACT is a causal system, consists of these rules. One scenario may generate different test cases depending on the selected factors.

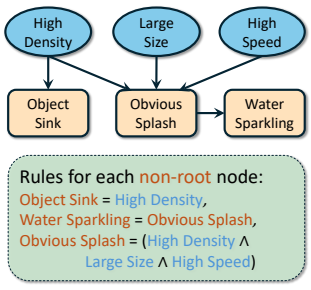


Figure 2: An example causal graph and system: "*throwing something into a swimming pool*". Blue denotes root nodes and orange denotes non-root nodes.

For clarity, all variables in our system are *Boolean*, meaning the rules are Boolean functions. This simplification reasonably abstracts physical relationships, avoiding complex calculations while preserving essential causal structures. Continuous properties, such as speed or mass, can be binarized (e.g., "fast" vs. "slow" or "light" vs. "heavy"), as we often make judgments using such discrete cat-

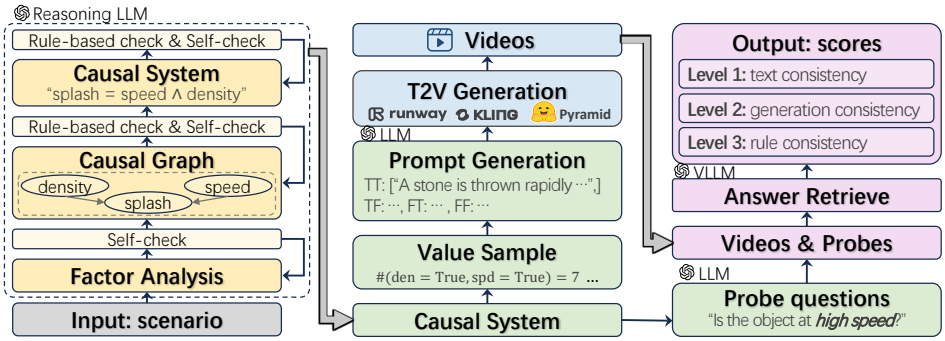


Figure 3: Pipeline of VACT

egories in daily life. Additionally, variables must be visually discernible[2] (*visibility*) to ensure suitability for video evaluation, and all root nodes can be *independently* sampled. If a video generation model learns the physical laws of a scenario, it should have learned the rule f. Thus, by analyzing variable states in generated videos under different conditions, we can assess whether the model understands the underlying law.

3.2 LLM-AIDED AUTOMATIC GENERATION OF THE TEST CASES

As discussed in Section 1, extracting key causal rules from a scenario is challenging due to its complexity and diversity, requiring creativity (to imagine alternative scenarios) and commonsense (to identify common causal patterns). Fortunately, the advanced commonsense reasoning of LLMs enables automation of this task. We designed a multi-step annotation process using LLMs. As shown in Figure 3 (yellow part), the system takes a scenario (a short text description) as input and prompts the LLM to: (1) analyze key causal factors, (2) construct a causal graph linking variables and outcomes, and (3) derive Boolean expressions representing these relationships. The final test case consists of the expression with the scenario. We designed self-checking and external checking for these steps, leveraging the LLM's self-correction ability to improve result reliability. For detailed generation requirements, inspection indicators and the full process, see Appendix B.

Crowd experiments. We evaluate the effectiveness of the automatic generation by crowd experiments. We collected 20 diverse scenarios (listed in Appendix C) and generated three causal systems for each. For comparison, three undergraduates manually annotated the same scenarios using identical instructions given to the LLM, resulting in another 60 causal systems. Another five undergraduates then blindly scored both human and LLM annotations based on three criteria: requirement (adherence to visibility, binarity and root-independence), rationality (reasonableness of factor selection), and soundness (accuracy of causal rules). As shown in Table 1, LLM-generated annotations surprisingly outperformed those from human, demonstrating the effectiveness of the LLM-driven process and its strong alignment with human reasoning. For further details and analysis, see Appendix D.

Table 1: Scores from crowd experiments

Source	Requirement	Rationality	Soundness	Average
LLM	**3.91**$\pm$0.02	**3.49**$\pm$0.04	**3.78**$\pm$0.03	**3.73**$\pm$0.02
Human	3.80$\pm$0.03	**3.51**$\pm$0.04	3.63$\pm$0.04	3.65$\pm$0.03

3.3 AUTOMATIC INTERVENTION EXPERIMENT PIPELINE

Given a causal system, our testing as an intervention experiment contains five parts: sampling, prompt generation, video generation, answer retrieval, and evaluation, as shown in Figure 3. Details of these steps can be found in Appendix E.

Sampling. We sample various combinations of root values $\mathbf{X}$ for intervention experiments. The number of samples per $\mathbf{X}$ value is determined by the metrics outlined in Section 4 and detailed in Appendix F.4. In our experiments, we collected approximately 30 - 45 samples per causal system.

Prompt generation. Given $\mathbf{X}$ values, we use an LLM to generate sentences to constrain the scenario accordingly. For example, in the scenario "throwing something into a swimming pool", the prompt "*a large rock was thrown quickly into the pool*" sets three root variables: high density, large size, and high speed to true, while "*a tiny stone is gently placed on the water*" alert large size and high speed to false. These sentences serve as text prompts for video generation.

Video generation. The prompts generated are provided to the tested VGM. These models are treated as black boxes, requiring no constraints on their structure.

[2]The visualization here is a relative requirement. For example, although density is essentially invisible, we can infer the density of an object through its visible material.

Answer retrieval. Each generated video serves as an observation of the intervention experiments for the causal system. We check (1) whether it follows the text description of variable values $\mathbf{X}$ and (2) whether the generated values $\mathbf{Y}$ align with the causal rules. Following Meng et al. (2024), we use a vision-LLM (VLLM) to retrieve the observed values $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ by prompting the VLLM with "yes-no" questions (i.e., probes, generated alongside the causal system).

We adopt an LLM and an Video-LLM to automate the steps *prompt generation* and *Answer retrieval*. To ensure the correctness, we performed random manual checks. We found that the vast majority of the results are reliable. The detailed analysis and check results are shown in Appendix G.

4 THREE LEVELS OF CAUSAL ABILITY AND THE CORRESPONDING METRICS

To assess the deviation of the model's understanding of the objective world, we propose a three-level framework of causal capabilities with corresponding evaluation metrics. The detailed mathematical definitions are provided in Appendix F. Here, we focus on an intuitive description of them.

Text consistency. The first level assesses whether the model accurately reflects the state of every variable described in the prompt. By generating a video from a detailed prompt specifying certain variable values, the resulting video should correctly reflect those values. This ensures the model faithfully interprets input text—a fundamental requirement for our intervention experiments, where we need to control video variables through text. We use two types of prompts: "root" specifies *all root* variables $\mathbf{X}$ and "all" specifies *all* variables $\mathbf{V}$. For each setting, the metric is measured by the average accuracy of whether observed values match the described ones.

Generation consistency. The second level evaluates whether the model stably produces the same outcomes given identical causes $\mathbf{X}$, or if its outcomes vary arbitrarily due to unrelated factors like random seed or wording differences. To measure this, we group samples by identical $\mathbf{X}$ values, and calculate the mean variance of outcomes Y_i within each group. To address errors from imperfect text consistency, we use two scoring criteria: Groundtruth-based grouping ($\mathbf{X}$) evaluates end-to-end consistency, while observation-based grouping ($\hat{\mathbf{X}}$) ignores condition generation errors. As text consistency improves, both scores should converge.

Rule consistency. The third level, our main long-term goal, tests the model's ability to learn and apply causal rules consistent with the real world. For sampled videos $\mathbf{S}$, the rule consistency is calculated as the average of a score $m(\mathbf{Y}(s), \hat{\mathbf{Y}}(s))$. We use the average accuracy among variables $\mathbf{Y}$: $\sum_{Y_i \in \mathbf{Y}} 1(Y_i = \hat{Y}_i)/|\mathbf{Y}|$, and threshold 0-1 score $\mathbb{1}\{\text{mean}(\mathbb{1}(Y_i = \hat{Y}_i)) \geq t\}$ as the score m. We also distinguish two scores, one using the groundtruth $pa(Y_i)$ and another using the observed $\hat{pa}(Y_i)$ to get the expected $Y_i = f(pa(Y_i))$ where the latter excluding errors from unexpected causes.

These metrics can be also applied to individual videos, convenient to identify specific instances where the model's performance deviates, providing insights into its learning mechanisms. See Appendix F for detailed definitions and some analysis in Appendix I.2.

5 A BENCHMARK OF CAUSAL RULE TESTING

In this section, we use the collected 60 causal systems from 20 different scenarios (see Section 3.2) as a testbed to evaluate causal learning of prevailing VGMs. We found that these models occasionally generate videos that are off-topic or with missing subjects and confusing logic. So we allowed the VLLM to answer "N/A" (in addition to yes/no) during answer retrieval, filtering out all observations marked as "N/A" across all metrics. Here, rule consistency is calculated as the average accuracy score. For details on the models, costs, the impact of N/A, sample efficiency, and threshold-based rule consistency, see Appendix H.1 to H.5.

Table 2 shows our benchmarking results on some prevailing open- or closed-source models. We observed that all the existing models did not perform satisfactorily, with only minor differences between them. Specifically:

Text consistency: The accuracy ranged from 55-65%, with random guessing at 50%, indicating that models struggle to generate variables accurately based on the provided text, for both causal and outcome variables (no significant difference between all and root). While text fidelity has improved (Sun et al., 2024), our tests require handling multiple variables simultaneously. Additionally, some values correspond to less common scenarios (like *feather* instead of stone into water). The

Table 2: VACT benchmark on prevailing VGMs.

Model Names	N/A ratio	Text Consistency ↑		Generation Consistency ↓		Rule Consistency ↑	
		all	root	truth	observe	truth	observe
CogVideoX1.5-5B	.07	$.56_{\pm.01}$	$.61_{\pm.01}$	$.10_{\pm.00}$	$.09_{\pm.01}$	$.55_{\pm.01}$	$.72_{\pm.02}$
CogVideoX-5B	.07	$.58_{\pm.01}$	$.64_{\pm.02}$	$.09_{\pm.00}$	$.09_{\pm.01}$	$.56_{\pm.01}$	$.71_{\pm.03}$
CogVideoX-2B	.09	$.56_{\pm.01}$	$.63_{\pm.01}$	$.09_{\pm.01}$	$.09_{\pm.01}$	$.59_{\pm.02}$	$.72_{\pm.03}$
VideoCrafter2	.12	$.55_{\pm.01}$	$.58_{\pm.02}$	$.08_{\pm.01}$	$.06_{\pm.01}$	$.53_{\pm.02}$	$.73_{\pm.03}$
Pyramid Flow	.10	$.56_{\pm.01}$	$.61_{\pm.02}$	$.07_{\pm.00}$	$.06_{\pm.01}$	$.56_{\pm.01}$	$.72_{\pm.03}$
HunyuanVideo	.07	$.58_{\pm.01}$	$.63_{\pm.01}$	$.08_{\pm.01}$	$.07_{\pm.01}$	$.57_{\pm.01}$	$.70_{\pm.02}$
Pika	.10	$.57_{\pm.01}$	$.60_{\pm.01}$	$.09_{\pm.00}$	$.08_{\pm.01}$	$.56_{\pm.01}$	$.76_{\pm.02}$
Hailuo	.07	$.59_{\pm.01}$	$.64_{\pm.01}$	$.10_{\pm.00}$	$.08_{\pm.01}$	$.59_{\pm.01}$	$.73_{\pm.02}$
Gen-3 Alpha	.06	$.63_{\pm.01}$	$.63_{\pm.01}$	$.08_{\pm.00}$	$.08_{\pm.01}$	$.57_{\pm.01}$	$.74_{\pm.02}$
Kling	.07	$.63_{\pm.01}$	$.64_{\pm.01}$	$.07_{\pm.00}$	$.07_{\pm.01}$	$.57_{\pm.02}$	$.71_{\pm.02}$

low score here highlights the models' difficulty in handling less common, variable-replacement situations, implying that models are strongly limited to the common situations and cannot easily generalize to combine independent variables in scenarios.

Generation consistency: Roughly estimated, with a two-point distribution variance of $p(1-p)$, a value around 0.1 corresponds to a 10% deviation rate, indicating that the model has learned a relative stable "rule" of outcomes, that is, producing relatively consistent outcomes for the same $\mathbf{X}$. However, this stability is not necessarily a positive sign. When considering both level 1 and level 2 results, despite around 40% of root variables being generated incorrectly, the truth and observation scores for metric 2 are very close. This suggests that the model's stability reflects a "*degenerative*" rule, where models often generate a fixed outcome $\mathbf{Y}$ ignoring variations in $\mathbf{X}$ (like a constant function). Just as shown in Figure 1, any object entering water always generates a splash. We also confirmed it through manual inspection, see I.1.

Rule consistency: Finally, we directly assess the correctness of the rules learned by the models. The results align with our previous analysis: the model's average rule accuracy is below 60% for the truth and only around 70% for observation, with random guessing corresponding to 50%. Further analysis with the threshold scores provided in the Appendix H.5 shows that fewer than 20% of the rules match the groundtruth in 95% of the samples, while nearly 30% of the rules have an accuracy below 65%. These findings clearly indicate that the models have not correctly understood the relationship between outcomes and causes (or parents), revealing weak rule learning of the current models.

We also benchmark some models using the human-annotated causal systems, obtaining similar results (shown in Appendix H.6). This serves an evidence for both the effectiveness of our automatic annotation and the validity of our benchmark conclusions.

6 Conclusion & Limitation

In this paper, we propose an automated system for modeling causal relationships in scenarios and evaluating the causal behavior of VGMs. By combining LLM's commonsense understanding with intervention experiments, our automatic system can assess the causal learning in VGMs across diverse domains, scenarios, and rules. We validated its effectiveness through crowd experiments and manual checks. We introduced three progressive causal metrics to comprehensively analyze the model's causal behavior. Using this system, we created a benchmark and identified key causal flaws in existing models. As a long-term target, this work lays the foundation for large-scale detection of shortcut or biased learning, supplement comprehensive training datasets, or reinforcement learning.

