CAUSAL MARKOV SUPERVISION: INJECTING CAUSAL INDUCTIVE BIAS INTO VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models should answer "what-if" questions about interventions on an environment. We introduce CAUSAL MARKOV SUPERVISION (CMS), a post-training procedure that injects a user-specified causal DAG into a video world model to improve intervention simulation. Each training episode is labeled with an event trace of DAG-variable outcomes and their timing, and a pre-trained video world model learns to predict traces along with predicts frames and actions after post-training. The DAG supervises temporal ordering and constrains attention so trace predictions satisfy its Markov property. In a realistic, AAA-quality Unity environment, confounding by player choice causes standard world-model training to fail on the causal effect of a "trapped chest" mechanic on level completion. CMS accurately simulates identifiable intervention outcomes without degrading pretrained visual fidelity. With intervention labels, it also exploits intervention-induced independences to simulate beyond the observational training distribution. Finally, we show that a vision–language model can automatically label training episode with DAG event traces.

1 INTRODUCTION

World modelling aims for agents that "imagine" (simulate) the consequences of actions before acting by learning representations of environment dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2020). Reproducing trajectories like those in logged episodes, however, is insufficient for reasoning about *interventions* that alter environmental mechanisms in ways unseen during training. Generalizing to such changes requires *causal representations* of how the environment responds (Schölkopf et al., 2021; Bareinboim et al., 2022).

One approach is to scale multimodal training, then use causality-sensitive probes to test which mechanisms and interventions the learned representation captures *a posteriori* (Schölkopf et al., 2021; Bau et al., 2017; Geiger et al., 2021). However, in many applications, we know *a priori* what interventions matter to us and what causal structures govern their outcomes. Traditional causal inference begins with a question and a DAG (Pearl, 2009); modern deep generative world models lack a compatible workflow.

CAUSAL MARKOV SUPERVISION (CMS) fills this gap by injecting a user-specified causal DAG into a pretrained video world model. It constrains the learned dynamics to satisfy the causal Markov properties implied by the DAG. Each training episode receives a *DAG trace* recording each DAG variable's realized outcome and timing in that episode. Then, post-training extends a world model to predict DAG traces jointly with the video-frame and action predictions learned during pretraining. CMS's *Markov loss* compares paired predictions of a masked trace outcome or image tokens. The paired queries differ only in history that the DAG deems irrelevant given the evidence. By penalizing disagreement in the pair's predictions, CMS teaches the transformer causal Markovian lessons about what history *not* to attend to when generating rollouts. CMS naturally adapts to intervention data by extending Markov loss to account for the additional independences induced by interventions. CMS can work with pretrained world models, and we show that the DAG-trace labels it requires can be generated automatically from training episodes with vision–language models.

Our contributions are: (1) a DAG-forward workflow for specifying the causal abstractions a world model should reason about; (2) a novel post-training objective for incorporating causal Markov assumptions into representations learned by block-causal transformer-based video world models; (3)

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

an extension that incorporates intervention labels; and (4) an evaluation in an AAA-quality Unity simulator benchmark with realistic video game mechanics built on a ground-truth causal structure. CMS recovers causal structure and predicts intervention outcomes under strong confounding while preserving real-time, high-fidelity video generation.

2 Background and Related Work

World models. Latent world models learn environment dynamics from observations and actions, then predict or simulate future trajectories (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2023; Schrittwieser et al., 2020). Video world models render those trajectories as frames, making the learned dynamics visually inspectable (Alonso et al., 2024; Valevski et al., 2025; Kanervisto et al., 2025). We use a transformer architecture that separates temporal dynamics from frame generation (Kanervisto et al., 2025; Huang, 2025). A temporal backbone updates a compressed state from the history, while a frame head renders each time step (Kanervisto et al., 2025). This separation was initially motivated by speed: the backbone advances in one transformer pass, while the frame head contains the repeated iterations used by popular generative methods (e.g., MaskGIT, diffusion transformers) (Kanervisto et al., 2025; Chang et al., 2022; Alonso et al., 2024). But it also disentangles representation of environmental dynamics for the rendering of high-dimensional visuals – an benefit we leverage with CMS. We use the WHAM subclass of this architecture and treat speed and visual quality as table stakes for improved causal reasoning (Kanervisto et al., 2025).

Causal representations, DAGs, and the Markov property . Causal inference studies the effects of *interventions*: actions that change a data-generating process (Eberhardt & Scheines, 2007). For a world model, this means changing a mechanism underlying state transitions, not merely taking an action within that mechanism. A model trained on logged trajectories can will learn to reproduce the latter but struggle with the former; to "imagine" beyond training, it must also predict environmental changes that break action-outcome patterns seen during training (Hafner et al., 2019; 2020; 2023). Predicting intervention outcomes provably requires causal inductive bias (Pearl, 2009; Shpitser & Pearl, 2008; Schölkopf et al., 2021). Such bias may emerge from multimodal scale (Huang, 2025), self-supervision (Bardes et al., 2024), regularization, or curated data (Locatello et al., 2019). Because these sources are implicit, existing work often probes learned representations for evidence of causal structure (Bau et al., 2017; Schölkopf et al., 2021). We instead make the bias explicit: when the interventions and relevant causal mechanisms are known a priori, we encode them into the world model.

