TECHNICAL REPORT
R-155
May, 2026

Counterfactual Debugging the World Model Transfer Gap

Mingxuan Li[1*] **Kai-Zhan Lee**[1*] **Michael Dennis**[2] **Elias Bareinboim**[1]
[1] Columbia University, [2] Google DeepMind
[1]{ml,eb,kl}@cs.columbia.edu, [2]dennismi@google.com

Abstract

Policies achieving strong performance in simulators or learned world models can fail when deployed into the real environment. While there must be environmental differences that account for these failures, not all differences are equally responsible. A benign error of irrelevant visual details may co-exist with a failure to predict a single critical transition causing an inevitable catastrophic failure. In this paper, we introduce *counterfactual debugging* as an approach for *world model transfer gap attribution* to identify the time steps whose transition or reward errors are *causally responsible* for a performance degradation exhibited in a real trajectory, rather than only visually or statistically different. We develop a scalable algorithm for computing these attributions that exploits the sparsity of causal errors by recursively divide-and-conquer, significantly reducing computational costs and achieving an exponential speedup. The resulting ranked attribution report can better explain the performance gap in the real trajectory. Experiments across several distinct environments with a variety of injected world-model failures (observation corruption, reward misprediction, and physics violations) demonstrate that counterfactual debugging correctly identifies the errors that are responsible for performance gap, providing theoretically grounded, actionable insights for model improvement.

1 Introduction

A world model can be thought of as a simulator of the domain in which the agent aims to operate. Learning such a predictive model of the environment's transition dynamics and reward function, can unlock the ability to simulate experience for planning or policy optimization [39, 40]. In practice, recent world models have achieved remarkable success in complex domains, from mastering Atari games with discrete latent representations [18, 35] to planning with learned dynamics in continuous control [16, 22], to generating fully imagined worlds[7]. Yet, a persistent challenge remains: when the simulator or learned world model is wrong, policies trained on it can fail catastrophically once deployed in the real environment, despite success in simulation [22].

The difficulty is not merely that world models make errors, every learned model does, but that only some of these errors matter. A world model may hallucinate visual details that are irrelevant to the task, or subtly mispredict a single transition that cascades into a qualitatively different outcome. Such a single misprediction could lead to autonomous driving car crash or robots hurting its operator but it may not be distinguishable visually nor from the average prediction accuracy. Fig. 1 makes this concrete: a few occluded frames in a simple Pong game rollout suffice to knock the agent's return from 21 down to 15, even though visually most of the two trajectories look nearly identical. A raw visual comparison flags every perturbed frame equally, but only a small subset of those frames is the actual cause of the transfer gap in the real world; pinpointing that causal subset, rather than the full set of visual or statistical anomalies, is the diagnostic we should aim for.

*Equal contribution.

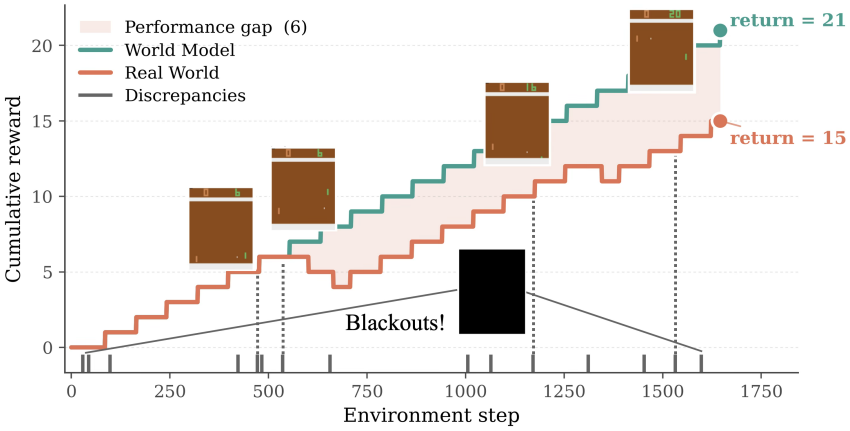


Figure 1: **Which transition discrepancy should be blamed for the performance degradation?** A PPO policy on Atari Pong running under a world model (green) scores full mark, 21, while the same policy running against a turbulent "real world", where random blackouts occlude the screen (red) only scores 15. Crucially, most of the occluded frames have little bearing on the final outcome, only a small, hard-to-identify subset is *causally responsible* for the collapse. **In this example, somewhat surprisingly, only two of the blackout frames are causally responsible for the performance degradation that happens more than 50 steps later**. See more details in Sec. 4.1.

Existing tools fall short of finding such causal subsets: aggregated prediction-error metrics [18, 35] and model-uncertainty heuristics [22] flag many or all bad transitions but cannot single out which ones drive the transfer gap, while saliency based methods [13] and policy explanation over state features methods [5, 26] surface correlations in visual features or actions rather than causal attributions on the world model's dynamics. The most closely related solution is the line of feature attribution methods (e.g., SHAP[25] and LIME[33]). However, they condition on static inputs and cannot model the sequential, causal structure of a multi-step trajectory, where an error at step t propagates forward through subsequent transitions. Therefore, none of these answers the targeted question: at which time steps do the world model's prediction errors cause the policy to underperform in the real world? [2]

In this paper, we develop a method that answers the above question directly by allocating counterfactual Shapley credits to world model's predicted dynamics at each time step (rather than actions [24]). We treat each time step's model-predicted transition as a player in a game and measure its causal contribution to the transfer gap. We do so by constructing *counterfactual rollouts*: at each time step, we can choose whether to use the world model's prediction or the sampled real world's transition, then measure how the trajectory unfolds, and potentially changing the rewards. By systematically intervening at different subsets of time steps and measuring the resulting return differences, we obtain per-time-step counterfactual Shapley values (ϕ-values) that quantify how much each time step's dynamics causally contributes to the transfer gap. This formulation yields actionable diagnostics. The world model's prediction deviating from the real world at those most blamed steps is the root cause of the transfer gap. This also directly informs immediate remediation: steps with high attribution are where the model needs improvements, such as more training data, architectural capacity, or calibration. Specifically, our contributions are:

- We develop an *explainable* and *scalable* counterfactual debugging algorithm for attributing the world model transfer gap to individual time steps with a coarse-to-fine recursive scheme that exploits causal sparsity to remain tractable even for trajectories up to **1 million** steps.
- We develop the theoretical foundation for *faithful* world-model transfer gap attribution, establishing correctness conditions under which the newly proposed counterfactual debugging method pinpoints all causally responsible steps and allocates credits proportionally to what could have been achieved if the real world dynamics at those steps had been the same as the world model's.
- We validate the method on Atari, ProcGen, and MiniGrid environments with controlled dynamics shifts: observation corruption, physics violations, and reward misprediction, showing that it correctly localizes a variety of errors and outperforms prior methods significantly in both attribution quality and sample efficiency.

[2] See Sec. D for an extended discussion.

2 Background

In this section, we introduce the theoretical framework for explaining world model behaviors from a causal perspective. We use uppercase letters to denote random variables (X), lowercase for their realizations (x) and bold letters for sets ($\mathbf{V}$) throughout.

Structural Causal Models and Markov Decision Processes. We follow the semantics of Structural Causal Models (SCMs) to ground the discussion. SCMs provide the mathematical foundation for counterfactual reasoning needed to isolate causal contributions [3, 30]. An SCM $\mathcal{M}$ is a tuple $\langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{U}) \rangle$ where $\mathbf{V}$ represents the endogenous variables, $\mathbf{U}$ represents the exogenous variables, $\mathcal{F} = \{f_i\}$ are causal mechanisms $V_i := f_i(\mathbf{Pa}_i, \mathbf{U}_i)$, $\mathbf{Pa}_i \subseteq \mathbf{V}$ are the endogenous parents of V_i, and $P(\mathbf{U})$ represents the exogenous distribution. The counterfactual variable $V_{\mathbf{x}}(\mathbf{u})$ is the value of V when $\mathbf{X}$ is set to $\mathbf{x}$ for $\mathbf{U} = \mathbf{u}$.