However, we acknowledge several limitations in our current work. First, although the LLMs can generate high-quality testbeds, occasional errors may still occur. For scenarios requiring extremely high quality assurance, human assistance is still recommended. Second, our causal system construction involves certain simplifications, such as focusing only on visualized factors and binarizing variables, which may need refinement for more complex scenarios, such as extending binarization to multiple discrete levels. This remains our future work. Lastly, our evaluation assumes that model generate high-quality videos but some models still struggle with text understanding and coherent video generation, hindering the analysis of their causal behavior. We view our system as a forward-looking tool, believing that as video generation models rapidly improve, causal behavior analysis will become more critical.

REFERENCES

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation, 2024. URL https://arxiv.org/abs/2406.03520.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7346–7356, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. URL https://arxiv.org/abs/2104.08718.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. URL https://arxiv.org/abs/2205.15868.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling, 2024. URL https://arxiv.org/abs/2410.05954.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective, 2024. URL https://arxiv.org/abs/2411.02385.

Kuaishou, 2024. URL https://klingai.com.

Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models, 2024. URL https://arxiv.org/abs/2310.11440.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation, 2023. URL https://arxiv.org/abs/2311.01813.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation, 2024. URL https://arxiv.org/abs/2410.05363.

MiniMax, 2024. URL https://hailuoai.video.

OpenAI, 2024a. URL https://openai.com/index/introducing-openai-o1-preview/.

OpenAI. Sora is here, 2024b. URL https://openai.com/index/sora-is-here.

Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1789–1798, 2017.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Pika, 2024. URL https://pika.art.

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

Runway, 2024. URL https://runwayml.com/research/introducing-gen-3-alpha.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Rui Sun, Yumin Zhang, Tejal Shah, Jiahao Sun, Shuoying Zhang, Wenqi Li, Haoran Duan, Bo Wei, and Rajiv Ranjan. From sora what we can see: A survey of text-to-video generation, 2024. URL https://arxiv.org/abs/2405.10674.

Tencent. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1160–1169, 2020.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, pp. 1–20, 2024.

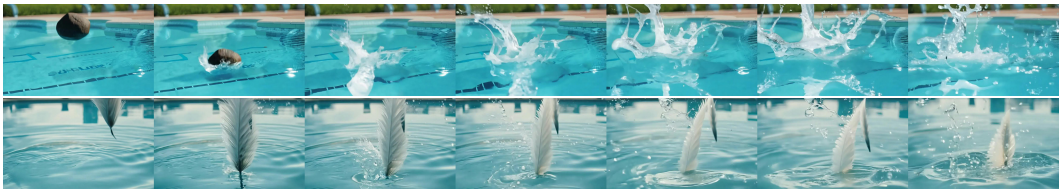
Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *ACM Computing Surveys*, 57(2):1–42, 2024.

A THE "STONE" AND "FEATHER" EXAMPLE FOR OTHER MODELS

In Figure 1, we demonstrate that OpenAI's Sora (OpenAI, 2024b) fails to distinguish between the different effects of a stone and a feather falling into water. This is not an isolated case of Sora. In figure 4, we show the generation of CogVideoX-2 (Hong et al., 2022) and Runway Gen-3 Alpha (Runway, 2024), showing that this spurious correlation is a common phenomenon that may exist in various models. These models seem to "directly" substitute the stone with the feather, without understanding the significant differences in the outcomes.

(a) OpenAI Sora Generation

(b) CogVideoX-2 Generation



(c) Gen-3 Alpha Generation

Figure 4: Videos generated by (a) OpenAI Sora, (b) CogVideoX-2 and (c) Gen-3 Alpha, shown as frames. For each model, the text prompt of the **Above** is: *a stone is thrown into a swimming pool*; **Below** is: *a feather is thrown into a swimming pool*. Both generation show *noticeable splashes*, which is correct for the above (stone) scene but **incorrect** for the **below** (feather) scene.

On the one hand, this spurious correlation, we believe, comes from the distribution of the data set. We found that videos of stones being thrown into water are abundant online, while videos of feathers being thrown into water are significantly less common. As supporting evidence, a search for "thrown stone into water video" returns approximately 180,000,000 results, while replacing "stone" with "feather" reduces the results to around 31,000,000. This data bias means that the model may have seen enough scenes of stones entering the water during training but not enough scenes of feathers doing the same. Additionally, this issue stems from the widespread overfitting of current VGM models, which causes them to rely heavily on common data in the dataset without fully understanding the underlying rules of the scene; in contrast, the current LLM like GPT-4o can more effectively grasp the different outcomes caused by various objects falling into the water. In this case, the language model can distinguish that feathers falling into the water will not cause splashes.

B DETAILS OF AUTOMATIC GENERATION OF CAUSAL SYSTEMS

B.1 DETAILS OF GENERATING PROCESS

We use the official API of OpenAI o1 model (o1-2024-12-17) (OpenAI, 2024a) to generate the causal systems. The three tasks are divided and prompted sequentially, with the LLM completing them through multiple rounds of dialogue. Throughout this process, the entire dialogue history is retained within the context window. The model will proceed to the next task either once the

maximum number of attempts is reached or when the external checks are passed and the LLM retains its answer after a self-check.

We require that the generated content for each step includes a file containing specific information, where:

- **Factor analysis**: a json file as a list of dictionary containing:
  - "`type`": choices from "factor" or "result".
  - "`name`": the name of the factor or result variable. They could be some words or a short sentence that can summarize the key meaning.
  - "`explanation`": A short explanation about how the factor or result can affect the scenario and why the variable is visible, binary and important.
- **Causal Graph**: a dot file that constructs a digraph, which first declares each factor as a node, then declares some directed edges between nodes.
- **Causal System**: a json file as a list of dictionary containing:
  - "`scenario`": a string describing the event,
  - "`roots`": a list of strings, each of which is a name of cause variable,
  - "`non_roots`": a list of strings, each of which is a name of outcome variable,
  - "`rules`": a dictionary where each outcome variable corresponds to a Boolean function of its parents in the causal graph. The boolean function should be expressed as a disjunctive normal form (DNF), where each conjunctive clause are expressed as a dictionary ($A \wedge B \wedge \neg C$ expressed as $\{$`'A': True, 'B': True, 'C': False`$\}$. And the DNF is expressed as a list of the dictionary-expressed conjunctive clause.

The complete generation process consumes roughly 20k reading tokens (10k cached) and 10k prediction tokens, costing about $0.74 per causal system. This is approximately one-third the cost of manual labeling, which is 15 CNY per annotation.

B.2 Requirement: rule-based & self correction

We have specific requirements for both the internal results and the final output causal systems. The detailed requirements can be found in the prompt in Appendix B.3. To ensure these requirements are met as thoroughly as possible, we have designed a check-and-correction loop.

Except for the first step "factor analysis", we use both the rule-based check and self-check for the answer generated by LLM. For the "causal graph", we check the following requirements by a Python program:

- whether the generated answer consists of a legal dot file,
- whether the graph is a DAG,
- whether there is an isolated node in the graph.

For the "causal system", we check that

- whether the returned rules keep the legal format, that is, it is a json file, with the correct keys (roots, non-roots, rules) and all values are in the correct format. Especially for the rules, we define a standard format to use a python list of dictionary to represent a disjunctive normal (DNF). We check whether the generated answer is a legal DNF.
- whether the rules leads to the same causal graph generated in the "causal graph" step,
- whether all the non-root nodes have exact one DNF and the root nodes do not have their DNF.

If any requirement has not been met, an error message will be the feedback to the LLM with the full history, and the LLM is required to regenerate its answer given the error message and the history information.

If the rule-based check has passed, we prompt the LLM to further check its answer by itself. The self-check prompts repeat the requirement in a more detailed way. These prompts are shown in Appendix B.3.

Although the current reasoning models like OpenAI o1 has learned to self-check during its thinking steps, we find the explicit self-check prompt can further help to improve the performance. For example, when asked to identify key factors in the scenario "Knife slicing through (butter)", o1 initially identifies "*Butter is cold and firm*", where, while accurate, the temperature is not easily visible in the video. After a self-check process, o1 revises its answer to "Butter is in block form", a factor that can be more easily identified in the video. We believe that it could be because in this step, an LLM can think in more detail about whether the answer satisfies the condition without having to take into account the generation task at the same time.

Considering that we have adopted a step-by-step strategy, we also allow the model to regret the previous answer in the subsequent steps. For example, when generating causal rules, if the model finds that the previous causal graph is unreasonable during the process, we allow the model to generate `<regenerate_graph>` to go back to the previous step. While this situation is rare, we have found that it effectively reduces the likelihood of the model producing low-quality answers.

We allow the model to generate `<keep_answer>` after self-checking. If this occurs, we skip the subsequent checking steps. We found that after a total of three checks, most conditions are met, and the model is typically satisfied with its answer, generating `<keep_answer>`.

B.3 PROMPTS

In this section, we provide all of the prompts we use to facilitate the LLM to generate causal systems.

Prompt for Identifying Key Factors in a Scenario:

> You will be provided with a brief description of a scenario. There could be some physical phenonmenon in this scenario. Please identify some **important** and **common** potential factors whose changes could significantly influence some important outcome of the scenario. These factors can fall into one of the following categories:
>
> 1. The objects or their properties in the scenario.
>
> 2. The object in the environment or the properties of the environment.
>
> 3. The actions or some properties of the action.
>
> For each factor, ensure that it meets the following criteria:
>
> 1. It should be **visible** and easily recognizable in a video.
>
> 2. It should be **binary**, meaning it can be clearly labeled as either "yes" or "no", rather than a continuous value.
>
> 3. It should be **independent**, not dependent on other factors.
>
> 4. Its effect on the outcome should be **deterministic** (i.e., it directly leads to a certain result, rather than just increasing or decreasing the probability).
>
> 5. The resulting effect should also be **visible** in a video.
>
> If there is a pair bracket in the description, it means the content in the bracket is expected to be a variable (factors or outcome). For example, "A (large) stone is thrown into a swimming pool (and splash water)." means we expect "does the water splash" as one of the outcome and whether the stone is large enough is expected as one of "factors". But notice that it does not mean that other factors or outcomes are not allowed, you can also propose other factors or outcomes.
>
> Please organize your answer as a **json** file as a list of dict, where each dict is like { "type": "factor_or_result", "name": "factor_or_result_name", "explanation": "how it affects

the scenario and why you believe it is important and common"}. Start your answer with a ⟨json⟩tag and end with a ⟨/json⟩tag.

Prompt for Causal Graph Construction:

Based on the factors you proposed and their expected results, generate a causal graph that summarizes the physical relationships between them. In the graph, include only the most important and common factors or results; omit any overly detailed or trivial ones.

The graph should be a **directed acyclic graph**, where:

- Each **node** represents a factor or a result.

- Each **edge** represents a direct causal relationship between two nodes.

The graph should be formatted in **DOT** format. Begin the DOT file with a ⟨dot⟩tag and end it with a ⟨/dot⟩tag.

Prompt for Causal Rule Generation:

Given the causal graph you generated, please create a Boolean expression for each **non-root** factor (factors with incoming edges) that represents the conditions under which that factor is **true**. The Boolean expression for each non-root factor should involve only the **parent factors** (i.e., the factors directly connected to it in the causal graph). The condition should be expressed as a **disjunctive normal form** (DNF), which is a disjunction (OR) of conjunctions (AND) of literals.

Your response should include a set of boolean expressions, formatted as a 'dict[str, list[dict[str, bool]]]', where the key is the name of this non-root factor and the value is a list of conditions (disjunctions), where each condition is a conjunction clauses (AND). Each condition is represented as a dictionary, where the key is the name of the parent factor and the value is a boolean value (True or False).

For example, if a factor A is true when B is true or (C is true and D is false), the boolean expression should be '{"A": [{"B": True}, {"C": True, "D": False}]}'.

Your final answer should be a JSON file with the following keys

- "roots": a list of root factors.

- "non_roots": a list of non-root factors.

- "rules": a dictionary where each non-root factor is associated with its corresponding Boolean expression.

Please begin your response with a ⟨json⟩tag and end with a ⟨/json⟩tag.

For self-check prompt for factors:

Please review the factors you have proposed. Ensure that each factor satisfies the following 5 requirements:

1. It should be **visible** and easily recognizable in a video.

2. It should be **binary**, meaning it can be clearly labeled as either "yes" or "no", rather than a continuous value.

3. It should be **independent**, not dependent on other factors.

4. Its effect on the outcome should be **deterministic** (i.e., it directly leads to a certain result, rather than just increasing or decreasing the probability).

5. The resulting effect should also be **visible** in a video.

Please ensure that the content in the bracket has been correctly identified as a variable (factor or outcome) in your answer.

Additionally, filter out any factors that are:

- **Too detailed**, **corner-case**, or **uncommon** in the scenario.

- Have an effect that is **too indirect** or difficult to understand.

If necessary, you may regenerate the factors to meet the criteria. It's OK to keep your previous answer by just generate ⟨keep_factor⟩ without any other words but you should carefully check every requirement for every factor and result.

For self-check prompt for graph:

Please review your causal graph. Ensure that it meets the following criteria:

1. All nodes are **visible** and **binary**.

2. All root nodes are **independent** of each other, which means the choice of one root node should not influence the choice of another root node.

3. All edges in the graph is a **direct** and **deterministic** causal relation

4. Include all **important** causes and results, while omitting trivial or overly detailed nodes.

Please ensure that the content in the bracket has been correctly identified as a variable (factor or outcome) in your answer.

If necessary, regenerate the causal graph to meet these requirements. It's OK to keep your previous graph if it already meets the criteria by just generate ⟨keep_graph⟩ without any other words but you should carefully check every requirement for every node and edge.