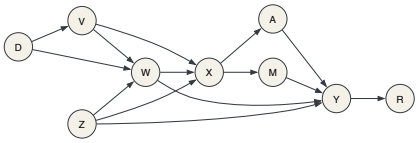
A causal DAG makes this bias explicit (Pearl, 2009; Peters et al., 2017). The modeler starts with intervention questions and defines a DAG over sufficient abstractions for answering those questions (Appendix A). And while DAGs traditionally come from human domain knowledge, language models have emerged as sources of implicit causal knowledge (Kıcıman et al., 2024). We show how CMS can use vision–language models to construct a DAG and its traces from video.

A causal DAG states what causes what, and provides a Markovian abstraction for reasoning about a system, even when the underlying dynamics are non-Markovian (Pearl, 2009; Peters et al., 2017). Each variable is independent of its non-descendants conditional on its parents; d-separation implies further conditional independences (Appendix D) (Pearl, 1988; Geiger et al., 1990). An intervention cuts a target off from its parents by deleting its incoming edges (graph surgery, Appendix A) (Pearl, 2009; Eberhardt & Scheines, 2007). The now-parentless target is unconditionally independent of its non-descendants.

We use these DAG-derived Markovian independences to guide attention-based prediction. When a target is independent of a predictor under given conditions, the model should learn to ignore that predictor. And when an ignored predictor becomes relevant again given new evidence (collider dependence in DAG terms), it should stop ignoring it. To our knowledge, CMS is the first video-world-model training method to use this reasoning.

3 Workflow: Modelling a Dungeon Crawler

We illustrate CMS in a visually rich Unity dungeon crawler with AAA-style graphics and common game mechanics. Basic player policies generate the training episodes; Appendix B.1 details the environment.



(a) The causal DAG.



(b) Glow, poison, and ward runes.



(c) Injured player, green exit.

Figure 1: The chest-room mechanic. (A) DAG over dungeon D, curse Z, rarity V, trap W, glow X, ward M, opening A, injury Y, and exit R. (B) At 18.4 seconds, the player opens ($A = 1$) a glowing chest ($X = 1$) as ward runes fire ($M = 1$) and poison spills out ($W = 1$). (C) At 22.8 seconds, the injured player ($Y = 1$) takes the green, good exit ($R = 1$). Panels B and C are unaltered frames from Unity pretrain clip 005578; Appendix B.1 details rendering.

Motivating Example: Chest Glow Mechanics. Consider a game-studio analyst deciding how to modify their game's loot-chest mechanic. The chests glow to signal rare loot, and there are signs that this may improve level outcomes and engagement. This motivates the question, can manipulating the glow mechanic improve player engagement?

Complicating this question, chests may release poison traps, and players may be cursed because of past choices. Not only do cursed players encounter more traps, but the curse causes chests to glow even without rare loot, in order to lure them into traps. Cursed players also suffer more injuries. So the analysit must be cautious; Weakening the curse–glow link could dilute the stakes of the curse mechanic, while an overly strong trap–glow link could cause cursed players to simply avoid glowing chests.

The studio won't allow the analyst to run an A/B test, the gold standard of causal inference, for fear it could harm the player experience. However, the analyst has access to a cutting-edge world action model, that not only simulates gameplay visuals, but also how the studio's players tend to play the game. The analyst seeks to use the world model to simulate how that gameplay would unfold under different interventions on the glow mechanic. Given they know the intervention distribution of interest, they apply CMS post-training so the model can better simulate from that target distribution.

Step 1: Define the intervention query. The analyst defines binary abstractions: A for opening the chest, X for its glow, and R for reaching the good exit. For $x \in \{0, 1\}$, the intervention target is $\gamma_x = \mathbb{P}(R = 1 \mid \text{do}(X = x), A = 1)$. Conditioning on $A = 1$ restricts the query to opened chests; $\text{do}(X = x)$ randomizes glow as an A/B test would. The effect of forcing glow rather than no glow is $\gamma_1 - \gamma_0$.

The observational analogue replaces intervention with conditioning: $\gamma_x^{\text{obs}} = \mathbb{P}(R = 1 \mid X = x, A = 1)$. A standard world model may conflate $P(R \mid \text{do}(X), A)$ with $P(R \mid X, A)$. Here, the observational and interventional contrasts have opposite signs (Table 2), so that conflation reverses the conclusion.

Step 2: Define the DAG. Trap status W and curse status Z confound the glow–outcome relationship. The DAG must include all common-cause confounders (satisfying *causal sufficiency*). The analyst also includes variables that complete the causal narrative between glow X and outcome R; for example, whether in the dungeon (D) in included for narrative completeness. Table 1 lists the variables, and Figure 1(A) shows the resulting DAG.