In the standard reinforcement learning literature [40], the environment is usually modeled as a Markov Decision Process (MDP), which is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition probability, $\mathcal{R}$ is the reward function, and γ is the discount factor. To examine MDP-based world models from a causal perspective, we cast MDPs as SCMs. [3]

Definition 2.1 (MDP-SCM)**.** An MDP-SCM is an SCM with endogenous variables $\mathbf{V} = \{S_t, X_t, Y_t\}_{t=1}^{T} \cup \{Y\}$ (states, actions, per-step rewards, and outcome $Y = f_Y(Y_{1:T})$), mutually independent exogenous variables $\mathbf{U} = \{U_{S_t}, U_{X_t}, U_{Y_t}\}_{t=1}^{T}$, and structural equations defining system dynamics: $S_1 = f_S(U_{S_1}), S_{t+1} = f_S(S_t, X_t, U_{S_{t+1}}), X_t = \pi(S_t, U_{X_t}),\ Y_t = r(S_t, X_t, U_{Y_t})$.

We use $P(S)$ to denote the initial state distribution induced by U_{S_1} and the discounted total reward as the outcome $Y = \sum_t \gamma^{t-1} Y_t$. Value functions $V(s)$, Q-values $Q(s, x)$, and advantages $A(s, x)$ under MDP-SCM follow the standard RL definitions [40].

Counterfactual Shapley Values. Shapley values [38] distribute credit among players in cooperative games. Given players $\mathbf{X}$ and value function $f : 2^{\mathbf{X}} \to \mathbb{R}$, a *coalition* $\mathbf{Z} \subseteq \mathbf{X}$ is a subset of players, and $f(\mathbf{Z})$ measures their joint contribution. The Shapley value ϕ_t quantifies player X_t's fair share of the total value $f(\mathbf{X}) - f(\emptyset)$ with desirable properties [38], including *efficiency*: $\sum_t \phi_t = f(\mathbf{X}) - f(\emptyset)$. A commonly used representation for efficienct Shapley value computation is the kernel formulation. Let $\mathbf{z} \in \{0, 1\}^T$ denote the coalition mask ($z_t = 1$ iff $X_t \in \mathbf{Z}$), the Shapley value ϕ_t is written as:

$$\phi_t = \sum_{\mathbf{z} \in \{0,1\}^T} \kappa_t(\mathbf{z}) \cdot f(\mathbf{z}), \tag{1}$$

where $|\mathbf{z}| = \sum_t z_t$ and $\kappa_t(\mathbf{z}) = \frac{1}{T\binom{T-1}{|\mathbf{z}|-1}}$ if $z_t = 1$ while $\kappa_t(\mathbf{z}) = -\frac{1}{T\binom{T-1}{|\mathbf{z}|}}$ if $z_t = 0$.

For credit assignment, Li et al. [24] proposes to treat actions $\mathbf{X} = \{X_1, \ldots, X_T\}$ as players, and given trajectory $\tau = (s_{1:T}, x_{1:T}, y_{1:T})$ (the *evidence* for counterfactual reasoning), one can calculate Counterfactual Shapley values (ϕ-values) to quantify how much X_t contributed to outcome Y. They use counterfactual natural total effect (NTE, [23]) to measure a set of actions' causal contributions to the outcome, which is the expected difference between the observed outcome and a counterfactual outcome where a subset of actions are replaced. And it's defined as,

$$f(\mathbf{Z}) = \mathrm{NTE}(\mathbf{Z}, Y | \tau) = \mathbb{E}_{\mathbf{u}' \sim P(\mathbf{U}), \mathbf{u} \sim P(\mathbf{U}|\tau)} \big[Y(\mathbf{u}) - Y_{\mathbf{Z}(\mathbf{u}')}(\mathbf{u}) \big], \tag{2}$$

where $Y(\mathbf{u})$ is the observed outcome and $Y_{\mathbf{Z}(\mathbf{u}')}(\mathbf{u})$ is the counterfactual with actions within the coalition $\mathbf{Z}$ replaced by values sampled from a baseline policy π_{base} under $\mathbf{u}'$, while exogenous $\mathbf{u}$ for other variables is fixed. In our world model attribution setting, time steps can be thought of as the players and the total credit is the return gap of deploying the policy in the world model and the real world which is defined formally in the next section (Def. 3.1).

3 Counterfactual Attribution for World Model Transfer Gap

A visual discrepancy between the predicted and real world transitions does not always cause performance degradation, as illustrated in Fig. 1. In this section, we define such performance degradation as the world model transfer gap formally and introduce our causal attribution method for explaining it. Proof, definitions and assumptions details are provided in Sec. E.

[3] For more details, see Sec. 8 in Bareinboim [3].

3.1 The World Model Transfer Gap from A Causal Lens

We call the performance difference resulting from deploying a policy trained on the world model in the real world environment the "world model transfer gap" which we define formally as follows.

Definition 3.1 (World Model Transfer Gap)**.** Let MDP-SCM $\mathcal{M}_{\text{wm}}$ and $\mathcal{M}_{\text{env}}$ denote the world-model and real world whose transitions and rewards may differ, $f_S^{\text{wm}} \neq f_S^{\text{env}}$ and $r^{\text{wm}} \neq r^{\text{env}}$, while other components are the same. Let π^*_{wm} be an optimal policy under $\mathcal{M}_{\text{wm}}$ and trajectory $\tau \sim \mathcal{M}_{\text{env}}^{\pi^*_{\text{wm}}}$ be generated under the real world $\mathcal{M}_{\text{env}}$ following the same policy. The *world model transfer gap* is,

$$\Delta(\mathcal{M}_{\text{env}}, \mathcal{M}_{\text{wm}}) \;:=\; Y^{\tau}_{\mathcal{M}_{\text{env}}} - \mathbb{E}^{\pi^*_{\text{wm}}}_{\mathcal{M}_{\text{wm}}}[Y], \tag{3}$$

We are interested in identifying the time steps at which the transitions or rewards are causally responsible for the transfer gap between its observed outcome in the real world, $Y^{\tau}_{\mathcal{M}_{\text{env}}}$, and the expected outcome under the world model, $\mathbb{E}^{\pi^*_{\text{wm}}}_{\mathcal{M}_{\text{wm}}}[Y]$. In the environment underlying Fig. 1, the reward function r is the same but transition function f_S in the real world occasionally occludes the screen, causing the agent to miss the ball and lose points. The blackouts occur in 15 steps but only two out of the fifteen steps, 536 and 1311, are causally responsible for the gap (Fig. 3). Formally, we cast this as an attribution problem for the world model transfer gap.

Definition 3.2 (Attribution for World Model Transfer Gap, Informal)**.** A world-model transfer gap attribution function ϕ maps a trajectory, a sequence of dynamics selectors, a world model, and a policy to a real vector denoting each step's attribution to the transfer gap,

$$\phi : \tau \times \{\text{wm}, \text{env}\}^T \times \mathcal{M}_{\text{wm}} \times \Pi \times [T] \mapsto \mathbb{R}^T \tag{4}$$

where $\{\text{wm}, \text{env}\}^T$ is the sequence of dynamics selectors determining whether at time step t, $\{f_S, r\}$ is from $\mathcal{M}_{\text{wm}}$ or $\mathcal{M}_{\text{env}}$. The attribution function ϕ is causal if it satisfies the following properties,

- $\mathbf{D}_1$ **Causal Admissibility**: Steps that do not cause the performance gap have zero credit;
- $\mathbf{D}_2$ **Causal Power**: Steps that cause the performance gap are assigned non-zero credits;
- $\mathbf{D}_3$ **Causal Normality**: Steps that are more likely to affect the outcome have more credit;
- $\mathbf{D}_4$ **Causal Effect Scaling**: Steps affecting the outcome more are assigned more credits.