For self-check prompt for rules:

Please review your answer. Ensure your answer meets the following criteria:

1. The "roots" and "non_roots" list must be consistent with the causal graph.

2. For the bool expressions:

- All the nonroot factors are included in the rules dict, and no other factors are mistakenly included as keys.

- All variables in the Boolean expressions are exactly the parents of the corresponding non-root factors in the causal graph.

- The boolean expressions should correctly represent the physical rules in the real world.

If necessary, regenerate the json file to meet the requirements. It's OK to keep your previous rules if they already meet the criteria by just generate ⟨keep_rule_json⟩ without any other words but you should carefully check every requirement for every variable and rule.

If you find that you need to modify your generated causal graph, please generate ⟨regenerate_graph⟩⟨dot⟩... ⟨/dot⟩where the content between ⟨dot⟩and ⟨/dot⟩is the new causal graph.

C 20 SCENARIOS IN CROWD EXPERIMENTS AND BENCHMARK

The 20 scenarios used in our crowd experiments and benchmarks are listed below. These scenarios vary in the types of relationships they involve, their complexity, and the extent to which they include variables. To simulate situations where users may already have specific variables of interest, we also designed a "bracket" representation to prompt the LLM, indicating that the content within the brackets MUST be treated as a variable. Note that the "stone into water" scenario is not included in the list, as it serves as our debug case for adjusting the prompt and providing an example for human annotators.

1. A small ball impacts the ground.
2. A bullet is shot towards an object.
3. A hand squeezes a sponge.
4. A burning ball of paper was thrown into a pile of paper.
5. A burning candle is placed with (wind and rain).
6. A person strikes an ice block with a hammer.
7. Sunlight shines on the water surface, (creating sparkling reflections).
8. Two children of (different weights) are sitting on a seesaw.
9. Pour one liquid into another.
10. Rubber eraser rubs off (pencil) marks on paper.
11. Knife slicing through (butter).
12. Swinging a bat to hit a ball.
13. A boot stomps into a puddle of mud.
14. A ray of light is shining on a wooden block.
15. Flag waving (in the wind) at the top of pole.
16. A broom drags across the (dirty) ceramic floor.
17. After being released, the ball rolls down the slope on its own.
18. A paper airplane is thrown and glides through the air.
19. Drop dye into the water.
20. Sprinkle (iron) filings around a magnet.

We also show some LLM-generated examples of various relationships between variables on the above 20 scenarios. These examples illustrate the diversity and effectiveness of automatic generation.

In the scenario "A hand squeezes a sponge", the LLM identifies key factors like "Sponge is wet", "Hand applies strong grip", and "Hand fully releases the sponge". It generates diverse relationships by considering the states of the objects, the actions, and their sequence. The model recognizes state-based relationships (e.g., wet sponge, strong grip), causal relationships (hand's grip expels water), and temporal relationships (the sequence of squeezing and releasing). Additionally, it captures interaction relationships, where the sponge's wetness and the hand's pressure influence the outcome, such as "Water is expelled".

In the scenario "After being released, the ball rolls down the slope on its own", the LLM identifies factors such as "Is ball on slope", "Is slope steep enough", and "Is path clear of obstacles". The model links the position of the ball and the steepness of the slope to the ball's ability to roll, understanding that the ball will move if both conditions are satisfied. It also incorporates the influence of obstacles, recognizing that any obstruction along the path can prevent the ball from reaching the bottom. The LLM successfully identifies the relationship between the final outcome, "Ball reaches bottom," and the various factors involved, while considering the entire process, including the potential for obstacles to interrupt the ball's descent.

In the scenario "Rubber eraser rubs off (pencil) marks on paper", the LLM identifies factors like "Is pencil mark", "Eraser in contact", and "Rubbing motion present". These factors work together

to determine the outcome, "Pencil mark removed". The model recognizes that the presence of a pencil mark and the eraser's contact are necessary for the process to start. Additionally, the rubbing motion, combined with the eraser's pressure, results in the final outcome of mark removal.

D DETAILS OF CROWD EXPERIMENT

We conducted a crowd experiment to validate our automatic annotation of causal systems based on scenario descriptions. We first invite three undergraduates (2 from physics school and 1 from computer science school) to annotate the same 20 text scenarios. We provide them with the same requirements as we provided to LLM. We first check their annotation with first 5 attempts and then feedback some obvious misalignment with our requirement. We also instructed the annotators to avoid (1) referencing textbooks, as we wanted them to rely on commonsense rather than professional background knowledge, (2) using LLMs or other automatic annotation tools, to ensure their annotations reflected human intuition, and (3) communicating with each other to prevent bias. For human annotators, we prompted them to think in three steps similar to LLM; but we only collected the final rules. In order to ensure the seriousness of the annotators, we took a small number of samples and asked the annotators to explain their annotation reasons, which were checked by the authors. For the purpose of real comparison, we allowed a small number of non-systematic errors or deviations in the annotations — because this reflects the true level of human annotators.

These 60 annotations collected for the 20 scenario will be randomly shuffled together with the 60 annotations generated by LLM and given to five other annotators for scoring. The five annotators were also undergraduates (3 from computer science, 1 from mathematics, and 1 from economics).

The scoring standard we provide is:

- **Requirement**: whether the annotation meets all of our requirements including visibility, binary, and root node independence.
- **Rationality**: whether all the nodes in the causal system are consistent with public knowledge and common; and whether the most important factors and causal relations are included in the annotation.
- **Soundness**: whether all the rules in the causal graph are correct and definitive (from both physics and commonsense).

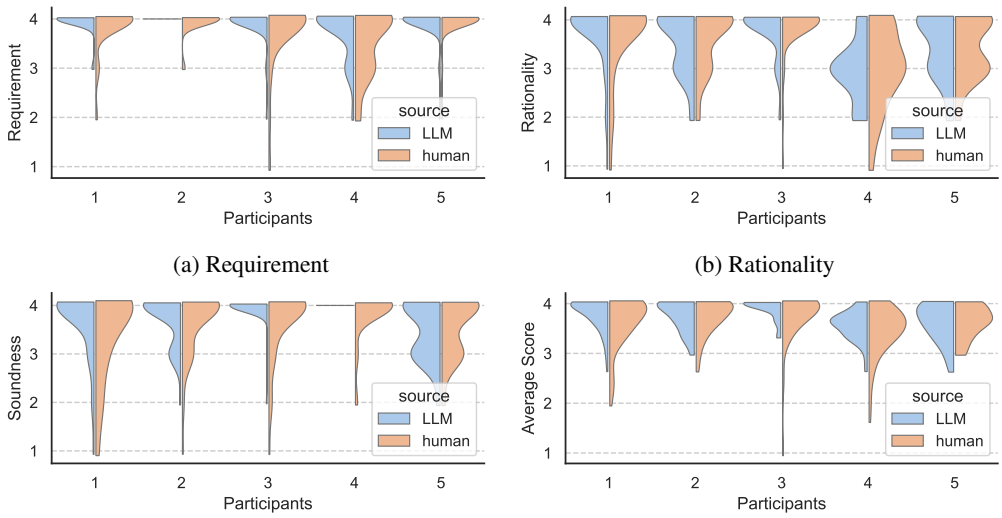
Each criterion is scored on a scale of 1-4, where

- 4: the annotation is completely correct (or meet the requirement),
- 3: there are minor errors,
- 2: there are obvious errors,
- 1: there are essential errors and the annotation needs to be rewritten.

The average scores have shown in Table 1 in the main paper. Here, we provide the detailed distribution of each scorer in Figure 5.

For "requirement" and "soundness", the LLM achieve excellent performance with a larger proportion of scores clustering around the top rating of 4 and the average score is significantly higher than human annotations. For rationality, the LLM- and human-annotation can not be clearly distinguished. The overall tendency of the five raters was consistent. Surprisingly, scorer 2 and 4 gave full marks of 4 points to all 60 items of LLM in requirement and soundness respectively.

Several examples highlight the reasons for the superior performance of the LLM in certain areas. Regarding the Requirement scores, the explicit guidelines provided in the prompt ensured that the LLM annotations generally met the requirements, resulting in consistently high scores. In contrast, human annotators occasionally failed to adhere to these requirements, either due to imprecise expressions or inadvertent oversights. For instance, in the scenario "Pour one liquid into another", one human annotator included the nodes "the densities of the liquids differ greatly" and "the chemical structures of the liquids are similar", both of which are unobservable factors. The LLM, however, avoided such missteps.



(a) Requirement (b) Rationality

(c) Soundness (d) Average

Figure 5: The violin plot as detailed distribution of 5 scorers. The width shows the number of the samples. The x-axis represents the 5 annotators.

In terms of Soundness, where we require that the rules in the causal graph be both correct and definitive, human annotations displayed considerable variability across different scenarios. Some annotations included many nodes and rules, while others were sparse. In cases where a larger number of rules were included, human annotators sometimes overcomplicated their annotations, which led to errors. For example, in the scenario "A bullet is shot towards an object", a human annotator included the rule (A bullet hole appeared on the back of the object)= (The bullet moves quickly) $\wedge$ (The object is hard). The increased complexity of the rule, while addressing multiple factors, led to inaccuracies. The LLM, by contrast, considered fewer factors and produced simpler, more accurate rules.

For the Rationality criterion, which required the inclusion of the most important factors and causal relationships, human annotators excelled in some scenarios but failed to fully account for relevant factors in others. This variability resulted in a broader distribution of scores, with a greater number of high and low ratings for human annotations. Overall, the performance of both human and LLM annotations in this category was similar.

We took great care to ensure that all annotations generated in this experiment adhered to ethical guidelines, ensuring that no violent, pornographic, discriminatory, or offensive content was included in the annotated scenarios. To safeguard against potential ethical violations, we closely monitored the content throughout the annotation process and implemented a strict review mechanism. Additionally, all annotators were explicitly instructed on the importance of maintaining a respectful and non-harmful approach in their work.

In recognition of the effort and time invested by the annotators, they were compensated at a rate of 100 CNY per hour, which is in line with standard industry practices for similar tasks. This compensation not only reflects the value of their contributions but also ensures that the annotators were fairly incentivized for their participation in the study. Furthermore, we provided a feedback loop for annotators, encouraging them to express any concerns or challenges they faced during the annotation process, fostering an open and transparent working environment.

E Details of test pipeline

Here we introduce the details of the test pipeline. For step "prompt generation", see Appendix E.1. For step answer retrieval, see Appendix E.2 and Appendix E.3.

E.1 DETAILS OF TEXT PROMPT GENERATION

Given a causal system as a test case, we need to generate some text prompts, which constrain the variable values in the scenario and are used to prompt the VGMs to generate corresponding videos. (In other words, they are used as the input of the tested T2V models.)

The step can be automated by an LLM. In this paper, we utilize the OpenAI gpt-4o (gpt-4o-2024-08-06) to finish it. To reduce communication overhead, we adopt the strategy of generating first and then sampling from the generated sentences, which is slightly different from the one described in the pipeline. Specifically, we provide the LLM with the original sentence description of the scenario and the list of variables (roots and non-roots separately). We require the model to generate m sentences for every 2^N possible combinations of $\mathbf{X}$ where $N = |\mathbf{X}|$. We find for most situation $N < 5$, the strategy of generating all value combinations at once is effective and works better than generating one value at a time. We observed that the former allows the model to consciously distinguish different values of $\mathbf{X}$. For cases where N is too large, we take the approach of generating one value of $\mathbf{X}$ at a time. In our experiments, we set $m = 10$. In this setting, for each causal system, About 500 tokens are reading and 500 - 1000 tokens are generated by gpt-4o, costing about $0.005.

The prompt we use in the step is shown as follows:

Prompt for sentence generation without results:

> You are a helpful assistant to generate corresponding short description about a scenario given some conditions. You will be provided with a short sentence to describe a scenario as well as some factors (variables) in the scenario. You should generate some short sentences which are slightly different from the originial sentence and describe the situation where the scenario is the same but the corresponding variables take given different value from original situation.
>
> The scenario is: {scenario}
>
> In this scenario, there are some factors are considered as (binary) variables and you should generate new description to change the original scenario to meet the corresponding value.
>
> Factors: {str(factors)}.
>
> There are also some results variables which are the outcome of the above factors: {non_roots}. The values of these variables should not be mentioned in the generated sentences.
>
> Each variable can take value as "yes" or "no" independently so that there are 2**{num_factors} = {num_comb} compositions. You should generate {num_sent} sentences for each yes/no composition for these variables.
>
> Please make sure (1) each sentence meet and explicitly express the corresponding value of variables and (2) the generated sentences as diverse as possible. Notice that you can add, delete or modify some words in original description to get the new sentence.
>
> Your answer should be following the schema provided. Here,
>
> - factors: The names of provides variables.
>
> - compositions: Samples for all compositions. It is a list (len = 2**{num_factors} = {num_comb}) where each element has two parameters:
>
> value: a list of bool. One-to-one correspondence with the values or the variables in the factors list.
>
> samples: a list contains the given number of generated sentences.

Prompt for sentence generation with results:

You are a helpful assistant to generate corresponding short description about a scenario given some conditions. You will be provided with a short sentence to describe a scenario as well as some factors (variables) in the scenario. You should generate some short sentences which are slightly different from the originial sentence and describe the situation where the scenario is the same but the corresponding variables take given different value from original situation.

The scenario is: {scenario}

In this scenario, there are some factors are considered as (binary) variables and you should generate new description to change the original scenario to meet the corresponding value.

Factors: {str(factors)}.

There are also some results variables which are the outcome of the above factors with their expected value: {non_roots}. In each possible composition of factor values, you should first induce the corresponding value of the results variables and then generate the sentences.

In these sentences, please explicitly and clearly express the corresponding value of both the factors and the results variables in the generated sentences. The rules of the results: {"\n".join(rules)}

Each variable can take value as "yes" or "no" independently so that there are 2**{num_factors} = {num_comb} compositions. You should generate {num_sent} sentences for each yes/no composition for these variables.