Step 3: Create the trace labels. Each episode needs a DAG trace recording every variable's outcome and realization time. The analyst can derive traces from micro-level event logs or use a vision–language model (VLM) to detect outcomes and timing from frames. Sections 5 and 6 and Appendix E compare frontier VLMs on this task.

4 Causal Markov Supervision

Preliminaries. Let G be the DAG over the environment and action variables $\mathcal{X}_G$, and let $\tau(E)$ be the time at which $E \in \mathcal{X}_G$ is realized. In intervention setting ξ, let G^ξ be the DAG after deleting edges into the intervention target, with $G^\varnothing = G$; $\text{Pa}_{G^\xi}(E)$ denotes the parents of E in that graph. Let $\mathcal{S}$ denote latent environment variables omitted from G. Before $\tau(E)$, the trace entry for E is ⑦;

Variable	Indicator	Variable	Indicator
D	Dungeon (elsewhere).	M	Protective runes fire (ward does not fire).
Z	Player is cursed (not cursed).	A	Player opens the chest (bypasses it).
V	Chest contains a rare item (ordinary item).	Y	Player is injured (not injured).
W	Chest is trapped (not trapped).	R	Player reaches the good exit (bad exit).
X	Chest glows gold (does not glow).		

Table 1: Binary variables in Figure 1(A): each indicator describes value 1, with value 0 in parentheses.

Route	No glow	Glow	Glow − no glow
Observational	$\gamma_0^{\text{obs}} = .444$	$\gamma_1^{\text{obs}} = .234$	$-.210$
Interventional	$\gamma_0 = .391$	$\gamma_1 = .554$	$+.162$

Table 2: Ground-truth observational analogues and interventional targets at $A = 1$. The opposite contrast signs show that a model unable to distinguish observation from intervention reaches the wrong conclusion.

afterward it remains at its realized value. Let $B_{s\ldots t}$ denote the sequence of frames and actions from time s to t, and let $\zeta_{s\ldots t}$ denote the trace-variable realizations over the same interval. ζ_t^E denotes the trace entry for E at time t. Let $\mathbf{o}_t = (\zeta_t^E)_{E\in\mathcal{X}_G}$ be the snapshot of trace outcomes at time t, with ⍰ for pending variables. We write $H_{s\ldots t} := (B_{s\ldots t}, \mathbf{o}_{s\ldots t})$ for the corresponding episode history. We use subscripts such as $< t$, $\leq t$, $> t$, and $\geq t$ for histories before and after time t. Because outcome snapshots are compact, the model retains $\mathbf{o}_{0\ldots t}$ while using a window of at most k frame/action blocks. With $m_t = \max(0, t - k + 1)$, its input history is $H_t^{(k)} = (B_{m_t\ldots t}, \mathbf{o}_{0\ldots t})$.

Let $\Psi_\theta[I_s, Q, H]$ denote the mapping entailed by the shared block-causal transformer backbone with learned weights θ. I_s encodes the intervention on a transition to $s + 1$ (Section 4.4). Q denotes the query prefix: $Q_\varnothing$ is the learned no-query prefix, whereas $Q_\chi(\eta)$ gives the model selected trace values as evidence for a conditional-independence statement χ. The argument η includes the evidence and uses MASK in place of any value the model must predict; variables omitted from η are also masked. We use $Q(\eta)$ as a shorthand when we don't need to refer to an explicit conditional-independence statement.

4.1 CHRONOLOGICAL PREDICTION

CMS extends the world model's step-by-step prediction of frames and actions to also include the trace. It must predict *when* variables in the trace are realized as well as their realized outcomes. We model *when* with a hazard function that provides the probability of each event occurring at each time step conditional on the sliding history window.

$$\begin{aligned}
&(1)\quad r_{t-1}^{(k)} = \Psi_\theta[I_{t-1}, Q_\varnothing, H_{t-1}^{(k)}] && (4)\quad \kappa_{E,t} = \textstyle\prod_{P\in \mathrm{Pa}_{G^\xi}(E)} \mathbf{1}[\zeta_{t-1}^P \neq ⍰] \\
&(2)\quad \pi_{E,t}^{\text{hazard}} = g_E^{\text{hazard}}(r_{t-1}^{(k)}) && (5)\quad \widetilde{\lambda}_{E,t} = \kappa_{E,t}\lambda_{E,t} \\
&(3)\quad \lambda_{E,t} = \text{sigmoid}(\pi_{E,t}^{\text{hazard}})
\end{aligned} \tag{1}$$

The backbone maps $H_{t-1}^{(k)}$ and the prefix tokens to a trace-position hidden state $r_{t-1}^{(k)}$. The shared HazardHead g^{hazard} produces a logit and raw hazard $\lambda_{E,t}$ for each E, predicting realization at time t before the outcome is observed. The gate in Equation (1) allows a child to resolve only after all its parents have realized, yielding the rollout hazard $\widetilde{\lambda}_{E,t}$. Figure 2 shows how the HazardHead operates in conjunction with the other heads.