A formal version of the causal attribution desiderata is presented in Sec. E.1. These desiderata encode natural requirements for a causal attribution of the world model transfer gap. $\mathbf{D_1}, \mathbf{D_2}$ together guarantee that the attribution generates both sufficient and necessary allocations: steps receive credit if and only if they cause the transfer gap. $\mathbf{D_3}, \mathbf{D_4}$ require that the scale of the allocated credits reflects each step's relative contribution: using the dynamics from the world model instead of the real world at a time step that could mitigate the transfer gap more receives proportionally more credit.

We adapt the Counterfactual Shapley value framework to solve this. The key idea is to treat each step's dynamics selector as a player in a cooperative game and measure its contributions to the gap. Specifically, given a trajectory τ sampled from the real world under policy π^*_{wm} (denoted as π for simplicity), a subset of steps' causal contribution is measured by NTE,

$$f(\mathbf{Z}) = \text{NTE}(\mathbf{Z}, Y|\tau) = \mathbb{E}_{\mathbf{u}' \sim P(\mathbf{U}), \mathbf{u} \sim P(\mathbf{U}|\tau)}\left[Y^{\pi}_{\mathcal{M}_{\text{env}}}(\mathbf{u}) - Y^{\pi}_{\mathbf{Z}(\mathcal{M}_{\text{wm}}, \mathbf{u}')}(\mathbf{u})\right], \tag{5}$$

where $Y^{\pi}_{\mathcal{M}_{\text{env}}}(\mathbf{u})$ is the outcome under the real world $\mathcal{M}_{\text{env}}$ following policy π and $Y^{\pi}_{\mathbf{Z}(\mathcal{M}_{\text{wm}}, \mathbf{u}')}(\mathbf{u})$ is the counterfactual outcome with transitions/reward functions replaced by the world model $\mathcal{M}_{\text{wm}}$ at coalition steps $t, z_t = 1, z_t \in \mathbf{Z}$ with exogenous variables $\mathbf{u}'$ while other steps still follow the real world $\mathcal{M}_{\text{env}}$ and $\mathbf{u}$. This measures the difference between the return of a policy rolled out under the real world and that of when a subset of steps' dynamics are replaced by the world model. Unlike statistical or visual prediction error, this captures the *downstream causal effect* of each step's dynamics on the transfer gap: what return could have been achieved if the real world dynamics had been the same as the world model's.

Inserting Eq. (5) into the counterfactual Shapley values (ϕ-values, [24]), we obtain a causal attribution method for explaining the world-model transfer gap.

Theorem 3.3. *ϕ-values satisfy the world model transfer gap causal attribution criteria.*

To compute ϕ-values, the NTE need to be evaluated under a given coalition $\mathbf{z}$. The NTE defined in Eq. (5) averages over both posterior and prior exogenous variables which can be estimated by

the *conditional* causal contribution given trajectory τ: $f(\mathbf{z}) = Y^{\tau}_{\mathcal{M}_{\text{env}}} - Y_{\sigma_{\mathbf{z}}}$, where $Y^{\tau}_{\mathcal{M}_{\text{env}}}$ is the observed return and $Y_{\sigma_{\mathbf{z}}}$ is the counterfactual return under intervention $\sigma_{\mathbf{z}}$ which specifies step dynamics (action, reward, next state) under coalition $\mathbf{z} \in \{0,1\}^T$:

$$\sigma_{\mathbf{z}}(t) = \begin{cases} x_t^{\mathbf{z}} = x_t, r_t^{\mathbf{z}} = r_t, s_{t+1}^{\mathbf{z}} = s_{t+1} & z_t = 0,\ s_t^{\mathbf{z}} = s_t \\ x_t^{\mathbf{z}} \sim \pi, r_t^{\mathbf{z}} \sim r, s_{t+1}^{\mathbf{z}} \sim f_S, \text{ s.t. } r, f_S \in \mathcal{M}_{\text{wm}} & \text{otherwise} \end{cases} \tag{6}$$

where $s_t^{\mathbf{z}}, r_t^{\mathbf{z}}, s_{t+1}^{\mathbf{z}}$ are the counterfactual state, reward, and next state at time t. s_t, x_t, r_t, s_{t+1} are the observed state, action, reward, and next state at time t from trajectory τ. Algo. 2 implements Eq. (6).

While theoretically appealing, ϕ-values are very challenging to compute in general due to the combinatorial number of coalitions for exact evaluation and sampling variance for approximation. In the next section, we develop an efficient estimation procedure exploiting the sparsity of causal steps, making ϕ-values tractable even for extremely long trajectories up to **1 million** steps. [4]

3.2 Coarse-to-Fine Counterfactual Debugging

Naïve ϕ-value estimation treats each of the T time steps as a separate player, requiring $O(2^T)$ coalition evaluations. Even with coalition sampling, accurate estimation is still computationally expensive [23, 24]. When causal responsibility is sparse: only a few time steps actually drive the transfer gap, most of this computation is wasted on steps of the trajectory that contribute negligibly.

We exploit this sparsity with a *coarse-to-fine* recursive scheme (Algo. 1, RECURSIVE-ϕ). The trajectory is initially partitioned into B contiguous blocks, each treated as a single composite player (line 3). ϕ-values are estimated over this reduced B-player game by Eq. (1) and Eq. (5) (line 5). Blocks whose per-step attribution $|\Phi_j|/|\mathcal{B}_j|$ falls below a pruning threshold ϵ are declared inactive and excluded from further computation (line 4, 6-7). The remaining active blocks are subdivided by a factor of B and the process repeats, progressively refining resolution only where the block-level attribution is non-negligible.

Example 1 (RECURSIVE-ϕ in Atari Pong.)**.** In Fig. 1 example, the buggy Pong trajectory (rollout) spans $T \approx 1750$ environment steps with two causal blackouts (the occlusions that account for the $21 \to 15$ return drop) alongside several non-causal blackouts. With a split factor $B = 2$, round 0 partitions the trajectory into two halves and estimates each block's attribution; any block containing only non-causal steps yields negligible attribution and is pruned, because intervening on a non-causal blakcout leaves the two causal blackouts in place and the return unchanged. Each subsequent round halves whatever blocks still carry signal, recursively pruning sub-blocks of non-causal steps at every resolution. The algorithm isolates the two causal steps after $\lceil \log_2 1750 \rceil = 11$ rounds, spending a total budget that scales with the number of causal steps $K = 2$ rather than the length T.

Design choices. We make the following design choices for RECURSIVE-ϕ.

Block granularity and split factor. The split factor B controls the resolution doubling rate. With $B = 2$ (binary splitting), the algorithm runs for at most $\lceil \log_2 T \rceil$ rounds. Each round re-estimates ϕ-values only over the active blocks, pruning inactive blocks, so the effective number of players n is bounded by the split B times the number of active individual players remaining. When n is small enough that $2^n \leq M$, the algorithm switches from sampling to exact enumeration under counterfactual simulation rollout budget M, eliminating sampling variance entirely for this round.

Pruning threshold. The threshold ϵ governs the sparsity–accuracy trade-off. A block is pruned when its per-step attribution magnitude is below ϵ, which sets the attribution of all its constituent time steps to zero. In the sparse-causality regime, where only $K \ll T$ time steps have non-negligible causal effect, aggressive pruning reduces the effective problem size from T to $O(K)$ players within a few rounds, yielding a total cost of $O(2^K \log_B T)$ counterfactual simulations rather than $O(2^T)$.

Coalition sampling within each round. At each resolution level, if the total number of possible coalitions is less than the coalition budget M, we enumerate all coalitions to calculate ϕ-values exactly. Otherwise, we use the optimal proposal distribution Q_n^* [24] to sample coalitions over the n active blocks. Either way, each coalition $\mathbf{z} \in \{0,1\}^n$ is expanded to a full T-length mask by broadcasting: all time steps within an active block receive the same intervention indicator.

[4] Even SOTA Genie model only runs at 24 FPS, retaining consistency for a few minutes, roughly 14K steps [7].