Please make sure (1) each sentence meet and explicitly express the corresponding value of variables and (2) the generated sentences as diverse as possible. Notice that you can add, delete or modify some words in original description to get the new sentence.

Your answer should be following the schema provided. Here,

- factors: The names of provided factor variables.

- results: The names of provided results variables.

- compositions: Samples for all compositions. It is a list (len = 2**{num_factors} = {num_comb}) where each element has three parameters:

value: a list of bool. One-to-one correspondence with the values or the variables in the factors list.

results: a list of bool. One-to-one correspondence with the values or the variables in the results list. Calculated by the given rules.

samples: a list contains the given number of generated sentences.

E.2 Details of probe question generation

We utilize GPT-4o-mini-2024-07-18 to generate questions for each variable. In a single conversation, we provide a short description of the scenario along with the factors that should be focused on. We instruct the model to generate questions for all root and non-root factors simultaneously. The prompt we design requires the model to generate a simple yes-no question for each factor in the scenario, ensuring that the questions are directly focused on the specific factor without incorporating any assumptions or conditions related to other factors.

The prompt we use in this step is shown as follows:

Prompts for Probe Question Generation:

You are a helpful assistant to help generate some questions about some factors in a scenario. You will be provided with a short description of a scenario and some factors that should be focused on. You should generate **ONE** yes-no questions for **EACH** of the factors in

the scenario. These questions will be used to asked a video language model to test the actual situation in a video about the scenario. Notice that your questions should be simple, clear and direct to the target factor, and should not contain any assumption or conditions about other factors.

The scenario is {scenario}.

The factors are: {factors}.

E.3 Details of answer retrieval

We tested two models to answer questions based on video content: Gemini and OpenAI 4o. Gemini has built-in video reading capabilities, extracting one frame per second for processing. In contrast, OpenAI 4o can process multiple images, so we extract one frame every 10 frames from the video and provide these key frames to the model for question answering. Ultimately, we adopted OpenAI 4o as the primary model for our experiments due to its superior performance.

For each video, we need to ask multiple questions. To ensure that the model relies strictly on the video content rather than commonsense or context, we explored two distinct questioning strategies. The first strategy involves asking one question at a time, ensuring the independence of each answer, though this approach incurs higher costs. The second strategy involves asking all the questions in a single round, within a single prompt. To avoid the model inferring subsequent questions based on prior answers or external commonsense, we topologically sort the nodes in the causal graph, ensuring that result variables are queried before cause variables. This method prevents the model from reasoning through previous answers when addressing subsequent questions. Additionally, we specify in the prompt that the model should answer based solely on the video.

For each question, we allow the model to respond with True, False, or N/A. Some videos suffer from lower generation quality, or fail to align with the textual descriptions, causing critical factors to be unobservable. In these cases, when the video does not provide enough evidence to answer the question, we allow the model to respond with N/A.

The prompt we use in this step is shown as follows:

Prompt for Video Analysis and Question Answering:

You are a professional video analysis expert, specialized in answering questions based on video content. Please answer the following question based **strictly** on the video provided. Ensure that your response is based on the video itself, and not on your own guesses or general knowledge.

You will be provide some yes/no questions related to the video. Your answer should be in “true”, “false” or “N/A”. Besides, you should provide a brief explanation or evidence for your answer.

You should answer “N/A” if:

1. The video quality is too low, or the content is too unclear to make any meaningful inference.

2. The content in the video is not continuous or complete. The temporal and spatial discontinuities in the video make it impossible to make reasonable predictions.

3. The question asks about something that cannot be observed or recognized in the video (e.g., an object, event, or action that is not present).

4. The video does not provide enough context or evidence to form a conclusion.

5. The answer is unclear or could be interpreted in multiple ways, leading to ambiguity.

6. The question asks about an action, and the necessary prior action (for example, the ball hitting the ground before it can bounce) is not observed. Without the prior action, it is impossible to determine if the subsequent event occurred.

if you believe you can answer yes or no with a reasonable degree of confidence, you should not answer "N/A". Especially, if the question asks about whether something is present, or an event has occurred, and the videos shows that it is absent or has not occurred, you should answer "false" instead of "N/A". For these questions, you can answer "N/A" only if the video quality is too low to make a meaningful inference.

If the question asks about an object, and the object is not observed, answer "false". Do not answer "N/A".

For detect an action, you should refer to some continuous frames to make sure the action is happening, instead of just one frame.

In addition, you should judge each question as independently as possible, and do not answer another question based on the content of another question. In particular, the content of another question itself should not be used as the basis for answering the current question.

Based on the above guidelines, please answer the following questions:

"\n".join({questions})

F DETAILED DEFINITION FOR METRICS

In this subsection, we give a detailed definition for our proposed metrics in Section 4.

First we review the definitions and symbols. Let $\mathbf{V}$ be a set of variables representing all factors of interest in a causal system. Let G a directed acyclic graph with node set $\mathbf{V}$ and edge set $\mathbf{E}$. For every $V_j \in \mathbf{V}$, let $pa(V_j) = \{V_k \in \mathbf{V} : V_k \to V_j \in \mathbf{E}\}$ be the set of nodes that has a directed edge pointing to V_j. Suppose there is a deterministic structural equation model over $\mathbf{V}$. That is, for every $V_j \in \mathbf{V}$ such that $pa(V_j) \neq \emptyset$, there exists a function f_j such that $V_j = f_j(pa(V_j))$. Denote $\mathbf{X} = \{V_j \in \mathbf{V} : pa(V_j) = \emptyset\}$ and $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$. We also write $\mathbf{X} = (X_1, X_2, \ldots, X_{m_1})$ and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_{m_2})$ as random vectors. Then $\mathbf{X}$ is called the set of root (or cause) variables, and $\mathbf{Y}$ is called the set of non-root (or outcome) variables. In structural equation $Y_j = f_j(pa(Y_j))$ for every $Y_j \in \mathbf{Y}$, the function f_j is called the rule of Y_j. The structural equations can be equivalently represented as $\mathbf{Y} = f(\mathbf{X})$. Since the value of non-root variables is determined by root variables, we also write $Y_j = f'_j(\mathbf{X})$ for every $Y_j \in \mathbf{Y}$. Let $D(\mathbf{X})$ denote the domain of $\mathbf{X}$, that is, the set of all possible values of $\mathbf{X}$.

In our pipeline, we use a large language model for generating prompt from the given causal system and specified variables, a video generation model for generating video from the prompt, and an multi-modale LLM for retrieving the value of variables from the video. For specified $\mathbf{X}, \mathbf{Y}$, let $f_P(\mathbf{X}, \mathbf{Y})$ denote the generated prompt under the given causal system, with specifying both $\mathbf{X}$ and $\mathbf{Y}$. Let $f_P(\mathbf{X})$ denote the generated prompt under the given causal system with only specifying only $\mathbf{X}$. Note that f_P includes an independent error ε_P implicitly, so it is not a deterministic function of $\mathbf{X}$ and $\mathbf{Y}$. For a prompt P, let $f_V(P)$ denote the video generated by video generation model with prompt P. Finally, let $\hat{\mathbf{X}}, \hat{\mathbf{Y}} = f_A(f_V(P))$ denote the **observation** of all variables from the generated video. For simplicity, we also write $\hat{\mathbf{X}}, \hat{\mathbf{Y}} = f_{VA}(P)$. In this situation, we also call $\mathbf{X}, \mathbf{Y}$ the **ground truth**. For the i-th sample, let $\mathbf{X}_i, \mathbf{Y}_i$ denote the ground truth and $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i$ denote the observation. For any $V \in \mathbf{V}, X \in \mathbf{X}$ and $Y \in \mathbf{Y}$, we use V_i, X_i, Y_i or $\hat{V}_i, \hat{X}_i, \hat{Y}_i$ to denote the corresponding component of $\mathbf{X}_i, \mathbf{Y}_i$ or $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i$, just as we use V, X, Y to denote the corresponding component of $\mathbf{X}, \mathbf{Y}$. We also use X_{ij} to denote the component X_j in vector $\mathbf{X}_i$. For variable $Y_j \in \mathbf{Y}$, we use $\hat{pa}(Y_j)$ to denote the observed value of $pa(Y_j)$.

F.1 TEXT CONSISTENCY

For text consistency, let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n_1}$ be n_1 samples that are i.i.d. are uniform distributed over $D(\mathbf{X})$. Let $\mathbf{Y}_i = f(\mathbf{X}_i)$ for $i = 1, 2, \ldots, n_1$.

Since we have specified the value of every variable in the prompt, we expect that the value of every observed variable matches with its ground truth. However, due to the internal causal mechanism in the video generation model, the value of outcome variables in the video may be influenced by the

value of root variables in the video. Therefore, we propose two versions of metric: s_1^{all} by comparing the observed value of all variables with their ground truth, and s_1^{roots} by comparing the observed value of only root variables with their ground truth. For s_1^{roots}, we generate prompt $P_i = f_P(\mathbf{X}_i)$ by specifying only root variables, and for s_1^{all}, we generate prompt $P_i = f_P(\mathbf{X}_i, \mathbf{Y}_i)$ by specifying both $\mathbf{X}_i$ and $\mathbf{Y}_i$. Finally, we get observation $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i = f_{VA}(P_i)$ by generating video from prompts and asking questions from videos.

The metrics for text consistency is defined as:

$$s_1^{\text{all}} = \frac{1}{n_1(m_1+m_2)} \sum_{i=1}^{n_1} \sum_{V \in \mathbf{V}} \mathbb{1}(V_i = \hat{V}_i), \quad s_1^{\text{roots}} = \frac{1}{n_1 m_1} \sum_{i=1}^{n_1} \sum_{X \in \mathbf{X}} \mathbb{1}(X_i = \hat{X}_i),$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

F.2 Generation consistency

For generation consistency, we construct some groups of samples. Samples within the same group should have the same ground truth. Therefore, by comparing observations within the same group, we can test whether generations for the same ground truth are consistent.

Formally, let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_2}$ be n_2 different values that are randomly selected from $D(\mathbf{X})$. We construct n_2 groups, with r samples in each group, that is, letting

$$\mathbf{X}_1 = \cdots = \mathbf{X}_r = \mathbf{x}_1, \ldots, \mathbf{X}_{(n_2-1)r+1} = \cdots = \mathbf{X}_{n_2 r} = \mathbf{x}_{n_2}.$$

For $i = 1, 2, \ldots, n_2 r$, let $P_i = f_P(\mathbf{X_i})$ be the generated prompt and $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i = f_{VA}(P_i)$ be the observation.

To measure the inconsistency of observations within a group, we propose two versions of metric: s_2^{truth} and s_2^{observe}. For s_2^{truth}, we assume that text consistency holds, that is, observation of root variables should remains the same within each group. Therefore, we compare all variables for each group. For s_2^{observe}, we allow for observation of root variables to be different within each group. Relatively, we see the observed root variables as the truth understood by the video generation model. So we reconstruct the groups by partitioning the samples by $\hat{\mathbf{X}}_i$, and compare the observed outcome variables within each group.

Formally, for an index set $\mathbf{S} \subseteq \{1, 2, \ldots, n_2 r\}$ and variable $V \in \mathbf{V}$, denote $\bar{V}_{\mathbf{S}} = \frac{1}{|\mathbf{S}|} \sum_{i \in \mathbf{S}} \hat{V}_i$ be the mean, and $d(V, \mathbf{S}) = \frac{1}{|\mathbf{S}|} \sum_{i \in \mathbf{S}} \left(\hat{V}_i - \bar{V}_{\mathbf{S}} \right)^2$ be the sample variance of V in subgroup $\mathbf{S}$. For group index $k = 1, 2, \ldots, n_2$, let $\mathbf{S}_k = \{(k-1)r+1, (k-1)r+2, \ldots, kr\}$ be the index of samples within group k. Then we have

$$s_2^{\text{truth}} = \frac{1}{n_2 m_2} \sum_{k=1}^{n_2} \sum_{Y \in \mathbf{Y}} d(Y, \mathbf{S}_k).$$

For definition of s_2^{observe}, for each $\mathbf{x} \in D(\mathbf{X})$, let $\mathbf{S}_{\mathbf{x}} = \{i : \hat{\mathbf{X}}_i = \mathbf{x}\}$, and let $\mathcal{S} = \{\mathbf{S}_{\mathbf{x}} \neq \emptyset : \mathbf{x} \in D(\mathbf{X})\}$. Then we have

$$s_2^{\text{observe}} = \frac{1}{m_2 |\mathcal{S}|} \sum_{Y \in \mathbf{Y}} \sum_{\mathbf{S}_{\mathbf{x}} \in \mathcal{S}} d(Y, \mathbf{S}_{\mathbf{x}}).$$

F.3 Rule consistency

For rule consistency, we generate samples for each outcome variable independently. For each $Y_j \in \mathbf{Y}$, let $\mathbf{S}_j^T = \{\mathbf{x} \in D(\mathbf{X}) : f_j'(\mathbf{x}) = 1\}$ be the set of values of $\mathbf{X}$ that making $Y_j = f_j'(\mathbf{X}) = 1$, and let $\mathbf{S}_j^F = D(\mathbf{X}) \setminus \mathbf{S}_j^T$. Then for ground truth $\mathbf{X}$ and $\mathbf{Y} = f(\mathbf{X})$, we have $Y_j = 1$ if and only if $\mathbf{X} \in \mathbf{S}_j^T$.

To test whether the video generation model has learned this rule, we draw n_3 samples $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n_3}$ uniformly from $\mathbf{S}_j^T$, and n_3 samples $\mathbf{X}_{n_3+1}, \mathbf{X}_{n_3+2}, \ldots, \mathbf{X}_{2n_3}$ uniformly from $\mathbf{S}_j^F$. Comparing to drawing sample uniformly from $D(\mathbf{X})$, this sampling method avoids the bias

that may arise when $|\mathbf{S}_j^T|/|\mathbf{S}_j^F|$ is near 0 or 1. For $i = 1, 2, \ldots, 2n_3$, let $P_i = f_P(\mathbf{X}_i)$ be the generated prompt and $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i = f_{VA}(P_i)$ be the observation.