For each $E \in \mathcal{X}_G$, let $\rho_{E,t}$ indicate that E is unrealized at $t - 1$ and $d_{E,t}$ that it realizes on the transition to t ($d_{E,t} = \mathbf{1}[\zeta_{t-1}^E = ⍰, \zeta_t^E \neq ⍰]$). The hazard loss uses the raw $\lambda_{E,t}$ from Equation (1), not the gated rollout hazard. It scores both E realizing and E remaining unrealized when E was unrealized at $t - 1$:

$$\mathcal{L}_{\text{hazard}} = -\frac{\sum_{t,E} \rho_{E,t} \left[d_{E,t} \log \lambda_{E,t} + (1 - d_{E,t}) \log(1 - \lambda_{E,t})\right]}{\sum_{t,E} \rho_{E,t}}. \tag{2}$$

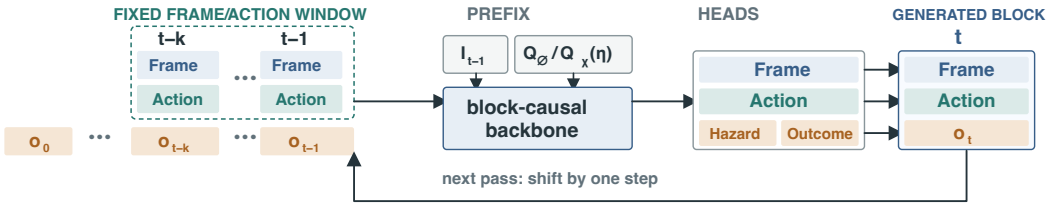


Figure 2: A bounded window of frames/actions plus a full history of trace outcomes $\mathbf{o}$ feed the backbone. Generated outputs are appended for the next pass.

Even when the gate blocks realization, the raw hazard receives a gradient toward zero for premature predictions.

The OutcomeHead produces outcome logits $\pi^{\text{outcome}}_{E,t} = g^{\text{outcome}}_{E}(r^{(k)}_{t-1})$. Only transitions on which E realizes contribute to its loss term. If no variable realizes in a batch, this outcome term is zero; the hazard term is likewise zero if no variable is unrealized at the start of a transition.

$$\mathcal{L}^{\text{chron}}_{\text{outcome}} = -\frac{\sum_{t,E} d_{E,t} \log \text{softmax}(\pi^{\text{outcome}}_{E,t})_{\zeta^E_t}}{\sum_{t,E} d_{E,t}}. \tag{3}$$

Separate heads predict the next frame and action. The chronological objective is

$$\begin{aligned}\mathcal{L}_{\text{chron}} = \alpha_{\text{frame-pred}}\mathcal{L}^{\text{chron}}_{\text{frame}-\text{pred}} + \alpha_{\text{frame-ref}}\mathcal{L}^{\text{chron}}_{\text{frame}-\text{ref}} \\ + \alpha_{\text{action}}\mathcal{L}_{\text{action}} + \alpha_{\text{hazard}}\mathcal{L}_{\text{hazard}} + \alpha_{\text{outcome}}\mathcal{L}^{\text{chron}}_{\text{outcome}}.\end{aligned} \tag{4}$$

$\mathcal{L}_{\text{frame}-\text{pred}}$ scores predicted frame content (discrete tokens in our implementation); $\mathcal{L}_{\text{frame}-\text{ref}}$ refines those predictions (bridge-flow squared error in our implementation). Only the frame-prediction distribution, not refinement, enters the Markov loss. Appendix F gives the frame-model details and Appendix G gives the weights.

4.2 Markov Supervision

Causal Markov Supervision exploits conditional independences implied by the DAG's Markov property.

Baseline query loss. We anchor Causal Markov Supervision by training the model to predict masked prediction queries. With query $Q(\eta)$, the model receives a random subset of an episode's final outcomes and marks the rest MASK in the query; their input-trace entries are set to ⍰. Given the episode history, intervention setting, and revealed values, the query objective trains predictions for the hidden outcomes and next-frame tokens:

$$\mathcal{L}_{\text{query}} = \alpha_{\text{query-outcome}}\mathcal{L}^{\text{query}}_{\text{outcome}} + \alpha_{\text{query-frame}}\mathcal{L}^{\text{query}}_{\text{frame}-\text{pred}}. \tag{5}$$

Markov supervision seeks to force the model to ignore irrelevant input based on conditional independence. Baseline query loss prevents the model from satisfying that goal simply by generating bogus predictions.

4.3 Extending Markovian assumptions to histories.

The conditional independences and equalities below are evaluated within a fixed intervention setting ξ; we suppress ξ in the probability notation. To extend these independences to episode histories, we make the following assumptions.

- $\mathcal{X}_G$ is causally sufficient with respect to the query variables of interest.
- $\mathcal{X}_G$ includes all query-relevant actions. Other actions may be omitted from G and captured in a separate action channel.
- Latent environment state $\mathcal{S}_{\leq t}$ and past observations $B_{\leq t}$ may affect future state and observations, but do not causally influence the outcomes of $\mathcal{X}_G$.

- Once a variable $E \in \mathcal{X}_G$ is realized, it may causally influence subsequent observations and environment state.