Algorithm 1 RECURSIVE-ϕ

Input: Trajectory τ (length T), policies π, world model $\mathcal{M}_{\text{wm}}$, budget M, split B, threshold ϵ
Output: Per-step attributions $\hat{\phi}_{1:T}$

Initialize $\hat{\phi}_{1:T} \leftarrow \mathbf{0}$, all steps active
for $r = 0, 1, \ldots, \lceil \log_B T \rceil - 1$ **do**
  Partition $\{1, \ldots, T\}$ into $\min(B^{r+1}, T)$ contiguous blocks
  Identify active blocks $\mathcal{A}$ (those containing at least one active step)
  Estimate block-level ϕ-values $\{\Phi_j\}_{j \in \mathcal{A}}$ by Eq. (1) and Eq. (6)
  Distribute: $\hat{\phi}_t \leftarrow \Phi_j / |\mathcal{B}_j|$ for $t \in \mathcal{B}_j$, $j \in \mathcal{A}$
  Prune: deactivate steps in block j if $|\Phi_j|/|\mathcal{B}_j| < \epsilon$
end for

return $\hat{\phi}_{1:T}$

Correctness of Coarse-to-Fine Estimation. The coarse-to-fine scheme is sound whenever the causal steps are sparse and do not cancel out other's effect on the transfer gap when grouped into a block. Concretely, we define a sparse transfer gap attribution problem where at most K steps carry a meaningful signal of magnitude $\geq \delta$, and the rest are exactly zero (Def. E.3). Two conditions then suffice to show the correctness of the algorithm. (i) *Bounded interaction* (Assumption E.4): the block-level ϕ-values Φ_j deviates from the sum of its per-step values by at most η. This holds with $\eta \ll \delta$ in our setting, because non-causal steps contribute zero to the transfer gap in every coalition, effectively killing all interaction terms that involve them [43]. (ii) *Sign consistency* (Assumption E.5): causal steps share a sign, which is natural because systematic world-model errors compound along a rollout rather than cancel [22]. Under these conditions, the algorithm provably finds what it should:

Theorem 3.4 (Correctness of Algo. 1, informal)**.** *Under the conditions above, with a small enough threshold ϵ, with high probability, Algo. 1 (a) loses no causal step, (b) pinpoints all K of them in $\log_B T$ rounds, and (c) recovers their attributions up to bounded errors while zeroing out the rest.*

In words: as long as the causally responsible dynamics errors are few, sizeable, and compound with time, the recursive scheme zooms in on them in logarithmically many rounds without losing any along the way, and shrinks to exact enumeration once the surviving sub-game is small.

4 Experiments

We evaluate RECURSIVE-ϕ on Atari [6], ProcGen [10], and MiniGrid [9] with controlled world-model failures, answering three questions: **(Q1)** can RECURSIVE-ϕ localize the time steps which are causally responsible for the transfer gap; **(Q2)** does RECURSIVE-ϕ explain more transfer gap than other baselines; **(Q3)** does RECURSIVE-ϕ scale to long trajectories with limited number of simulations (rollouts) when baselines can not? All results are averaged over 10 coalition-sampling seeds with environment seeds held fixed for controlled world-model failures. We compare the proposed counterfactual debugging method, RECURSIVE-ϕ against a causal baseline FLAT-ϕ (Algo. 3) which is a non-recursive naïve implementation of the ϕ-value estimation and two non-causal baselines SHAP [25] and LIME [33]. The details are provided in Sec. B; extended results are in Sec. C.

Sim-to-sim protocol. For the ease of testing controlled errors, rather than training a world model, we simulate real world vs. world model discrepencies by modifying the simulator: an unchanged base simulator serves as the world model $\mathcal{M}_{\text{wm}}$, and a perturbed simulator $\mathcal{M}_{\text{env}}$ stands for the real world with unexpected events (perturbations). A pretrained policy π from $\mathcal{M}_{\text{wm}}$ rolls out in $\mathcal{M}_{\text{env}}$; the goal is to causally attribute the transfer gap (Def. 3.1) to individual steps. Specfically, we evaluate each method by if removing the top K (known by design) detected steps can close the transfer gap.

4.1 Experiment 1: Observation Blackouts on Atari Pong

We test our method on Atari Pong with a pretrained PPO agent [36]. At randomly selected time steps the observation emitted by $\mathcal{M}_{\text{env}}$ is zeroed out, simulating a real world that experiences unexpected

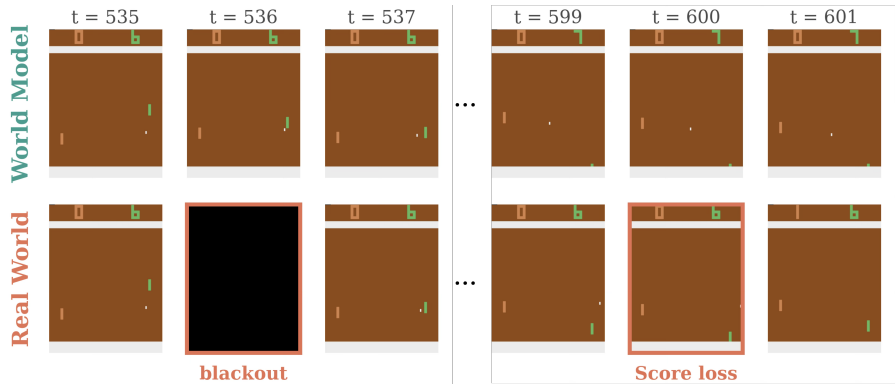


Figure 2: **Observation blackout on Pong.** Top row: the world model $\mathcal{M}_{\text{wm}}$ observations. Bottom row: real world $\mathcal{M}_{\text{env}}$ has blackouts at $t = 536$ but score loss does not surface until $t = 600$.

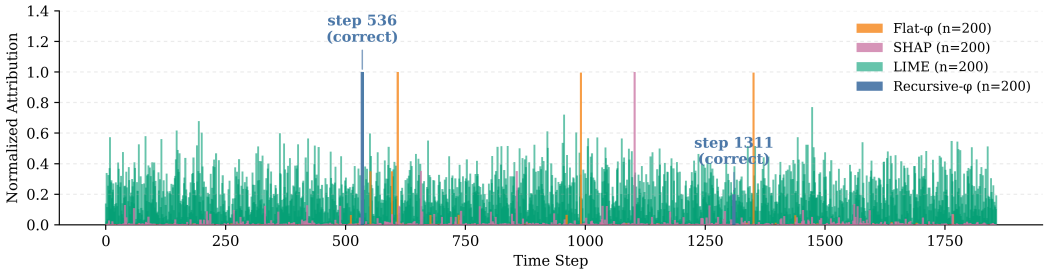


Figure 3: **Localization of causally important steps.** Only RECURSIVE-ϕ (blue) correctly captures the two causally important steps 536 and 1311.

sensor disconnections. Fig. 2 illustrates the error and its delayed effect: the causally relevant blackout happens at $t = 536$ yet the observable score loss does not surface until $t = 600$.

Results. *(i) localization (Q1).* Fig. 3 shows the per-step $|\phi|$-value trace on the full ~ 1860-step Pong trajectory at $L = 2000$: RECURSIVE-ϕ concentrates all mass on the two causal blackouts at steps 536 and 1311 with every other steps at zero. Other baselines' attribution spikes scatter across the horizon, completely miss the two causally important steps. *(ii) Attribution quality (Q2).* The top half of Table 1 reports the fraction of the transfer gap recovered by removing each method's top-K blackouts: RECURSIVE-ϕ explains 100% at every length, while FLAT-ϕ matches at $L \leq 300$ then collapses (10% at $L = 500$, 0% beyond) once the 5k-rollout sweep can no longer separate the two causal sites from sampling noise. The non-causal baselines fare worse still: SHAP recovers only at $L = 100$ and is at or below 20% from $L = 300$ onward, while LIME never localizes the causal blackouts at any L. *(iii) Sample efficiency (Q3).* The bottom half of Table 1 gives the smallest budget at which the gap is fully explained: RECURSIVE-ϕ's cost grows mildly with L and has *no* spread across 10 seeds (pruning collapses the inactive horizon and the surviving sub-game switches to exact enumeration), whereas FLAT-ϕ already needs $>25\times$ more at $L \in \{100, 300\}$, SHAP needs $>35\times$, and both time out from $L = 500$; LIME times out at every L. The gap widens monotonically with L.