We also propose two versions of metrics for rule consistency, s_3^{truth} and s_3^{observe}. For s_3^{truth}, we assume that text consistency holds, and check whether the value of observed outcome variables matches its ground truth. For s_3^{observe}, we see the observed parents of each outcome variabe as the truth understood by the video generation model. Therefore, we calculate the value of outcome variables from the rules and its observed parents, and compare them with observed outcome variables. Formally, we have

$$s_3^{\text{truth}}(Y_j) = \frac{1}{2n_3} \sum_{i=1}^{2n_3} \mathbb{1}\left(Y_{ij} = \hat{Y}_{ij}\right), \quad s_3^{\text{truth}} = \frac{1}{m_2} \sum_{Y_j \in \mathbf{Y}} s_3^{\text{truth}}(Y_j).$$

For s_3^{observe}, we propose a strategy to rebalance samples such that the expected value of Y_j, $f_j(\hat{pa}(Y_j))$, has equal weights over $\{0, 1\}$. Therefore, denote $g_j = \sum_{i=1}^{2n_3} f_j(\hat{pa}(Y_j))$ as the total number of samples such that the expected value of Y_j is 1, then we reweight each sample and define $s_3^{\text{observe}}(Y_j)$ as

$$s_3^{\text{observe}}(Y_j) = \frac{1}{2} \sum_{i=1}^{2n_3} \mathbb{1}\left(\hat{Y}_j = f_j(\hat{pa}(Y_j))\right) \left(\frac{f_j(\hat{pa}(Y_j))}{g_j} + \frac{1 - f_j(\hat{pa}(Y_j))}{2n_3 - g_j}\right),$$

$$s_3^{\text{observe}} = \frac{1}{m_2} \sum_{Y_j \in \mathbf{Y}} s_3^{\text{observe}}(Y_j).$$

F.4 Sample strategy for three-level metrics

We propose a unified sampling framework designed to optimize sample efficiency across different evaluation metrics. First, we perform sampling for each metric. Specifically, for Metric 1: text consistency, we collect n_1 samples, where the $\mathbf{X}$ values are uniformly random from the set $D(\mathbf{X}) = \{1, 0\}^{|\mathbf{X}|}$. For Metric 2: generation consistency, we collect n_2 groups, each containing r samples with the same $\mathbf{X}$ value. For Metric 3: rule consistency, for each $Y_j \in \mathbf{Y}$, we collect n_3 samples from the positive set $\mathbf{S}_j^T$ and the negative set $\mathbf{S}_j^F$, respectively. During each sampling step, we record the number of samples corresponding to different $\mathbf{X}$.

With the separate sampling results, we construct a total sample set, where for each possible $\mathbf{X}$ value, the sample count is the **maximum** across the three metrics. While each sample may be used multiple times to compute different metrics or different rule accuracies for Y_j, within the same metrics (or within metric 3 for the same Y_j), each sample is used only once. The framework ensures that no sample is reused within the calculation of any single metric. By doing so, we maintain the independent and identically distributed (IID) conditions for sampling, while preserving the integrity of each metric's evaluation criteria. The architecture also achieves significant storage efficiency, reducing redundancy compared to traditional independent sampling approaches, without compromising the statistical validity of the results. Finally, we use the total sample set to select the corresponding text prompts and generate videos.

In our benchmark, we set the parameters as follows: $n_1 = 10$, $n_2 = 5$, $n_3 = 10$, and $r = 3$. Using these values, we apply our strategy to draw samples. Appendix H.4 demonstrate that this sample size is sufficient for distinguishing between metrics across different models. Specifically, we draw n_1 samples for the evaluation of text consistency, $n_2 r$ samples for the evaluation of generation consistency, and $2n_3|\mathbf{Y}|$ samples for the evaluation of rule consistency. In contrast, without this strategy, a total of $N = 25 + 20|\mathbf{Y}|$ samples would be required for each causal system, which could significantly increase computational costs. The distribution of sample sizes for each causal system is depicted in Figure 6, which illustrates a considerable reduction in the number of samples needed by our approach.

F.5 Sample-based scores

Our metrics can also be applied to each sample, showing how each sample contributes to the evaluation. The definitions are as follows.

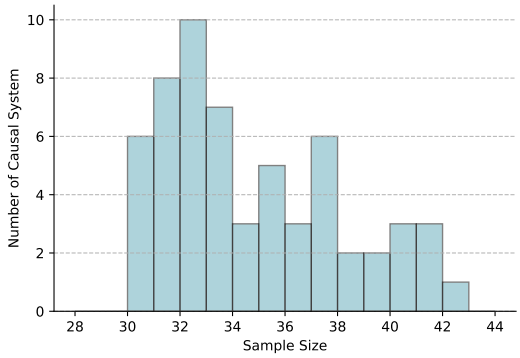


Figure 6: Distribution of sample sizes over causal systems.

For text consistency, the metrics for a sample i are defined as

$$s_{1,i}^{\text{all}} = \frac{1}{m_1 + m_2} \sum_{V \in \mathbf{V}} \mathbb{1}(V_i = \hat{V}_i), \quad s_{1,i}^{\text{roots}} = \frac{1}{m_1} \sum_{X \in \mathbf{X}} \mathbb{1}(X_i = \hat{X}_i).$$

For the sake of sample efficiency, samples with same ground truth $\mathbf{X}$ are reused in testing generation consistency and rule consistency. Let i be the index of a sample in a group $\mathbf{S}$. Similarly, let $\bar{V}_{\mathbf{S}} = \frac{1}{|\mathbf{S}|} \sum_{i \in \mathbf{S}} \hat{V}_i$ be the mean of observed values for variable $V \in \mathbf{V}$. Then the metrics for generation consistency on sample i is defined as

$$s_{2,i}^{\text{truth}} = \frac{1}{m_2} \sum_{Y \in \mathbf{Y}} (\hat{Y}_i - \bar{Y}_{\mathbf{S}_k})^2, \quad s_{2,i}^{\text{observe}} = \frac{1}{m_2} \sum_{Y \in \mathbf{Y}} (\hat{Y}_i - \bar{Y}_{\mathbf{S}_\mathbf{x}})^2,$$

where $\mathbf{S}_k$ and $\mathbf{S}_\mathbf{x}$, as defined in Appendix F.2, are groups which contains sample i.

For rule consistency, samples are reused so that some samples are contained in the test samples for multiple outcome variables. Let i be the index of a sample. Write $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_{m_2})$, and let $\mathbf{Z}_i \subseteq \{1, 2, \ldots, m_2\}$ be the index of all outcome variables whose test samples contains sample i. Then metrics for sample i are

$$s_{3,i}^{\text{truth}} = \frac{1}{|\mathbf{Z}_i|} \sum_{j \in \mathbf{Z}_i} \mathbb{1}\left(Y_{ij} = \hat{Y}_{ij}\right),$$

$$s_{3,i}^{\text{observe}} = \frac{1}{T_i} \sum_{j \in \mathbf{Z}_i} n_3 \mathbb{1}\left(\hat{Y}_j = f_j(\hat{pa}(Y_j))\right) \left(\frac{f_j(\hat{pa}(Y_j))}{g_j} + \frac{1 - f_j(\hat{pa}(Y_j))}{2n_3 - g_j}\right),$$

where

$$T_i = \sum_{j \in \mathbf{Z}_i} n_3 \left(\frac{f_j(\hat{pa}(Y_j))}{g_j} + \frac{1 - f_j(\hat{pa}(Y_j))}{2n_3 - g_j}\right).$$

G MANUAL VERIFICATION OF AUTOMATIC RESULTS

For automatic annotation of causal systems, we have verified the effectiveness through crowd experiments. Here we verify other automatic steps, including:

- Section G.1: generating text prompts from value combinations,
- Section G.2: generation probe questions from factors,
- Section G.3: retrieve observed value from videos.

For each step, we randomly choice some automatic generation in our 60 test cases from our VACT benchmark and manually check whether the automatic annotation is correct.

G.1 MANUAL VERIFICATION OF PROMPT CORRECTNESS

Below are randomly selected scenarios and corresponding prompts from our dataset. We manually verify the correctness by checking whether the two types of prompts (with and without non-root nodes) match the given values for the variables.

We examined a total of four scenarios and their corresponding 106 prompts. Nearly all of the prompts passed inspection, with the exception of **two**. The issue with the first prompt arises from our setting the variable "*sponge is wet*" to false, while the prompt only specifies that the hand is dry and fails to clarify the condition of the sponge. The second issue pertains to a prompt that was expected to contain only root nodes; however, it includes the word "*slide*", which introduces a non-root value.

We show these 106 samples as follows, where we marked the correct ones with ✓ and the 2 incorrect results with ×.

Scenario:

Rules for each non-root node:

Pile Catches Fire = (Ball Actively Burning ∧ Ball Contact Pile)

X_1 = Ball Actively Burning X_2 = Ball Contact Pile Y = Pile Catches Fire

Prompts:

Without non-root nodes:

(X_1 = False , X_2 = True)

A smoldering paper ball, now unlit, was tossed but missed the stack of old newspapers.✓

A ball of paper that had stopped burning eventually settled into a stack of paper.✓

A cooling ball of paper was placed carelessly into a mound of papers.✓

An extinguished paper sphere was accidentally dropped into a heap of documents. ✓

A barely expired paper ball softly landed in a collection of scraps. ✓

(X_1 = True , X_2 = True)

A flaming ball of paper crashed into a stack of old newspapers. ✓

A lit paper ball was hurled into a heap of documents.✓

A burning paper sphere landed directly in a pile of loose-leaf papers. ✓

A fireball of paper was tossed straight into a mound of papers. ✓

An ignited ball of paper rolled into a collection of scraps. ✓

(X_1 = True , X_2 = False)

A burning ball of paper was thrown close to but missed hitting a pile of paper. ✓

A flaming ball of paper flew past a stack of old newspapers without making contact. ✓

A lit paper ball was launched near a heap of documents, but it didn't touch them. ✓

(X_1 = False , X_2 = False)

A ball of paper, which had extinguished, was thrown away from a pile of paper.✓

A smoldering paper ball, now unlit, was tossed but missed the stack of old newspapers. ✓

A once aflame ball of paper, now out, was hurled and did not touch the paper heap. ✓

With non-root nodes:

(X_1 = False , X_2 = False, Y = False)

An unlit ball of paper passed by the paper pile without touching it, leaving the pile unburned.✓

(X_1 = False , X_2 = True, Y = False)

Since the ball wasn't on fire upon contact, the paper pile stayed unharmed.✓

A non-burning, thrown ball of paper landed on the pile but didn't ignite it. ✓

Though the ball reached the pile, it was not burning, and thus the pile remained safe. ✓

A ball that wasn't actively burning was thrown onto the pile, and the pile stayed unignited. ✓

The paper ball made contact with the pile, but without being on fire, the pile did not catch alight. ✓

(X_1 = True , X_2 = True, Y = True)

The flaming paper sphere, still ablaze, was thrown and hit the paper pile, which then caught fire.✓

A blazing ball of paper made contact with a stack of paper, causing the pile to ignite, since the ball was burning and it struck the pile.✓

(X_1 = True , X_2 = False, Y = False)

Despite being actively on fire, the paper ball missed the pile, and as a result, the pile did not catch fire. ✓

Even though the ball was burning, it did not make contact with the pile of paper, so the pile remained unburned. ✓

Scenario:

Rules for each non-root node:

Butter Sliced = (Butter Solid $\wedge$ Downward Slicing Motion Applied)

X_1 = Butter Solid X_2 = Downward Slicing Motion Applied Y = Butter Sliced

Prompts:

Without non-root nodes:

(X_1 = False , X_2 = True)

A knife pierces the soft butter with effortless downward motion.✓

The knife sweeps downward, slicing perfectly through softened butter.✓

Swiftly moving downwards, the knife glides through the creamy butter easily. ✓

(X_1 = True , X_2 = True)

A knife slides effortlessly downward through a solid block of butter. ✓

The solid butter yields smoothly as a knife slices through it with a downward motion.✓

With a straight down slice, the knife cuts cleanly through the solid butter. ✓

Cutting a solid piece of butter with a knife moving downward feels like slicing through soft clay.✓

A sturdy push downward sends the knife through the solidified butter seamlessly. ✓

(X_1 = True , X_2 = False)

Simply pressing a knife against the solid butter won't cut it.✓

The knife doesn't glide through the solid butter without a downward push. ✓

A knife pressed horizontally against the solid butter fails to cut through.✓

(X_1 = False , X_2 = False)

A knife resting on soft butter is ineffective without downward force. ✓

No downward motion makes the knife linger atop the soft butter. ✓

Simply resting a knife on soft butter won't achieve a cut. ✓

Without cutting downward, a knife barely breaks the soft butter surface. ✓

The knife sits idle against the soft butter, lacking downward pressure. ✓

With non-root nodes:

(X_1 = False , X_2 = False, Y = False)

There is no slicing of the butter, as it is neither solid nor subjected to a downward motion. ✓

With the butter not solid and without a downward motion, no slicing occurs. ✓

Neither solid state nor downward motion is present, leaving the butter unsliced.✓

The butter is not solid, and no downward slicing motion is applied, so the butter is not sliced. ✓

(X_1 = True , X_2 = True, Y = True)

Since the butter is solid and a downward force is used, the knife slices the butter.✓

Solid butter is easily sliced through as a downward slicing motion is applied.✓

The butter, being solid, is sliced through as a downward slicing motion is applied. ✓

The butter is solid, and a downward slicing motion is applied, resulting in the butter being sliced. ✓

With the butter in a solid state and a downward slicing motion in action, the butter gets sliced. ✓