Informally, G must represent the state variables and actions that can causally affect the trace variables in the intervention queries. An earlier history block can therefore inform E only through previously realized variables in $\mathcal{X}_G$ that causally influence that block. For $t < \tau(E)$, let $\mathcal{X}(H_{s\ldots t})$ be those variables realized in the history block $H_{s\ldots t}$. For an admissible block, conditioning on $C \subseteq \mathcal{X}_G$ gives

$$E \perp\!\!\!\perp \mathcal{X}(H_{s\ldots t}) \setminus C \mid C \quad \Longrightarrow \quad E \perp\!\!\!\perp H_{s\ldots t} \mid C. \tag{6}$$

Only histories that preserve the conditional independence are admissible; Appendix D.3 describes how to select them.

The Markov swap. Let $\tilde{H}_{E,C}$ be an admissible block of history before E realizes, with boundaries determined by the realization times of C and E. If $E \perp\!\!\!\perp \tilde{H}_{E,C} \mid C$, then

$$P(E \mid C = c, \tilde{H}_{E,C} = h) = P(E \mid C = c) = P(E \mid C = c, \tilde{H}_{E,C} = h'). \tag{7}$$

We call the comparison of h and h' at fixed $C = c$ a *Markov swap*. These conditional probabilities require support for the supplied values in the same intervention setting; unsupported swaps are excluded.

Reversing directions. Conditional independence is symmetric: $E \perp\!\!\!\perp \tilde{H}_{E,C} \mid C \iff \tilde{H}_{E,C} \perp\!\!\!\perp E \mid C$. Yet matching the model's predictions for E under h and h' does not automatically match its predictions for elements of $H_{E,C}$ under $E = e$ and $E = e'$. CMS trains both directions: one compares predictions of E across histories, and the other compares frame-token predictions across supplied values of E.

For a statement χ, each pass supplies $Q_\chi(\eta)$ with the values held fixed and MASK in place of the value to predict. The original episode remains unchanged; the model instead receives a copy of its window in which withheld trace entries are marked ⍰. To compare the first direction, we change the history and predict E; to compare the reverse, we change the supplied value of E and predict the same masked frame tokens. Both passes use the same intervention setting.

For a target $T \in \mathcal{X}_G$, let h be an admissible model-input window with withheld trace values marked ⍰; define its query-pass outcome distribution by

$$q_{\theta,T}(\cdot \mid \eta, h, I_{t-1}) := \mathrm{softmax}\big(g_T^{\mathrm{outcome}}\big([\Psi_\theta[I_{t-1}, Q_\chi(\eta), h]]_{\mathrm{trace}}\big)\big)\,. \tag{8}$$

Here the subscript trace selects the target's trace-position state; $q_{\theta,T}$ is a distribution, not a vector of logits. The statement χ selects the comparisons; its assignment η supplies the evidence for each backbone pass.

Case: Prediction between independent DAG variables. Let χ be the statement $E \perp\!\!\!\perp F \mid C$ for distinct DAG variables E and F. The backbone also receives the observed value h of a history block $\tilde{H}$. Choose $\tilde{H}$ before either outcome is visible so that $E \perp\!\!\!\perp F \mid C, \tilde{H}$ still holds (Appendix D.3). For two such values e, e' of E and f, f' of F, form four assignments:

$$\begin{aligned} \eta_{E=e} &= \{C = c, E = e, F = \textsc{mask}\}, & \eta_{E=e'} &= \{C = c, E = e', F = \textsc{mask}\}, \\ \eta_{F=f} &= \{C = c, F = f, E = \textsc{mask}\}, & \eta_{F=f'} &= \{C = c, F = f', E = \textsc{mask}\}. \end{aligned} \tag{9}$$

Each minibatch randomly chooses one direction δ (checking that the chosen direction has support in the data): predict F as E changes, or predict E as F changes.

For binary variables the swapped values are 0 and 1.

$$\mathcal{L}^{\mathrm{out}}_{\chi,\delta} = \begin{cases} \mathrm{JSD}(q_{\theta,F}(\cdot \mid \eta_{E=e}, h, I_{t-1}), q_{\theta,F}(\cdot \mid \eta_{E=e'}, h, I_{t-1}))\,, & \delta = 0, \\ \mathrm{JSD}(q_{\theta,E}(\cdot \mid \eta_{F=f}, h, I_{t-1}), q_{\theta,E}(\cdot \mid \eta_{F=f'}, h, I_{t-1}))\,, & \delta = 1. \end{cases} \tag{10}$$

Both directions use the same observed frames and actions; the model-input trace hides E and F, and only the query assignment changes between paired passes.