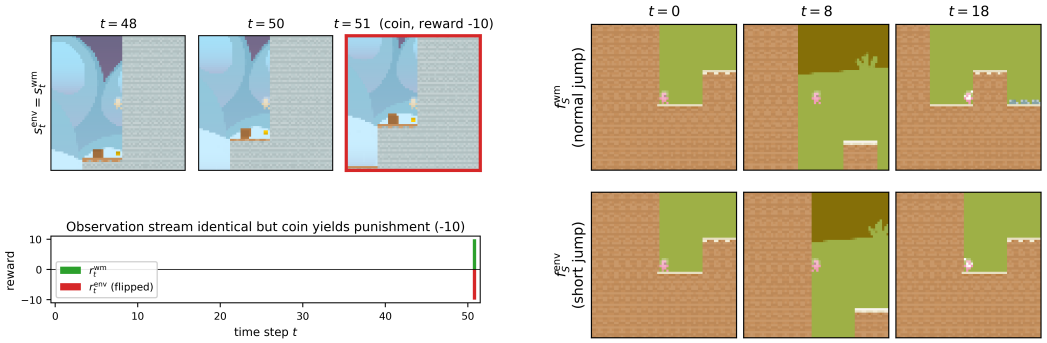
4.2 Experiment 2: Mechanistic Shifts on ProcGen CoinRun

We move from observational to mechanistic shifts on ProcGen CoinRun with a trained PPO policy under two variants with unexpected reward and transition dynamics shifts, respectively. *Variant A (poisonous coin):* every time a coin is collected, the wrapper draws $\text{Bernoulli}(p_A)$ and, on success, flips the reward sign (-10). The observation stream stays pixel-identical. *Variant B (windy day):* at every `UP`-containing action (jump) the wrapper draws $\text{Bernoulli}(p_B)$ and, on success, applies a horizontal wind that pushes the agent to the left, so the jump falls short. Fig. 4 illustrates both.

Results. *(i) Localization (Q1).* The per-step $|\phi|$ traces in Figs. 9 and 10 (Sec. C.3) show RECURSIVE-ϕ concentrating its mass on the realized causal steps, while FLAT-ϕ, SHAP, and LIME all rank non-causal steps above the causally important ones. *(ii) Attribution quality (Q2).* RECURSIVE-ϕ

Table 1: **Attribution quality and sample efficiency on Atari Pong.** Results are aggregated over 10 seeds; "-" marks cells where not enough seeds succeed within the rollout budget (5k) to derive meaningful statistics.

Trajectory length L	100	300	500	1000	3000
transfer gap explained (%, ↑)					
RECURSIVE-ϕ (ours)	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$
FLAT-ϕ	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	10 ± 32	0 ± 0	0 ± 0
SHAP	$\mathbf{100 \pm 0}$	20 ± 42	0 ± 0	0 ± 0	0 ± 0
LIME	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Rollouts required for successful attribution (↓)					
RECURSIVE-ϕ (ours)	$\mathbf{28 \pm 0}$	$\mathbf{34 \pm 0}$	$\mathbf{132 \pm 0}$	$\mathbf{148 \pm 0}$	$\mathbf{156 \pm 0}$
FLAT-ϕ	750 ± 264	4200 ± 1687	-	-	-
SHAP	1020 ± 590	-	-	-	-
LIME	-	-	-	-	-



(a) Variant A (poisonous coin).

(b) Variant B (windy day).

Figure 4: **CoinRun variants.** Both are visually plausible yet incur a large transfer gap.

recovers 100% of the transfer gap on every (variant, L) setting. FLAT-ϕ and SHAP track each other closely on Variant A; on Variant B they degrade as L grows, with SHAP collapsing faster. LIME never exceeds 30% on any experiment setting and already times out at $L \geq 300$ on Variant A. *(iii) Sample efficiency (Q3).* RECURSIVE-ϕ uses 30–300 rollouts with zero seed variance (pruning collapses the horizon to a sub-game small enough for exact enumeration), whereas FLAT-ϕ and SHAP are variance-dominated and need 20–100$\times$ more rollouts when they recover at all. LIME succeeds on at most $3/10$ seeds in any cell, requiring the full 5k cap on the one Variant A setting where it recovers.

4.3 Experiment 3: Sample Efficiency at Scale on MiniGrid

Experiment 3 focuses on sample efficiency (**Q3**) and pushes the length (L) of the trajectory to be explained from thousands of steps in previous experiments to **1 million** steps. We design a custom MiniGrid `CenterGoal5x5` (Fig. 5): a 5×5 room walled into a walkable interior with the goal at the center. A hand-crafted policy loops the 8-cell inner ring for n steps before heading to the goal; varying n controls L. The perturbation is a *recurring* reward error where a fixed penalty is subtracted at $K = 1, 2, 5$ randomly sampled time steps, and the ground-truth attribution set is exactly those steps. The deliberate simplicity (the causally important steps are known) lets us isolate *how the rollout budget grows with horizon*, independent of any modelling noise.

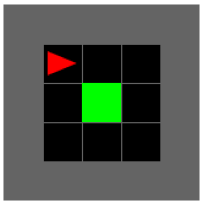
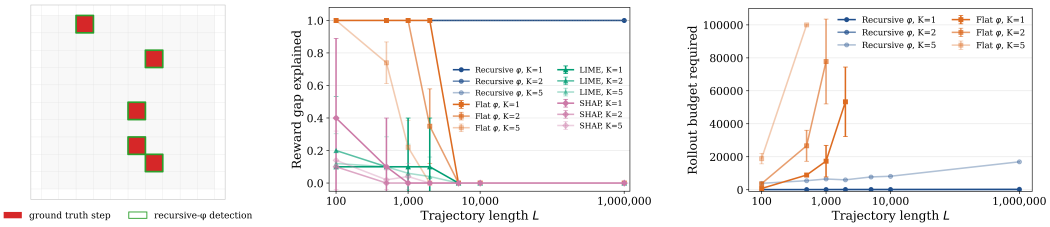


Figure 5: A custom MiniGrid world.

Results (Fig. 6). The example pixel map in Fig. 6a visualises the per-step $|\phi|$ trace for $K \in \{1, 2, 5\}$ at $L = 100$: RECURSIVE-ϕ places all mass exactly on the planted penalty steps with the rest at zero, confirming the accurate localization of RECURSIVE-ϕ. Fig. 6b shows that RECURSIVE-ϕ recovers 100% of the transfer gap at *every* L up to 1M across all number of perturbations K, whereas FLAT-ϕ matches only at $L = 100$ and is exhausted past $L = 2000, K = 5$ even at the largest budget we sweep; the non-causal baselines never exceed $\sim$20% gap-explained at any (L, K) and succeeds on

Table 2: **Attribution quality and sample efficiency on CoinRun variants.** Results are aggregated over 10 seeds; "-" marks cells where not enough seeds succeed within the rollout budget (5k) to derive meaningful statistics.

Trajectory length L	100	300	500	1000
transfer gap explained (%, ↑)				
Variant A: RECURSIVE-ϕ (**ours**)	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$
Variant A: FLAT-ϕ	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	0 ± 0	0 ± 0
Variant A: SHAP	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	0 ± 0	0 ± 0
Variant A: LIME	10 ± 9	0 ± 0	0 ± 0	0 ± 0
Variant B: RECURSIVE-ϕ (**ours**)	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$
Variant B: FLAT-ϕ	$\mathbf{100 \pm 0}$	90 ± 9	40 ± 15	20 ± 13
Variant B: SHAP	$\mathbf{100 \pm 0}$	50 ± 16	20 ± 13	20 ± 13
Variant B: LIME	30 ± 14	10 ± 9	30 ± 14	0 ± 0
Rollouts required for successful attribution (↓)				
Variant A: RECURSIVE-ϕ (**ours**)	$\mathbf{28 \pm 0}$	$\mathbf{34 \pm 0}$	$\mathbf{132 \pm 0}$	$\mathbf{136 \pm 0}$
Variant A: FLAT-ϕ	620 ± 114	3800 ± 611	-	-
Variant A: SHAP	800 ± 82	3950 ± 550	-	-
Variant A: LIME	-	-	-	-
Variant B: RECURSIVE-ϕ (**ours**)	$\mathbf{40 \pm 0}$	$\mathbf{60 \pm 0}$	$\mathbf{276 \pm 0}$	$\mathbf{76 \pm 0}$
Variant B: FLAT-ϕ	810 ± 165	1711 ± 653	3125 ± 1125	-
Variant B: SHAP	1080 ± 271	1620 ± 863	-	-
Variant B: LIME	467 ± 267	-	367 ± 133	-



(a) Per-step pixel map. (b) Transfer gap explained vs L. (c) Rollouts required vs L.