(X_1 = True , X_2 = False, Y = False)

The butter is solid but no downward slicing motion is applied, so the butter is not sliced.✓

Scenario:

Rules for each non-root node:

Water Emerges from Sponge = (Sponge is Wet $\wedge$ Hand Fully Compresses Sponge)

Sponge Shape Visibly Changes = (Hand Fully Compresses Sponge)

X_1 = Sponge is Wet X_2 = Hand Fully Compresses Sponge

Y_1 = Water Emerges from Sponge Y_2 = Sponge Shape Visibly Changes

Prompts:

Without non-root nodes:

(X_1 = False , X_2 = True)

A dry sponge is entirely compressed by a hand squeezing it.✓

The hand fully compresses a dry sponge with its grip.✓

A hand squeezes a dry sponge until it's fully compressed. ✓

Fully closing, a hand compresses a dry sponge.✓

The hand squeezes a dry sponge as much as it will go. ✓

(X_1 = True , X_2 = True)

The hand squeezes a wet sponge, fully compressing it. ✓

A hand grips a wet sponge and fully squeezes it.✓

The hand exerts force on a wet sponge, squeezing it flat.✓

A wet sponge is completely compressed by a hand. ✓

A wet sponge is gripped and fully squeezed by a hand. ✓

(X_1 = True , X_2 = False)

Squeezing a wet sponge, the hand stops before fully compressing it.✓

A hand grips a wet sponge, compressing it only slightly. ✓

The hand applies pressure but doesn't fully squeeze the wet sponge.✓

A hand gently squeezes a wet sponge without fully compressing it. ✓

A wet sponge is partially squeezed by a hand. ✓

(X_1 = False , X_2 = False)

The hand applies some pressure to a dry sponge but doesn't compress it completely.✓

A hand holds and gently squeezes a dry sponge without full compression. ✓

The hand grips and squeezes a dry sponge lightly, without full compression. ✓

A hand partially squeezes a dry sponge without complete compression. ✓

With non-root nodes:

(X_1 = False , X_2 = False, Y_1 = False, Y_2 = False)

The dry sponge remains unchanged when the hand gives it a gentle squeeze both in terms of shape and water release. ✓

(X_1 = True , X_2 = True, Y = True, Y_2 = True)

When the hand squeezes the wet sponge completely, the sponge visibly deforms and water emerges.✓

With a wet sponge being fully pressed by the hand, water seeps out and the sponge's form changes.✓

The wet sponge is fully compressed by the hand, resulting in a change in its shape and water being squeezed out. ✓

(X_1 = False , X_2 = True, Y_1 = False , Y_2 = True)

A dry hand compresses the sponge completely, causing its shape to change, but no water releases.×

(X_1 = True , X_2 = False, Y_1 = False , Y_2 = False)

The damp sponge is only partially squeezed by the hand, meaning no water is released and the shape remains consistent.✓

A hand lightly squeezes the wet sponge, leaving its shape and water content unchanged.✓

Although the sponge is wet, the hand does not fully compress it, so no water comes out, and its shape stays the same. ✓

Scenario:

Rules for each non-root node:

Ice Block Moves = (¬ Ice Block On Stable Surface)

Ice Block Cracks = (Hammer Head Metal)

X_1 = Ice Block On Stable Surface X_2 = Hammer Head Metal

Y_1 = Ice Block Moves Y_2 = Ice Block Cracks

Prompts:

Without non-root nodes:

(X_1 = False , X_2 = True)

A person strikes an ice block with a metal hammer, causing it to slide on the surface.×

A metal-headed hammer is wielded by a person to hit an ice block that's not stably placed.✓

Someone hits a sliding ice block with a metal hammer. ✓

An individual uses a metal hammer to strike an ice block that isn't on stable footing. ✓

(X_1 = True , X_2 = True)

A person uses a metal-headed hammer to hit an ice block resting on a stable base. ✓

An individual strikes a stable ice block with a metallic hammer.✓

A hammer with a metal head is used by a person to hit a stable ice block.✓

An ice block on a stable platform is struck by someone wielding a metal hammer. ✓

(X_1 = True , X_2 = False)

An individual hits a secure ice block with a hammer that lacks a metal head. ✓

Someone uses a non-metallic hammer to hit an ice block resting stably.✓

A person uses a hammer with a non-metal head to hit an ice block on a stable surface.✓

Striking a solidly placed ice block with a hammer that doesn't have a metal head. ✓

(X_1 = False , X_2 = False)

A person hits an ice block with a non-metal hammer, and the block is not stable.✓

Striking a shifting ice block with a hammer that has a non-metal head. ✓

The hand fully compresses a dry sponge with its grip.Using a non-metal headed hammer, a person hits an unsteady block of ice.✓

The ice block, not secure, is struck by a person with a non-metal hammer. ✓

Someone uses a hammer without a metal head to hit a loosely sitting ice block. ✓

With non-root nodes:

(X_1 = True , X_2 = True, Y_1 = False, Y_2 = True)

The ice block, resting securely on a stable surface, is struck by a hammer with a metal head, which causes it to crack. ✓

(X_1 = False , X_2 = False, Y = True, Y_2 = False)

The ice block on an unsteady surface moves but does not crack when struck with a non-metal hammer. ✓

Even on an unsteady surface, the ice block only shifts without cracking when hit by a non-metal hammer.✓

A non-metal hammer causes the ice block on an unstable surface to move but avoids cracking. ✓

An ice block shifts on its unstable foundation, though uncracked, under a non-metal hammer blow. ✓

(X_1 = True , X_2 = False, Y_1 = False , Y_2 = False)

The ice block, placed securely on a stable surface, does not move or crack when struck by a non-metal hammer.✓

Striking the ice block on a stable foundation with a non-metal hammer results in no movement or cracking.✓

A hammer with a non-metal head hits an ice block on stable ground, neither moves nor cracks it.✓

The ice block, secured by its stable surface, withstands the non-metal hammer blow without cracking or shifting. ✓

A non-metal hammer strikes the ice block on stable ground, leaving it neither cracked nor moved. ✓

G.2 Manual Verification of Factor-Question Alignment

To evaluate whether the generated videos comply with causal rules, we utilize a VLLM to extract the values of both root and non-root nodes. When posing "yes-no" questions about the video, it is essential to ensure that the questions are appropriately aligned with the relevant factors in each specific scenario.

In this section, We randomly selected 7 scenarios, comprising a total of 23 factor-question pairs, all of which were found to be correct.

Scenario: A small ball impacts the ground.

Factor	Question	Correctness
ball is deflated	Is the ball deflated?	✓
the ground is soft	Is the ground soft?	✓
ball bounces	Does the ball bounce?	✓

Scenario: Sunlight shines on the water surface, creating sparkling reflections.

Factor	Question	Correctness
direct sunlight present	Is direct sunlight present on the water surface?	✓
water ripples visible	Are water ripples visible on the surface?	✓
unobstructed water surface	Is the water surface unobstructed?	✓

Scenario: A person strikes an ice block with a hammer.

Factor	Question	Correctness
block is small	Is the ice block small?	✓
direct hammer strike	Is the hammer striking the ice block directly?	✓
block breaks	Does the ice block break when struck?	✓

Scenario: Flag waving in the wind at the top of pole.

Factor	Question	Correctness
is flag hoisted	Is the flag hoisted at the top of the pole?	✓
is there wind	Is there wind present in the environment?	✓
flag waving	Is the flag waving?	✓

Scenario: Flag waving in the wind at the top of pole.

Factor	Question	Correctness
is flag hoisted	Is the flag hoisted at the top of the pole?	✓
is there wind	Is there wind present in the environment?	✓
flag waving	Is the flag waving?	✓

Scenario: A broom drags across the dirty ceramic floor.

Factor	Question	Correctness
broom bristles contact floor	Are the broom bristles making contact with the floor?	✓
floor is wet	Is the floor wet?	✓
obstruction on floor	Is there an obstruction on the floor?	✓
floor becomes clean	Does the floor become clean after using the broom?	✓

Scenario: Drop dye into the water.

Factor	Question	Correctness
dye is water soluble	Is the dye water soluble?	✓
water is stirred	Is the water stirred?	✓
water becomes colored	Does the water become colored?	✓
water becomes uniformly colored	Does the water become uniformly colored?	✓

G.3 MANUAL VERIFICATION OF VLLM ANSWER RETRIEVAL CORRECTNESS

The answers provided by VLLM serve as the foundation for calculating the final score of the generated videos. Therefore, it is essential to manually verify the accuracy of these responses. In this section, we select four models and examine three distinct scenarios, each accompanied by three corresponding prompts.

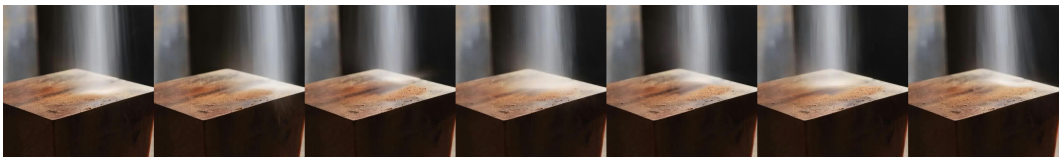
The sampled scenarios encompass both challenging and easy prompts, with and without non-root nodes, and feature answers classified as True, False, or N/A. A comprehensive explanation of the conditions under which VLLM provides an N/A response is available in H.3. For example, the explanation provided by VLLM for the N/A response regarding a video generated by Pika, as presented in Table 8, is: "*The images do not provide a clear view of the top of the boot. It is not possible to determine if it is sealed or not from the given angles.*" for the factor "*boot top sealed*", which is consistent with our observations.

Regarding the accuracy of model responses, we find that VLLM demonstrates sufficient capability to handle simple scenarios and prompts (such as those in Table 5, Table 7,Table 8,Table 9,Table 10, and Table 11). However, its performance declines when addressing more complex questions (such as those in Table 3, Table 4, and Table 6). Currently, the accuracy of this approach hovers around 95%, which is acceptable but still leaves room for improvement. The shortcomings in correctness primarily stem from two factors. First, the VGMs often generate videos with low quality and ambiguity, which increases the difficulty for VLLM to provide accurate answers. Additionally, VLLMs still lack the ability to clearly understand intricate details in images or videos, particularly when dealing with more complex questions. Nevertheless, we are optimistic that as the foundational capabilities of VLLMs continue to improve, the performance of this video description system will experience significant enhancement.

The checked question-answer pairs are shown below, accompanied by the generated videos. Our verification results are presented in a table that closely follows each prompt.

Scenario: A ray of light is shining on a wooden block.

Prompt-1:A beam of light grazes the polished surface of a wooden block in dust. (Videos:Figure 7;Results: Tabel 3)



(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

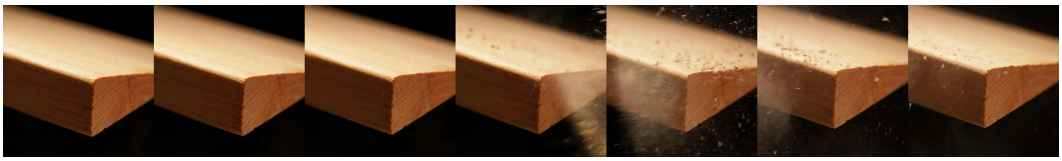
(d) Pyramid Generation

Figure 7: Model Generation

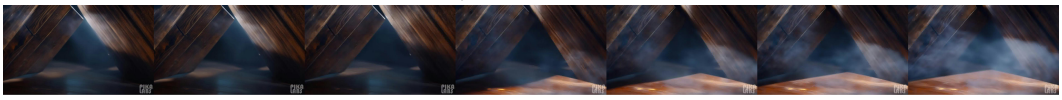
Model	Factor	Model Answer	Correctness
HunyuanVideo	in direct path	True	✓
	surface polished	False	×
	environment dusty	False	✓
	block illuminated	True	✓
	reflection visible	False	✓
	beam visible in air	True	✓
Pika	in direct path	False	✓
	surface polished	True	✓
	environment dusty	False	✓
	block illuminated	False	✓
	reflection visible	False	✓
	beam visible in air	False	✓
Hailuo	in direct path	N/A	✓
	surface polished	False	✓
	environment dusty	False	✓
	block illuminated	True	✓
	reflection visible	True	×
	beam visible in air	False	✓
Pyramid	in direct path	True	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	True	✓
	reflection visible	False	✓
	beam visible in air	True	✓

Table 3: Verification of VLLM Answer Correctness

Prompt-2:The polished surface of a wooden block directly catches the light amid dust. (Videos:Figure 8;Results: Tabel 4)



(a) HunyuanVideo Generation



(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

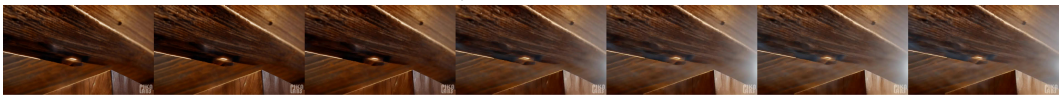
Figure 8: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	in direct path	False	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	False	✓
	reflection visible	False	✓
	beam visible in air	False	✓
Pika	in direct path	False	✓
	surface polished	False	×
	environment dusty	True	✓
	block illuminated	False	✓
	reflection visible	False	✓
	beam visible in air	True	✓
Hailuo	in direct path	False	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	False	✓
	reflection visible	False	✓
	beam visible in air	False	✓
Pyramid	in direct path	True	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	True	✓
	reflection visible	True	×
	beam visible in air	False	✓

Table 4: Verification of VLLM Answer Correctness

Prompt-3:A ray of light directly illuminates a polished wooden block and the environment is dusty, causing both the block to be lit and reflections to be visible, with the beam clearly seen in the air. (Videos:Figure 9;Results: Tabel 5)

(a) HunyuanVideo Generation



(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

Figure 9: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	in direct path	True	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	True	✓
	reflection visible	False	✓
	beam visible in air	True	✓
Pika	in direct path	True	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	False	✓
	reflection visible	False	✓
	beam visible in air	True	✓
Hailuo	in direct path	True	✓
	surface polished	True	✓
	environment dusty	False	✓
	block illuminated	True	✓
	reflection visible	False	✓
	beam visible in air	True	✓
Pyramid	in direct path	True	✓
	surface polished	False	✓
	environment dusty	True	✓
	block illuminated	True	✓
	reflection visible	False	✓
	beam visible in air	True	✓

Table 5: Verification of VLLM Answer Correctness

Scenario: A boot stomps into a puddle of mud.