Case: Predicting a DAG variable from independent episode history. Take two intact episodes i, j from the same intervention setting, with $C = c$ in both. Let h_i, h_j be realizations of an admissible block $\tilde{H}_{E,C}$ before E realizes. For $\chi = (E \perp\!\!\!\perp \tilde{H}_{E,C} \mid C)$, Equation (7) requires the predicted distribution for E to agree under h_i and h_j. The local Markov property is a special case: for $C = \mathrm{Pa}_{G^\xi}(E)$ and an admissible full pre-realization window,

$$E \perp\!\!\!\perp H^{(k)}_{\tau(E)-1} \mid \mathrm{Pa}_{G^\xi}(E). \tag{11}$$

With $\eta_C = \{C = c, E = \textsc{mask}\}$, compare the predictions under both histories with the same intervention prefix:

$$\mathcal{L}^{\mathrm{hist}}_{E|C} = \mathrm{JSD}(q_{\theta,E}(\cdot \mid \eta_C, h_i, I_{t-1}), q_{\theta,E}(\cdot \mid \eta_C, h_j, I_{t-1}))\,. \tag{12}$$

The episodes need not agree on the realized value of E: outcome cross-entropy learns that distribution, while the Markov loss penalizes dependence on the screened history.

Case: Predicting a frame from an independent DAG variable. Let $u < \tau(E)$ be a time in an admissible history block $\tilde{H}_{E,C}$, and let $h = H^{(k)}_{u-1}$ be its preceding context. For the VQ token $V_{u,p}$ at frame position p in frame u, the reverse Markov equality is

$$P(V_{u,p} \mid C = c, E = e, h, a_u) = P(V_{u,p} \mid C = c, E = e', h, a_u), \tag{13}$$

provided the independence still holds with this context and the supplied action a_u (Appendix D.3). For $\chi = (E \perp\!\!\!\perp \tilde{H}_{E,C} \mid C)$, set $\eta_{E=e} = \{C = c, E = e\}$ and $\eta_{E=e'} = \{C = c, E = e'\}$ in one episode, with the same observed frames and actions in both passes. The frame-prediction head returns a distribution for each masked VQ token p from the frame-position state and supplied action a_u:

$$q^{\mathrm{frame}}_{\theta,u,p}(\cdot \mid \eta, h, a_u, I_{u-1}) := \mathrm{softmax}\big(g^{\mathrm{frame}}_p([\Psi_\theta[I_{u-1}, Q_\chi(\eta), h]]_{\mathrm{frame}}; a_u)\big)\,. \tag{14}$$

With the same target tokens, mask $\mathcal{M}_u$, and actions in both passes, let $q^{E=e}_{u,p} = q^{\mathrm{frame}}_{\theta,u,p}(\cdot \mid \eta_{E=e}, h, a_u, I_{u-1})$ and define $q^{E=e'}_{u,p}$ analogously. The reverse-frame loss averages the per-position divergences:

$$\mathcal{L}^{\mathrm{frame}}_{E,u|C} = \frac{1}{|\mathcal{M}_u|} \sum_{p \in \mathcal{M}_u} \mathrm{JSD}\Big(q^{E=e}_{u,p}, q^{E=e'}_{u,p}\Big)\,. \tag{15}$$

We use NetworkX to enumerate the graph's conditional independences; Appendix D describes which histories may be included. Each minibatch samples one local statement χ_{local} and one global statement χ_{global}. For a variable-pair statement, $\mathcal{L}_\chi$ is Equation (10) in one randomly chosen direction; for a history statement it is the sum of Equations (12) and (15). The resulting Markov loss is

$$\mathcal{L}_{\mathrm{Markov}} = \alpha_{\mathrm{local}}\mathcal{L}_{\chi_{\mathrm{local}}} + \alpha_{\mathrm{global}}\mathcal{L}_{\chi_{\mathrm{global}}}. \tag{16}$$

4.4 Modeling interventions

CMS can also use episodes with interventions. The prefix I_{t-1} in each backbone pass describes the intervention on the transition to t. For an intervention $\xi = \mathrm{do}(E = e)$ applied when E realizes, I_{t-1} encodes no intervention for $t < \tau(E)$ and encodes $\mathrm{do}(E = e)$ for $t \geq \tau(E)$. The intervention replaces E's causal mechanism: deleting incoming edges to E gives G^ξ as defined above. This graph determines the parent gate in Equation (1) and which local and global comparisons are available, including those induced by the intervention.

4.5 Training

Each minibatch runs a chronological pass and an ordinary query pass, which trains predictions for withheld outcomes and masked frame tokens. Markov-supervised arms also run the paired query passes described above. The full objective is

$$\mathcal{L}_{\mathrm{post}} = \mathcal{L}_{\mathrm{chron}} + \mathcal{L}_{\mathrm{query}} + \mathbf{1}[\text{Markov arm}]\mathcal{L}_{\mathrm{Markov}}. \tag{17}$$

Appendix G gives the full training algorithm, loss term weights, data allocation, and optimization details.

5 EXPERIMENTS

Environment. We use 28-second, third-person Unity episodes at 10 Hz and 144×256 resolution. Appendix B describes the game mechanics and the simulator's underlying ground-truth SCM.