Figure 6: **Scaling to $L = 1M$ on MiniGrid `CenterGoal`.**

fewer than $2/10$ seeds throughout, so we omit their budget curves from Fig. 6c as the means are not statistically meaningful. The decisive comparison is the rollout cost (Fig. 6c) against the other causal baseline, FLAT-ϕ, whose budget scales near exponentially with L and blows past 10^5 rollouts already at $L = 500$ for $K = 5$, while recursive's budget grows essentially logarithmically. This matches the $O(2^K \log_B T)$ cost of Algo. 1: pruning collapses the inactive horizon round-by-round and the surviving budget concentrates on the K active steps, so a $10^4\times$ increase in L costs less than a $5\times$ increase in rollouts budget. RECURSIVE-ϕ therefore demonstrates the strong scaling property for trajectories up to **1M steps** that are structurally inaccessible to prior methods.

5 Conclusions

We introduced a framework for world models transfer gap attribution by identifying which time steps are *causally responsible* for the policy's real world performance degradation. By treating each time step's dynamics selector as a player in a game and computing counterfactual Shapley values over hybrid rollouts that selectively replace simulated/world-model predictions with turbulant real world transitions, our method produces precise per-step causal attributions that explains the return gap. The newly proposed RECURSIVE-ϕ is shown to be more sample efficient and scales to long trajectories by exploiting the sparsity of causal errors with a coarse-to-fine recursive estimation scheme. Experiments on Atari, ProcGen and MiniGrid environments with controlled failures demonstrate that RECURSIVE-ϕ correctly identifies the steps that are causally important. These results corroborate with the central premise of our work: not all model errors matter equally, and causal reasoning is necessary to distinguish the critical few from the irrelevant many. Our work paves the way for more principled policy transfer from world models to real world.

Acknowledgments and Disclosure of Funding

This research is supported in part by the NSF, ONR, AFOSR, DoE, Amazon, JP Morgan, The Alfred P. Sloan Foundation and MATS. We would also like to thank Jeffrey Heninger from MATS for his thoughtful review and suggestions.

References

[1] O. Ahmed, F. Träuble, A. Goyal, A. Neitz, M. Wuthrich, Y. Bengio, B. Schölkopf, and S. Bauer. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=SK7A5pdrgov.

[2] E. Alonso, F. Fleuret, A. Jelley, A. Kanervisto, V. Micheli, T. Pearce, and A. Storkey. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, page 58757–58791. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-1873. URL http://dx.doi.org/10.52202/079017-1873.

[3] E. Bareinboim. *Causal Artificial Intelligence: A Roadmap for Building Causally Intelligent Systems*. 2026. URL https://causalai-book.net.

[4] E. Bareinboim, J. Zhang, and S. Lee. An introduction to causal reinforcement learning. Technical Report R-65, Causal Artificial Intelligence Lab, Columbia University, December 2024.

[5] D. Beechey, T. M. S. Smith, and O. Şimşek. Explaining reinforcement learning with shapley values. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[6] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013. doi: 10.1613/jair.3912. URL https://doi.org/10.1613/jair.3912.

[7] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[8] L. Buesing, T. Weber, Y. Zwols, S. Racanière, A. Guez, J.-B. Lespiau, and N. M. O. Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *ArXiv*, abs/1811.06272, 2018. URL https://api.semanticscholar.org/CorpusID:53438249.

[9] M. Chevalier-Boisvert, L. Willems, and S. Pal. Minimalistic gridworld environment for gymnasium, 2018. URL https://github.com/Farama-Foundation/Minigrid.

[10] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2048–2056. PMLR, 2020. URL http://proceedings.mlr.press/v119/cobbe20a.html.

[11] J. García and F. Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16:1437–1480, 2015. URL https://dl.acm.org/doi/10.5555/2789272.2886795.

[12] M. Grabisch and M. Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999. doi: 10.1007/s001820050125. URL https://doi.org/10.1007/s001820050125.

[13] S. Greydanus, A. Koul, J. Dodge, and A. Fern. Visualizing and understanding atari agents. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1787–1796. PMLR, 2018. URL http://proceedings.mlr.press/v80/greydanus18a.html.

[14] A. H. Güzel, M. T. Jackson, J. L. Liesen, T. Rocktäschel, J. N. Foerster, I. Bogunovic, and J. Parker-Holder. Imagined autocurricula. 2025.

[15] D. Ha and J. Schmidhuber. World models. 2018.

[16] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2455–2467, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

[17] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

[18] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering Atari with Discrete World Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

[19] B. Hou, G. Li, J. Jia, T. An, X. Guo, S. Leng, H. Geng, Y. Ze, T. Harada, P. Torr, O. Mees, M. Pollefeys, Z. Liu, J. Wu, P. Abbeel, J. Malik, Y. Du, and J. Yang. World model for robot learning: A comprehensive survey. 2026.

[20] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

[21] W. Huang, J. Ji, C. Xia, B. Zhang, and Y. Yang. Safedreamer: Safe reinforcement learning with world models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=tsE5HLYtYg.

[22] M. Janner, J. Fu, M. Zhang, and S. Levine. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12498–12509, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

[23] K.-Z. Lee, D. Plecko, and E. Bareinboim. Causal explanations through counterfactual variable attributions. Technical Report R-135, Causal Artificial Intelligence Lab, Columbia University, May 2025. URL https://causalai.net/r135.pdf.

[24] M. Li, K.-Z. Lee, and E. Bareinboim. Counterfactual shapley credit assignment. In *Reinforcement Learning Conference*, 2026. URL https://openreview.net/forum?id=qmCAKnqN1u.

[25] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*, 2017.

[26] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere. Explainable reinforcement learning through a causal lens. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2493–2500. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/5631.

[27] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand. arxiv, 2019. URL http://arxiv.org/abs/1910.07113.

[28] G. Owen. Multilinear extensions of games. *Management Science*, 18(5):P64–P79, 1972. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2661445.

[29] K. Panaganti, Z. Xu, D. Kalathil, and M. Ghavamzadeh. Robust reinforcement learning using offline data. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=AK6S9MZwM0.

[30] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161.

[31] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, 2017.

[32] A. Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *North American Chapter of the Association for Computational Linguistics*, 2016.

[34] M. Rigter, M. Jiang, and I. Posner. Reward-free curricula for training robust world models. In *International Conference on Learning Representations*, 2024.

[35] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2019. ISSN 1476-4687. doi: 10.1038/s41586-020-03051-4. URL https://www.nature.com/articles/s41586-020-03051-4.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arxiv, 2017. URL http://arxiv.org/abs/1707.06347.

[37] L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton, 1953.

[38] L. S. Shapley et al. A value for n-person games. *Annals of Mathematics Studies*, 28:307–318, 1953.

[39] R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.

[40] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, second edition, 2018. ISBN 0262039249.

[41] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2017.

[42] M. Towers, A. Kwiatkowski, J. K. Terry, J. U. Balis, G. D. Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *CoRR*, abs/2407.17032, 2024. doi: 10.48550/ARXIV.2407.17032. URL https://doi.org/10.48550/arXiv.2407.17032.

[43] C.-P. Tsai, C.-K. Yeh, and P. Ravikumar. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023. URL http://jmlr.org/papers/v24/22-0202.html.