Prompt-1:An intense stomp by an open-topped boot into a puddle of watery mud occurs. (Videos:Figure 10;Results: Tabel 6)

(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

Figure 10: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	False	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓
Pika	watery mud	True	✓
	big downward stomp	False	✓
	boot top sealed	True	×
	mud splashes out of puddle	False	✓
	mud enters the boot	False	✓
Hailuo	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	N/A	✓
	mud splashes out of puddle	True	×
	mud enters the boot	False	✓
Pyramid	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	False	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓

Table 6: Verification of VLLM Answer Correctness

Prompt-2:In non-watery mud, no splashes occur, but mud enters an unsealed boot during light stepping. (Videos:Figure 11;Results: Tabel 7)

(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

Figure 11: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	watery mud	True	✓
	big downward stomp	False	✓
	boot top sealed	False	✓
	mud splashes out of puddle	False	✓
	mud enters the boot	False	✓
Pika	watery mud	True	✓
	big downward stomp	False	✓
	boot top sealed	True	✓
	mud splashes out of puddle	False	✓
	mud enters the boot	False	✓
Hailuo	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	True	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓
Pyramid	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	False	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	N/A	✓

Table 7: Verification of VLLM Answer Correctness

Prompt-3:A boot with a sealed top makes a big downward stomp into watery mud, causing mud to splash out of the puddle but none enters the boot. (Videos:Figure 12;Results: Tabel 8)

(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

Figure 12: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	True	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓
Pika	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	N/A	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	N/A	✓
Hailuo	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	N/A	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓
Pyramid	watery mud	True	✓
	big downward stomp	True	✓
	boot top sealed	True	✓
	mud splashes out of puddle	True	✓
	mud enters the boot	False	✓

Table 8: Verification of VLLM Answer Correctness

Scenario: Knife slicing through butter.

Prompt-1:The knife meets little opposition as it slices through the butter. (Videos:Figure 13;Results: Tabel 9)

(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

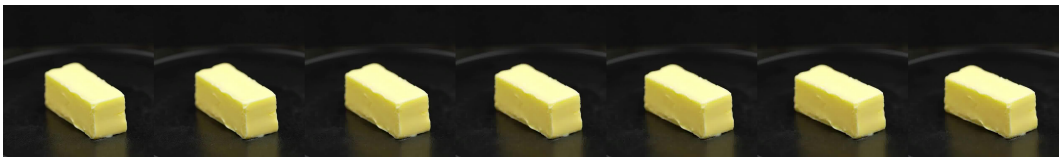
(d) Pyramid Generation

Figure 13: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	blade in contact with butter	True	✓
	Knife is moving against butter	True	✓
	Butter is sliced	True	✓
Pika	blade in contact with butter	True	✓
	Knife is moving against butter	True	✓
	Butter is sliced	True	✓
Hailuo	blade in contact with butter	True	✓
	Knife is moving against butter	True	✓
	Butter is sliced	True	✓
Pyramid	blade in contact with butter	True	✓
	Knife is moving against butter	True	✓
	Butter is sliced	False	✓

Table 9: Verification of VLLM Answer Correctness

Prompt-2:With no movement or contact, the butter sits undisturbed. (Videos:Figure 14;Results: Tabel 10)



(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

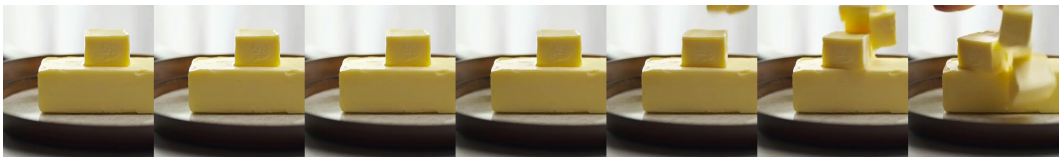
(d) Pyramid Generation

Figure 14: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓
Pika	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓
Hailuo	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓
Pyramid	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓

Table 10: Verification of VLLM Answer Correctness

Prompt-3:Contact with the butter is established, but without motion, the butter remains unsliced. (Videos:Figure 15;Results: Tabel 11)



(a) HunyuanVideo Generation

(b) Pika Generation

(c) Hailuo Generation

(d) Pyramid Generation

Figure 15: Model Generation

Model	Factor	Model Answer	Correctness
HunyuanVideo	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓
Pika	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓
Hailuo	blade in contact with butter	True	✓
	Knife is moving against butter	True	✓
	Butter is sliced	True	✓
Pyramid	blade in contact with butter	False	✓
	Knife is moving against butter	False	✓
	Butter is sliced	False	✓

Table 11: Verification of VLLM Answer Correctness

H Details and more discussion about benchmarks

H.1 Evaluated Models

To conduct a comprehensive benchmark, we evaluate a total of 6 open-source models and 4 closed-source models. Detailed information about the models included in our evaluation is provided in this section.

Open-Source Models:

For the open-source models, we benchmark

- CogVideoX (Hong et al., 2022), a recent state-of-the-art video generation model. Specifically, we use three versions in our experiment: CogVideoX1.5-5B, CogVideoX-5B, and CogVideoX-2B;
- VideoCrafter2 (Chen et al., 2024), the latest version of the VideoCrafter series, which is an open-source toolbox for video generation and editing;
- Pyramid Flow miniFLUX (Jin et al., 2024), utilizing its 768p checkpoint. This variant of the Pyramid Flow series supports the generation of both high-quality images and videos;
- HunyuanVideo (Tencent, 2025), developed by Tencent. HunyuanVideo is currently the largest open-source video generation model, with over 13 billion parameters, and provides performance comparable to leading closed-source models.

All of the open-source models used in our experiments were downloaded from the Huggingface website.

Close-Source Models:

For the close-source models, we benchmark

- Gen-3 Alpha (Runway, 2024), The latest version released by Runway shows improvements in fidelity, consistency, and motion compared to Gen-2;
- Pika (Pika, 2024), developed by Pika Labs, is used in its free beta version, accessed through the Pika Discord Bot;
- Hailuo (MiniMax, 2024), developed by MiniMax, is used in its T2V-01 version;
- Kling 1.0 (Kuaishou, 2024), a closed VGM released by Kuaishou.

We access all the closed-source models by calling their APIs, either through their official websites or third-party interfaces. Detailed information can be found in H.2. Some of the models provide an additional prompt enhancement trick but for fair comparison, we do not turn it on if there is an option. See discussion about this trick in Appendix J.

H.2 COST OF BENCHMARKING

We report the time and money cost of benchmarking each model here.

Open-Source Models:

Name	Device	Time / Video	Total Time (above 2000 videos)
CogVideoX1.5-5B	NVIDIA A800-SXM4-80GB	$\sim$ 15min	$\sim$ 500 GPU hours
CogVideoX-5B	NVIDIA A800-SXM4-80GB	$\sim$ 3min	$\sim$ 100 GPU hours
CogVideoX-2B	NVIDIA A800-SXM4-80GB	$\sim$ 1min	$\sim$ 33 GPU hours
Pyramid Flow	NVIDIA A800-SXM4-80GB	$\sim$ 2.5min	$\sim$ 83 GPU hours
HunyuanVideo	NVIDIA A800-SXM4-80GB	$\sim$ 10min	$\sim$ 330 GPU hours
VideoCrafter2	NVIDIA A800-SXM4-80GB	$\sim$ 3min	$\sim$ 100 GPU hours

Close-Source Models:

Name	API Source	Cost / Video	Total Cost (above 2000 videos)
Gen-3 Alpha	Useapi.net	Unlimited Subscription	$ 95
Pika	Useapi.net	Pika Discord Bot	Free
Kling	PiAPI	$ 0.13	$ 260
Hailuo	Official	Unlimited Subscription	$ 94.99

H.3 ABOUT N/A RESULTS

When we retrieve the observed values in a video by a VLLM, we allow the model to answer 'N/A' besides yes or no. We prompt the model the conditions of answering N/A as follows:

> 1. The video quality is too low, or the content is too unclear to make any meaningful inference.
>
> 2. The content in the video is not continuous or complete. The temporal and spatial discontinuities in the video make it impossible to make reasonable predictions.
>
> 3. The question asks about something that cannot be observed or recognized in the video (e.g., an object, event, or action that is not present).
>
> 4. The video does not provide enough context or evidence to form a conclusion.
>
> 5. The answer is unclear or could be interpreted in multiple ways, leading to ambiguity.
>
> 6. The question asks about an action, and the necessary prior action (for example, the ball hitting the ground before it can bounce) is not observed. Without the prior action, it is impossible to determine if the subsequent event occurred.

We report the N/A ratio in all observation in Table 2 and we also report the 'N/A : correct : incorrect' ratio for all test we used in Level 1 all s_1^{all} in Table 12.

We acknowledge that the appearance of N/A may introduce some bias to subsequent metrics. For example, if the model generates N/A in scenarios where it performs poorly, removing these N/A responses could lead to inflated scores. This would make the model appear better than it actually is, or falsely narrow the performance gap between different models. But as we mentioned in the introduction (Section 1), as a longer-term goal, our evaluation focuses more on the evaluation of the "world simulator", and the guarantee of general video generation quality should be taken as a prerequisite rather than the focus of this article. At the same time, we observe that better (newer, larger) models tend to have a lower N/A ratio, which is in line with our expectations and shows that as the model generation capability continues to improve, the probability of obvious serious errors will gradually decrease.

Table 12: The ratio of N/A variables, correct variables and incorrect variables for text consistency.

Name	N/A ratio	correct ratio	incorrect ratio
CogVideoX1.5-5B	.06	.53	.41
CogVideoX-5B	.06	.55	.39
CogVideoX-2B	.08	.52	.40
VideoCrafter2	.14	.48	.39
Pyramid Flow	.10	.51	.39
HunyuanVideo	.07	.55	.39
Pika	.11	.52	.38
Hailuo	.07	.55	.38
Gen-3 Alpha	.07	.60	.33
Kling	.07	.58	.35

H.4 Experiment for sample size

We conduct an empirical study to determine the minimum sample size required for statistically distinguishing performance metrics between two video generation models (VGMs). The experiment compares CogVideoX-2B (representing open-source models) and Pika (representing closed-source models) under a specific causal system where both models exhibited competent video generation quality. We vary sample sizes from 2 to 100 for text consistency, group sizes from 2 to 16 for generation consistency, and sample sizes for each outcome variable from 2 to 50 samples for rule consistency. To ensure statistical validity, we employ bootstrap resampling (1,000 iterations) with finite-population correction to estimate standard deviations of metric estimators. Standard deviations are adjusted for matching our scenario pool (60 causal systems). For text consistency metrics, we implement two evaluation protocols: 1) excluding missing (N/A) observations, and 2) treating N/A values as incorrect responses. Confidence intervals (95% coverage) are constructed using bias-corrected accelerated bootstrap methods centered on the minimum-variance unbiased estimator.

The results, visualized in Figure 16, reveal distinct sample size requirements across metrics. As a efficiency-accuracy trade-off, we established an operational criterion where the minimal sufficient sample size occurs when the confidence interval of one model's metric no longer overlaps with the point estimate of the competitor model. From the figure we can see that:

- For text consistency, drawing $n_1 = 10$ samples is enough to distinguish metrics between two models in most cases. When N/A observed variables are seen as incorrect, s_1^{all} between two models cannot be distinguished for any number of samples.
- For generation consistency, drawing $n_2 = 5$ groups can distinguish metrics between two models.
- For rule consistency, drawing $n_3 = 10$ samples for each outcome variable can distinguish metrics between two models.

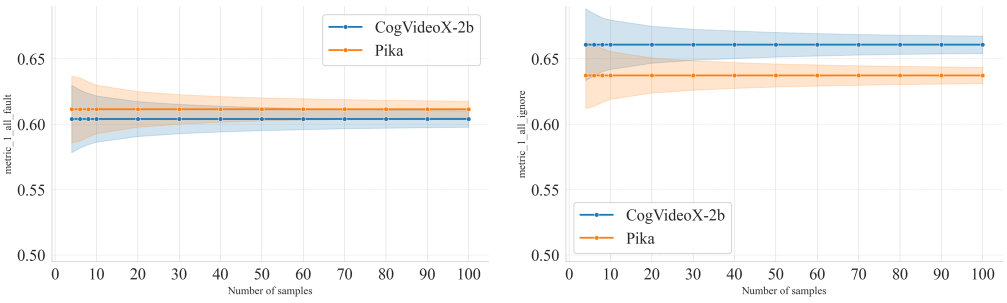
Based on these findings, our benchmark protocol adopts $n_1 = 10$, $n_2 = 5$, and $n_3 = 10$ as optimal parameters balancing statistical power and evaluation efficiency, leading to total 2079 video samples. The sample numbers of these 60 causal systems are shown in Figure 17.