Using VLMs to label traces. To test the use of VLMs to label traces and DAGs from video, we simulated 200 IID observational chest-room episodes. Each VLM receives 47 indexed frames sampled 0.6 seconds apart from each 28-second episode. Each model receives a text prompt with the definitions of the nine event variables. They are not provided with the DAG. The VLMs are instructed to create the trace based on detecting when the variables are realized and their outcomes. They are instructed to distinguish between having a value of 0 and being unrealized during the episode. We evaluate the VLMs on their ability to label traces accurately and reconstruct the underlying ground-truth DAG. For the second task, we compare the reconstructed DAG against the ground-truth DAG using both structural intervention distance (SID; lower is better) and F1 for detection of ground-truth d-separations.

Ablations. All arms start from the same 171M-parameter world-action model and tokenizer, pretrained on 16,000 episodes (Appendix F). The post-training pool has 1,400 observational episodes and, for each of the nine DAG variables, 700 seeds rendered under both $\mathrm{do}(E{=}0)$ and $\mathrm{do}(E{=}1)$; all $\mathrm{do}(X)$ episodes are held out, leaving 12,600. Four arms add the ingredients of CMS in turn (Appendix G): *Trace* learns to predict the trace, with query training; *Trace+Markov* adds Markov supervision; *Trace+Intervention* adds the non-X intervention episodes; and CMS adds both. Arms without interventions train on 12,600 observational episodes instead, so every arm sees the same number of episodes and updates. The untouched pretrained model is a video-only baseline. The evaluation set has 4,000 held-out seeds, 1,000 in each stratum of chest trap W and player curse Z, each rendered observationally and under $\mathrm{do}(X{=}1)$. Models continue each episode from the frames before the glow X is realized.

Evaluation target. We evaluate the analyst's query on these held-out prefixes: the joint outcome distribution $\gamma_{1,\text{joint}} = P(M, Y, R \mid \mathrm{do}(X{=}1), A{=}1)$ and its good-exit marginal $\gamma_1 = P(R{=}1 \mid \mathrm{do}(X{=}1), A{=}1) = .554$. The estimates $\hat{\gamma}_{1,\text{joint}}$ and $\hat{\gamma}_1$ are the outcome frequencies in each arm's generated traces (VLM labels for the pretrained model, which has no trace), reweighted to the simulator's (W, Z) distribution. Arms trained with interventions simulate $\mathrm{do}(X{=}1)$ directly. Given the DAG, both targets are identified from observational data by backdoor adjustment for W and Z (Appendix H.2), so the other arms generate observational continuations and reach the targets by importance resampling with backdoor-adjustment weights (Appendix H.3). We report five metrics. (1) $\mathrm{JSD}_{\hat{\gamma}_{1,\text{joint}}} = \mathrm{JSD}(\hat{\gamma}_{1,\text{joint}} \,\|\, \gamma_{1,\text{joint}})$ and (2) $\Delta_{\hat{\gamma}_1} = |\hat{\gamma}_1 - \gamma_1|$ measure the error of the simulated intervention. (3) Trace–video agreement $\hat{\rho}$ checks that the generated trace describes the generated video: GPT-5.6 Terra, the most accurate labeler in our VLM study, reads M, Y and R from the frames around each event's generated time without seeing the trace, and $\hat{\rho}$ is the fraction of its labels that match the trace. (4) SID and d-separation F1 compare a DAG learned from each arm's generated observational traces with the true DAG. (5) PSNR and SSIM compare generated frames with simulator video when the recorded actions are replayed. Appendix H describes further metrics.

6 RESULTS

VLM trace labeling. GPT-5.6 Terra returned complete traces for all 200 episodes and labeled 99.3% of event values correctly, the highest of the five VLMs (Table 3); its statistical DAG has SID 15 and d-separation F1 .530. Timing is harder than values: 30.8% of Terra's correct absence-based negatives were declared before absence was conclusive. A semantic refinement call lowers Terra's SID to 12 but its F1 to .416 (Appendix E).

Ablation. Table 4 reports the primary metrics (see Appendix I for additional detailed results). The two arms that simulate the held-out $\mathrm{do}(X{=}1)$ directly have the smallest errors in intervention prediction, $\Delta_{\hat{\gamma}_1} = .003$ (Trace+Intervention) and .063 (CMS), against .125 and .196 when the Trace and Trace+Markov traces are adjusted for W and Z; the intervals overlap. Notably, traces and videos agree on 35–61% of outcome values ($\hat{\rho}_3$), against 99.7% when the same labeler reads simulator video.

VLM	Complete	Value accuracy	SID↓	F1↑
GPT-5.6 Terra	200/200	99.3%	15	.530
Gemma-4-31B-IT	200/200	97.2%	27	.291
Grok-4.3	200/200	97.6%	36	.338
Grok-4-20	200/200	96.7%	36	.290
Qwen3-VL-4B	147/200	86.6%	45	.317

Table 3: VLM trace labeling on 200 observational episodes. SID and d-separation F1 compare each VLM's statistical DAG with the simulator DAG; Qwen's metrics use its 147 complete traces.