[44] W. Wiesemann, D. Kuhn, and B. Rustem. Robust markov decision processes. *Math. Oper. Res.*, 38(1):153–183, 2013. doi: 10.1287/MOOR.1120.0566. URL https://doi.org/10.1287/moor.1120.0566.

[45] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, 2022.

[46] H. Xue, T. He, Z. Wang, Q. Ben, W. Xiao, Z. Luo, X. Da, F. Castañeda, G. Shi, S. Sastry, L. J. Fan, and Y. Zhu. Opening the sim-to-real door for humanoid pixel-to-action policy transfer. 2025.

[47] S. Yang, Y. Du, K. Ghasemipour, J. Tompson, L. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2023.

[48] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. S. Boning, and C. Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f0eb6568ea114ba6e293f903c34d7488-Abstract.html.

Appendices

Contents

A Limitations and Future Work

A.1 Limitations.

Our method's computational cost, though reduced by coarse-to-fine estimation, still scales with the number of causal steps K, which may be prohibitive for very long trajectories with dense causal structure (i.e., when the sparsity assumption is violated). Additionally, the current framework assumes sign consistency of the causal steps which may not hold if within a rollout, there are both real world transition discrepancies that help the agent and those that hurt it. But the assumption usually holds empirically because systematic world model errors usually lead to performance loss rather than gain under a well-defined task reward function and an optimized policy under the world model. From a higher conceptual level, we assume the environment and world model can be described by MDP-SCMs while in practice, confounders are pervasive and non-MDP models are prevalent in decision making tasks [4]. For the world model transfer gap attribution under more general environment models beyond MDP-SCMs is an interesting direction for future work.

A.2 Future directions.

Several extensions are worth pursuing. First, applying the method to learned world models trained end-to-end (e.g., latent dynamics models) would test whether the attributions remain informative when the model's state representation differs from the simulator's. Second, the ranked attribution could be used to drive *active model improvement*: collecting additional training data specifically at the high-attribution steps to improve the world model where it matters most. Third, integrating the diagnostic into online model-based planning: flagging high-attribution predictions before the agent commits to a plan could yield a safety mechanism that detects when the world model's errors are likely to cause real-world failures.

B Detailed Experiment Setup

Common Protocol. All three experiments share the sim-to-sim protocol of Sec. 4: an unperturbed simulator stands in for the world model $\mathcal{M}_{\text{wm}}$, a wrapped simulator with controlled error stands in for the real environment $\mathcal{M}_{\text{env}}$, and the realised perturbation set is read off the wrapper as the mechanistic ground truth. Counterfactual rollouts use ALE's `clone_state`/`restore_state` API for Atari, ProcGen `gym3` state cloning for CoinRun, and direct state serialisation for MiniGrid; in all three cases the on-track branch of CTFSIM (Algo. 2) reuses the observed transition exactly, while off-track transitions are resampled from P_{env}. The self-baseline $\pi_{\text{base}} = \pi$ is used throughout, so ϕ-values measure whether each transition's deviation from the ground truth causally affected the return. Every cell is repeated over 10 independent coalition-sampling seeds with the environment seed and perturbation set held fixed, so per-seed variation in the flat estimators (FLAT-ϕ, SHAP, LIME) is averaged out rather than masked by a lucky draw. RECURSIVE-ϕ uses $M = 100$ coalitions per refinement round, split factor $B = 2$, pruning threshold $\epsilon = 10^{-10}$, discount $\gamma = 1$, and the optimal proposal $Q_n^\star$ of Li et al. [24] on each surviving sub-game; once $2^{|\mathcal{A}|} \leq M$ the algorithm switches to exact enumeration on the remaining active set, eliminating sampling variance. FLAT-ϕ uses the same $Q_T^\star$ proposal but allocates the full coalition budget over the T-player game in a single round; SHAP [25] and LIME [33] use the same T-player game but factual rollouts and the other thing that changes is how the value-gap signal $Y - Y^{\mathbf{z}}$ is converted into a per-step score (kernel-SHAP weights for FLAT-ϕ and SHAP, sparse local linear regression for LIME).

SHAP and LIME for Transfer Gap Attribution. We adapt SHAP and LIME to the problem of transfer gap attribution by treating the given real world rollout as the baseline instance $\{0\}^T$. And the influence function of the input instance is given by rolling out the policy again in the real world. Setting/Perturbing any of the steps to be 1 in the input instance means that we don't inject errors at that time step. For those steps with 0s, we randomly drawn with another seed (could be different from the seed used by the given trajectory) to decide whether an error is injected. The major difference between this factual simulation and the counterfactual simulation is that the environment randomness is not held the same as the one in the given trajectory. Thus, attribution methods based on such factual simulations evaluated influence functions cannot provide the causal attribution for a specific trajectory instance at all, as we see from the experiments.

Sample-efficiency Metric. For each method we sweep the coalition / sample budget over a grid over trajectory length and effective number of errors (K) (and environment variants for CoinRun) and record the smallest budget at which removing the method's top-K steps from the injected set restores the clean return on a re-rollout. K is identified by design or from RECURSIVE-ϕ outputs and verified before using. "-" in Tables 1 and 2 marks cells where fewer than three of the ten coalition seeds recover within the largest budget tried, so the conditional mean is not statistically meaningful.

Experiment 1 Setup (Sec. 4.1). We use `PongNoFrameskip-v4` via Gymnasium [42] with the standard Atari wrappers (`frameskip=4`, frame-stack $k = 4$, 84×84 grayscale) and the RL-Zoo3 PPO agent [32, 36] acting deterministically (argmax over logits, so identical observations yield identical actions across coalition rollouts). We sweep trajectory length $L \in \{100, 200, 300, 500, 1000, 3000\}$ with environment seed 100. The perturbation is an observation blackout: at a fixed set of step indices the wrapper zeroes the stacked-frame observation passed to the policy. The injected indices are drawn by shuffling $\{0, \ldots, L-1\}$ and slicing a length-dependent window of that shuffle ($|P| = 10$ for $L \in \{100, 200, 300, 500\}$ and $|P| = 20$ for $L \in \{1000, 3000\}$); fixing both shuffle seed and window keeps the perturbation set identical across coalition seeds and methods. Most blackouts are absorbed by the next few paddle moves, so the causal subset $K \leq |P|$ is recovered per trajectory as the smallest top-K ranking under recursive $|\phi|$ whose removal from the injected set restores the clean return; the same K is then handed to the flat / SHAP / LIME baselines for an apples-to-apples comparison. FLAT-ϕ and SHAP are swept over coalition budgets $M \in \{100, 200, 500, 1000, 2000, 5000\}$; LIME is swept over the same sample-count grid. The SHAP sweep is restricted to $L \leq 500$ and the LIME sweep is run at every L, but in both cases the longer-L cells exhaust the 5k cap before recovering on any seed (consistent with the "-" entries in Table 1).

Experiment 2 Setup (Sec. 4.2). CoinRun [10] with a deterministic argmax cleanba-PPO policy (checkpoint from cleanrl, 15 discrete actions, $3 \times 64 \times 64$ RGB observations) and trajectory lengths $N \in \{100, 300, 500, 1000\}$. Both variants are stochastic wrappers around the unmodified ProcGen `gym3` simulator: *Variant A (poisonous coin)* draws $\text{Bernoulli}(p_A)$ at every coin pickup and on success replaces the $+10$ reward with -10, leaving the pixel stream unchanged; *Variant B (windy day)* draws

Bernoulli(p_B) at every `UP`-containing action and on success applies a one-cell horizontal wind pushing the agent left, so the jump falls short. We use $p_A = 1.0$ (every coin flip fires) and $p_B = 0.7$, giving realised perturbation counts in the tens at $N = 1000$ on Variant B but a much smaller causal support because most short jumps are absorbed by the policy. The realised perturbation set is read off the wrapper attribute as ground truth; for the recovery metric a repair rollout replays the same wrapper with perturb_steps $= P \setminus$ top-K and the budget is recorded as recovered when its return matches the clean return exactly. FLAT-ϕ, SHAP, and LIME are each swept over the budget grid $\{100, 200, 500, 1000, 2000, 5000\}$; RECURSIVE-$\phi$ uses budget $M = 100$ per round.