H.5 Threshold-based metrics for rule consistency

For revealing more intuition under the evaluation of rule consistency, we implement the metrics by applying threshold during evaluation. Let t denote the threshold, then metrics for rule consistency are defined as:

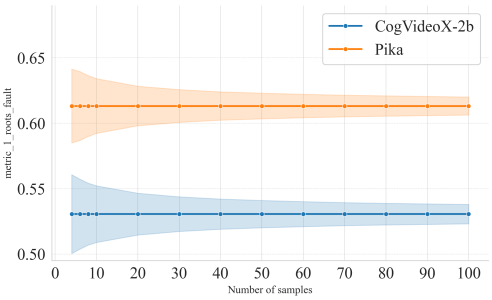
$$s_3^{\text{truth,threshold}} = \frac{1}{m_2} \sum_{Y_j \in \mathbf{Y}} \mathbb{1}\left(s_3^{\text{truth}}(Y_j) \geq t\right),$$

$$s_3^{\text{observe,threshold}} = \frac{1}{m_2} \sum_{Y_j \in \mathbf{Y}} \mathbb{1}\left(s_3^{\text{observe}}(Y_j) \geq t\right).$$

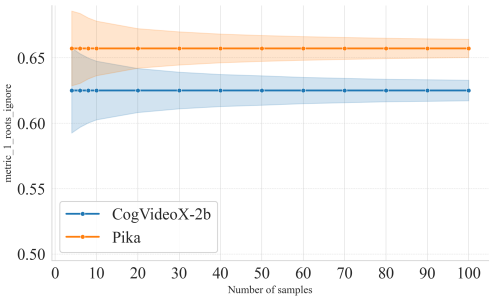


(a) s_1^{all} with N/A as incorrect

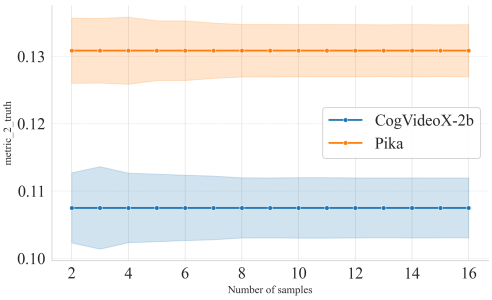
(b) s_1^{all} with N/A ignored



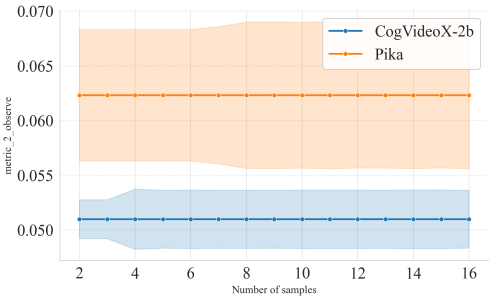
(c) s_1^{roots} with N/A as incorrect



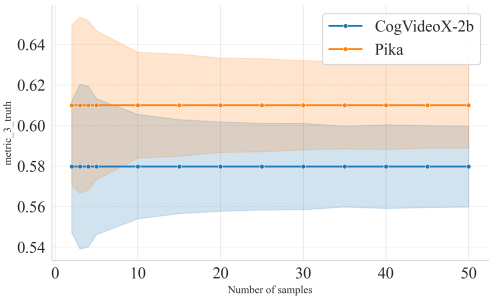
(d) s_1^{roots} with N/A ignored



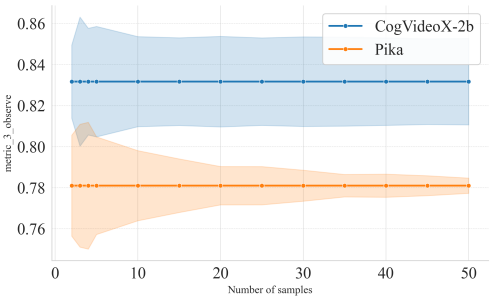
(e) s_2^{truth}



(f) s_2^{observe}



(g) s_3^{truth}



(h) s_3^{observe}

Figure 16: Estimated confidence interval for each metric as the sample size increases.

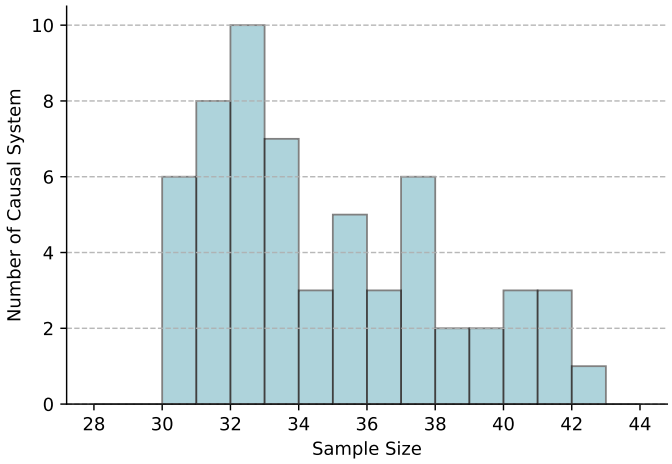


Figure 17: Sample numbers of the 60 causal systems in VACT benchmark.

These metrics measures the probability that for a given causal rule, the model gives a correct value for the outcome variable corresponding to this rule. We calculate these two metrics for threshold in $\{0.65, 0.75, 0.85, 0.95\}$ for each model.

Table 13: Metrics for rule consistency by applying threshold for each rule.

Name	$s_3^{\text{truth,threshold}}$				$s_3^{\text{observe,threshold}}$			
	0.65	0.75	0.85	0.95	0.65	0.75	0.85	0.95
CogVideoX1.5-5B	.19±.04	.08±.03	.02±.01	.00±.00	.61±.05	.48±.05	.30±.05	.15±.04
CogVideoX-5B	.24±.04	.10±.03	.00±.00	.00±.00	.60±.05	.45±.05	.28±.05	.14±.04
CogVideoX-2B	.32±.05	.17±.04	.06±.02	.04±.02	.59±.05	.54±.05	.34±.05	.23±.05
VideoCrafter2	.18±.04	.07±.03	.01±.02	.00±.01	.59±.05	.52±.05	.36±.05	.23±.05
Pyramid Flow	.23±.04	.07±.02	.00±.00	.00±.00	.63±.04	.45±.05	.32±.05	.21±.05
HunyuanVideo	.30±.05	.08±.03	.03±.02	.00±.00	.56±.04	.45±.05	.30±.05	.18±.05
Pika	.27±.05	.06±.02	.00±.00	.00±.00	.66±.05	.57±.05	.35±.05	.26±.05
Hailuo	.28±.05	.15±.04	.05±.02	.00±.00	.66±.05	.53±.05	.35±.06	.17±.05
Gen-3 Alpha	.26±.05	.15±.04	.03±.02	.00±.00	.63±.05	.51±.05	.39±.05	.23±.05
Kling	.23±.04	.11±.03	.05±.02	.01±.01	.60±.04	.45±.05	.29±.06	.20±.04

Results are shown in Table 13. From the table we can see that, for a specific model and threshold, $s_3^{\text{truth,threshold}}$ is much smaller than $s_3^{\text{observe,threshold}}$, showing that compared with incorrect understanding of causal rules, the incorrectness of outcome variable is much more caused by the inconsistency of root variables. For a threshold as high as 0.95, $s_3^{\text{observe,threshold}}$ is also significant for all models, revealing that these models have a correct understanding of causal rules for some causal systems. However, $s_3^{\text{truth,threshold}}$ is insignificant for threshold 0.95, which may because that the models do not understand the correct value of root variables described in the prompt.

H.6 Human-sourced benchmarking

To validate the effectiveness of automatically generated causal systems, we manually annotated an additional 60 causal systems for these 20 scenarios through crowd experiments under identical instructions. For these human-annotated causal systems, we conducted experiments using three video generation models: CogVideoX1.5-5B, Hailuo, and Pika. The metric results are presented in Table 14, with missing value (N/A) cases analogous to Appendix H.3 shown in Table 15, and threshold sensitivity experiments similar to Appendix H.5 summarized in Table 16.

Table 14: VACT benchmark on prevailing VGMs on human-sourced causal systems.

Model Names	N/A ratio	Text Consistency ↑		Generation Consistency ↓		Rule Consistency ↑	
		all	root	truth	observe	truth	observe
CogVideoX1.5-5B	.11	$.58_{\pm.01}$	$.58_{\pm.02}$	$.09_{\pm.01}$	$.08_{\pm.01}$	$.54_{\pm.02}$	$.69_{\pm.02}$
Pika	.18	$.57_{\pm.01}$	$.55_{\pm.02}$	$.07_{\pm.01}$	$.06_{\pm.01}$	$.54_{\pm.02}$	$.67_{\pm.02}$
Hailuo	.14	$.63_{\pm.01}$	$.62_{\pm.02}$	$.07_{\pm.01}$	$.08_{\pm.01}$	$.55_{\pm.01}$	$.70_{\pm.02}$

The results demonstrate that all metric scores derived from human-annotated causal systems closely align with those obtained from automated causal systems. This indicates that the automatically generated causal systems effectively capture scenario-specific features and critical variables while establishing valid rules. Notably, the N/A ratio in observational data increased across all models compared to results from automated causal systems. Concurrently, model performance on rule consistency metrics exhibited degradation. These observations suggest that video generation models face slightly bigger challenges in interpreting human-annotated causal systems, likely due to increased complexity and ambiguity in manually defined causal relationships.

Table 15: The ratio of N/A variables, correct variables and incorrect variables for text consistency on human-sourced causal systems.

Name	**N/A ratio**	**correct ratio**	**incorrect ratio**
CogVideoX1.5-5B	.12	.51	.37
Pika	.17	.48	.35
Hailuo	.13	.54	.33

Table 16: Metrics for rule consistency on human-sourced causal systems by applying threshold for each rule.

Name	$s_3^{\text{truth,threshold}}$				$s_3^{\text{observe,threshold}}$			
	0.65	0.75	0.85	0.95	0.65	0.75	0.85	0.95
CogVideoX1.5-5B	$.23_{\pm.04}$	$.14_{\pm.03}$	$.03_{\pm.02}$	$.03_{\pm.02}$	$.56_{\pm.04}$	$.47_{\pm.05}$	$.29_{\pm.04}$	$.11_{\pm.03}$
Pika	$.19_{\pm.04}$	$.09_{\pm.03}$	$.05_{\pm.02}$	$.05_{\pm.02}$	$.45_{\pm.04}$	$.38_{\pm.05}$	$.30_{\pm.04}$	$.20_{\pm.04}$
Hailuo	$.22_{\pm.04}$	$.12_{\pm.03}$	$.03_{\pm.01}$	$.01_{\pm.01}$	$.56_{\pm.04}$	$.42_{\pm.04}$	$.29_{\pm.05}$	$.16_{\pm.04}$

I CASE STUDY ON BENCHMARK RESULTS

I.1 ABOUT THE "DEGENERATIVE" RULES

Since our metric 2 only focus on the stability but not the correctness, we are worried that the lower (better, stabler) metric 2 combined with the poorer metric 1 and metric 3 (low accuracy) actually implies that the model learns shortcut on common scenario. In many cases, models ignore the changes in $\mathbf{X}$ but directly generate the most common results. We support our concern through some case studies.

In the scenario about "A burning candle is placed with (wind and rain).", a key outcome is whether the candle remains lit or is extinguished by these environmental factors. However, we found that most of the VGMs consistently generate a candle that continues to burn, without accounting for these influences. For Gen-3 Alpha, in three test cases of this scenario, the expected outcome—an extinguished candle—occurred 11, 10, and 10 times, respectively. However, the actual results were only 2, 0, and 3 instances where the candle was extinguished. This makes the "candle extinguished" result appear almost as a constant "False". Similar phenomenon can be found about the outcome "whether the pencil mark has been removed" in the scenario "Rubber eraser rubs off (pencil) marks

on paper.". Similarly, the statement "the water color is uniform" is always false after "Dropping dye into the water" regardless of "whether the water is stirred sufficiently".

I.2 About sample-based score

Here we demonstrate how the sample-based scores provide a more detailed analysis of model behavior by an example. Taking the model CogVideoX1.5-5B and the scenario "*A hand squeezes a sponge.*" as the example, one of the generated causal system is:

"hand squeeze sponge $\wedge$ sponge is wet $\rightarrow$ water is squeezed out".

By checking the scores of the generated videos, we observe that some videos have a metric 3 score (rule consistency) of 1.0 (full score), indicating that these videos comply with all rules. We show these videos are shown in Figure 18, corresponding to some successful generation. As comparison, some of generation have much lower metric 3 score and are shown in Figure 19. Intuitively, we can see the gap in generated causal content between them. In this way, we can select some better samples which could be used to further finetune the model to achieve better causal alignment in this scenario.

(a) No squeeze, not wet, no water squeezed out.

(b) Squeeze, not wet, no water squeezed out.

(c) Squeeze, wet (deeper color in first several frames), water squeezed out.

Figure 18: Good examples with rule consistency score 1.0.

(a) Squeeze, wet, no water squeezed out.

(b) Squeeze, not wet, water squeezed out (water droplets appear in the last two images).

Figure 19: Bad examples with rule consistency score 0.0.

J Discussion about LLM prompt enhancement technique

Sora (OpenAI, 2024b) inherits a technique from Dall-E (Betker et al., 2023) called prompt enhancement, where the model doesn't directly rely on the provided text prompt for generation. Instead, it first uses a pre-trained LLM to expand the prompt, adding missing elements such as environmental details and turning abstract concepts into more intuitive descriptions. Some models have already integrated this functionality into their latest VGM versions.

We indeed observed that this technique slightly improved the model's ability to correctly understand causal rules. However, when scenarios became slightly more complex, either the LLM's expansion did not address the relevant parts, or even if the LLM did provide an expansion, the VGM still failed to generate reasonable results. We believe that, this technique is not the ultimate solution to

creating a world simulator. On one hand, it supplements the VGM's shortcomings by leveraging the LLM's capabilities, but it doesn't address the VGM's core strengths. On the other hand, prompt enhancement cannot capture every detail because vision is much more complicated and informative than text, and once a scenario goes beyond the scope of the prompt, the VGM will struggle to respond appropriately.

To faithfully reflect the performance of the VGMs themselves, we disabled the prompt enhancement option for all closed-source models (where possible). Specifically, for Gen-3 and Hailuo, we turned off this feature. For Kling and Pika, however, we couldn't find any official description on whether this technique was used.