Arm	Step	$\Delta_{\hat{\gamma}_1}\downarrow$	$\mathrm{JSD}_{\hat{\gamma}_{1,\mathrm{joint}}}\downarrow$	$\hat{\rho}_3\uparrow$	SID↓	F1↑	PSNR↑	SSIM↑
Pretrained	–	–	—	–	—	–	24.43	.692
Trace	500	.125 [.012, .254]	.103	.354	19	.346	24.20	.688
Trace+Markov	500	.196 [.067, .307]†	.157	.424	18	.578	24.00	.675
Trace+Intervention	500	.003 [.002, .109]	.211	.613	28	.403	23.85	.656
CMS	100	.063 [.004, .141]	.237	.538	44	.320	23.89	.658

Table 4: Primary metrics. Step is the post-training checkpoint. Brackets are 95% stratified bootstrap intervals for $\Delta_{\hat{\gamma}_1}$ when every (W, Z) cell has at least 20 completed qualifying rollouts, else parametric (†; Appendix H.13).

Most disagreements concern the exit R which is last event in the episode and often gets cut off. Also, CMS videos sometimes show an exit of illegible colour. With recorded actions replayed, post-trained arms reach 23.8–24.2 dB PSNR, against 24.4 dB for the pretrained model.

7 Limitations

DAG-based methods are prone to model misspecification, and latent variables (not observable from video) can introduce additional challenges, though standard causal inference techniques can help mitigate some of these issues. More subtlety, a variable maybe be observable but the time of its initial realization maybe be latent. Additionally, realization timing and outcome may be correlated. We suggest an approach to modeling dependence between hazards and outcome in Appendix G.8, but more investigation is needed.

8 Conclusion

World model "imagination" is fundamentally an causal question. Causal Markov Supervision (CMS) introduces a way to "inject" a causal DAG into a cutting-edge video world model by training it to learn and reproduce the DAG's Markov property. CMS is supervised by *DAG traces* labels, and we show that frontier VLMs can create traces with accuracy. We demonstrate a post-training procedure for training a cutting-edge video world model to answer intervention questions correctly withou sacrificing visual fidelity, assuming the questions are identified given the DAG and data. In future work, we intend to extend this approach to multiworld counterfactual reasoning.

AI use statement

Prior to manuscript writing, we used LLM chat as a sounding board for refining ideas and to help with internal communication of progress, such as PowerPoint presentations. We also used AI-supported coding tools to build the Unity chest room environment. In particular, after creating the ground-truth causal model, we relied on AI-supported tools to form the connective tissue between that model and Unity. We used agents for task management, including coordinating the scheduling and use of limited compute resources. Code for the block-transformer architectures was sourced from internal repositories with published results that predate this work. During manuscript writing, we used LLM-based tools to help make high-level decisions about document structure. We also relied on LLM tools for editing grammar and syntax, debugging LaTeX, and clarifying mathematical notation.

The primary author was responsible for the prose. The authors reviewed the AI-assisted work and take responsibility for the final content of the paper.

ETHICS STATEMENT

This work uses a scripted game environment and synthetic images, actions, and event traces. It involves no human participants, personal data, or decisions about people, and therefore did not require human-subjects review. The vision–language models evaluated in the annotation study receive only rendered synthetic frames and task instructions. The main risk is scientific overgeneralization: a method validated with a known causal graph in one synthetic room should not be treated as establishing causal validity in an open-world or safety-critical setting. We mitigate this risk by stating the DAG, support, trace, and intervention assumptions explicitly; separating exact simulator targets from learned-model estimates; reporting non-estimable cases and support diagnostics; and limiting our claims to the evaluated environment. We intend to release the synthetic-data manifests, model and evaluation artifacts, and analysis code needed to audit the reported results, subject to the licenses and access conditions of the underlying software and model services.

REPRODUCIBILITY STATEMENT

The paper and supplement specify the supplied graph and verified Markov inventories, the chronological and query objectives, the ablation data allocation, the held-out intervention, the estimands, and the uncertainty procedure. Appendices D, G, H, and K provide the proof, training, evaluation, and artifact details. The accompanying artifacts pin data and checkpoint revisions, random seeds, cohort and continuation identifiers, per-pass exposure ledgers, cached rollout rows, and scripts for regenerating the reported tables and figures.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems*, 2024.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl's hierarchy and the foundations of causal inference. In Hector Geffner, Rina Dechter, and Joseph Y. Halpern (eds.), *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 507–556. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450395861. doi: 10.1145/3501714.3501737.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6541–6549, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Bau_Network_Dissection_Quantifying_CVPR_2017_paper.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.02997.

Dan Geiger, Thomas Verma, and Judea Pearl. Identifying independence in Bayesian networks. *Networks*, 20(5):507–534, 1990.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Xun Huang. Towards video world models. *Online essay*, 2025. URL https://www.xunhuang.me/blogs/world_model.html.

Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcárcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, Abdelhak Lemkhenter, Chentian Jiang, Gavin Costello, Gunshi Gupta, Marko Tot, Shu Ishida, Tarun Gupta, Udit Arora, Ryen W. White, Sam Devlin, Cecily Morrison, and Katja Hofmann. World and human action models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025. doi: 10.1038/s41586-025-08600-3.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=mqoxLkX210.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020.

Ilya Shpitser and Judea Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, 2025.