Experiment 3 Setup (Sec. 4.3). The custom `MiniGrid-CenterGoal5x5-v0` environment is a 5×5 room with the goal at the centre; a hand-crafted policy executes the deterministic clockwise pattern `[forward, forward, right]` for n inner-ring steps, then takes the BFS shortest path to the goal. Total trajectory length $T = n +$ (steps to goal), so $n \in \{100, 500, 1000, 2000, 5000\}$ for SHAP, LIME and FLAT-ϕ, and additionally $\{10\,000, 10^6\}$ only for RECURSIVE-ϕ (which is cheap enough to push to $T = 1$M and is the only curve drawn in Fig. 6c at the very end). The perturbation is a deterministic time penalty applied by a `TimePenaltyWrapper`: a constant -0.1 reward is subtracted at exactly $K \in \{1, 2, 5\}$ pre-committed randomly sampled time steps. K is exactly the cardinality of the causal set and the recovery / recall-at-K metric reduces to checking whether the top-K of the attribution output equals the planted set. FLAT-ϕ, SHAP, and LIME share the budget grid $\{100, 500, 1000, 4000, 8000, 10000, 20000, 40000, 60000, 100000\}$; RECURSIVE-$\phi$ uses $M = 100$ per round, so its total counterfactual cost is $\leq M \cdot \lceil \log_2 T \rceil$ rather than $M \cdot T$.

C Extended Experiment Results

We mainly provide extra experiment results in comparing FLAT-ϕ against RECURSIVE-ϕ because from the main experiment section, both non-causal baselines perform poorly due to their inability to pinpoint causal steps using associational influence functions.

C.1 Atari Pong: detection traces across coalition budgets

Fig. 3 in the main paper reports the per-step $|\phi|$ trace at a single flat budget ($M = 2000$). To make the comparison less anecdotal, Fig. 7 sweeps the coalition budget $M \in \{200, 500, 1000, 2000\}$ on the same $\sim$1860-step Pong rollout, with the same two ground-truth causal blackouts at steps 536 and 1311 and the same eight non-causal blackouts. Recursive uses the budget once per refinement round (with $B = 2$, $\epsilon = 10^{-10}$); flat uses the full budget over the T-player game.

Recursive is budget-stable. Across all four panels the recursive trace looks essentially identical: a saturated spike at step 536 and a smaller but isolated spike at step 1311, with every other position at machine zero. The reason is structural rather than statistical, once the coarse-to-fine pruning collapses the inactive bulk of the horizon (after the first two or three rounds), the surviving sub-game is small enough that any of the budgets in $\{200, 500, 1000, 2000\}$ already exceeds $2^{|\mathcal{A}|}$ and the algorithm switches to exact enumeration on the remaining blocks. Increasing M buys nothing once enumeration kicks in, which is why the recursive trace is visually invariant to M in this regime. The two causal sites are the unambiguous top-2 at every budget, so top-K recovery of the clean return is achieved already at $M = 200$.

Flat improves but does not converge in this range. Flat's trace, in contrast, changes substantially with M and never localises both causal sites cleanly. At $M = 200$ several non-causal positions (≈ 600, ≈ 660, ≈ 970) saturate at the per-method max and dwarf the causal site at step 1311, so the flat top-2 misses both ground-truth blackouts. At $M = 500$ the dominant spurious spikes shift (now ≈ 60, ≈ 125, ≈ 1700), step 536 rises into the top tier, but step 1311 remains buried under multiple non-causal peaks. At $M = 1000$ a single non-causal position near step 810 still saturates, and although step 536 is now reliably among the tallest spikes, step 1311 is comparable to four or five distractors and would still be missed by a top-2 readout. At $M = 2000$ flat is closest to recursive, step 536 is among the largest peaks and step 1311 is visible, but a non-causal spike near step 150 still saturates and the gap between causal and non-causal mass is not yet decisive. The per-method max-normalisation in each panel hides the absolute shrinkage of flat's variance with M, but what matters operationally, the *ranking* of causal vs. non-causal positions, only flips reliably at budgets well above 2000 (consistent with the $\geq 25\times$ gap reported in Table 1).

Implication for budget selection. The qualitative picture across the four panels reinforces the takeaway of the main-paper sample-efficiency curve: the practical question is not "how large does M need to be for flat to converge" but "does the coarse-to-fine scheme reach exact enumeration on the surviving sub-game." Once it does, recursive's output is deterministic up to the on-track coupling and increasing M has no effect; until flat does, its top-K readout is governed by sampling variance and can rank a non-causal blackout above a causal one even at $M = 2000$ on a horizon of less than 2000 steps. This is exactly the regime that motivates the recursive scheme.

C.2 Atari Pong: detection traces across trajectory lengths

Table 1 reports the headline sample-efficiency numbers from the L-sweep as a single point per length. Fig. 8 unpacks them into per-step $|\phi|$ traces (mean $\pm$ 1 SE over 10 coalition-sampling seeds; env seed and perturbation set held fixed within each length). Red ticks mark every injected blackout, the green dashed marker(s) the causal subset whose removal restores the clean return, and each panel header records the gap, the recovered K, and the minimum flat budget over the 10 seeds.

Sparse causal support persists as L grows. Although $|P| \in \{10, 20\}$ blackouts are injected at every length, only $K \in \{1, 1, 1, 1, 2\}$ are causal at $L \in \{100, 300, 500, 1000, 3000\}$. RECURSIVE-$\phi$ recovers the full clean return by removing a single blackout for $L \leq 1000$ and two for $L = 3000$. The remaining injected sites (red ticks visible in every panel) are absorbed within a few paddle moves and never enter the causal support, even when they fall close in time to a gateway blackout (e.g. the cluster between steps 600–700 at $L = 1000$). The transfer gap is not always equal to K: a single gateway blackout can perturb the agent's policy state long enough to shift the rally cadence and shave more than one point off the score within the fixed L-step window, which is why the gap is $2, 1, 4, 3, 3$ while K stays at 1 or 2. This is the $K \ll |P|$ regime that the coarse-to-fine scheme is designed for.

Recursive is causally faithful and budget-stable at every L. At every length, the blue recursive bars saturate on exactly the green-marker step(s) and sit at machine zero everywhere else, with error bars too small to see on the spike heights. The reason is structural: once pruning collapses the inactive bulk of the horizon (one or two rounds suffice in this regime), the surviving sub-game has size $|\mathcal{A}|$ small enough that $2^{|\mathcal{A}|} \leq M = 100$ and the algorithm switches to exact enumeration on the remaining blocks. From that round onward the recursive readout is deterministic up to the on-track coupling, which is why the per-seed standard error on the recursive bars is zero — the panel headers confirm $10/10$ coalition seeds restore the clean return at every length. A second smaller spike (sub-causal but non-zero) appears at $L \geq 500$ on a non-injected step within the surviving sub-game; replaying with that step removed leaves the return at the corrupted value, so it is correctly ranked below the gateway step and excluded from the recovered top-K.

Flat at $M = 100$ is below threshold and degrades with L. At $L = 100$ and $L = 300$, the orange flat bars at the same $M = 100$ budget recursive uses are dominated by sampling variance: at $L = 100$ several non-causal positions are within a factor of two of the causal spike's height, and at $L = 300$ the causal site barely separates from the floor of distractors. The minimum recovering flat budget rises sharply with L: $[500, 1000]$ at $L = 100$ and $[1000, 5000]$ at $L = 300$ across the 10 coalition seeds, both well above the $M = 100$ shown. For $L \geq 500$ the flat sweep is skipped because no budget in our range $\leq 10\,000$ recovered on any seed in pilot runs, so only the recursive trace appears. This is the regime in which the scaling gap of the two estimators becomes operationally decisive: at $L = 1000$ recursive uses 148 rollouts to localise the single gateway blackout, while flat does not converge in $> 70\times$ that budget.

Coalition-seed variance. Recursive's terminal output is invariant to the coalition seed in this regime: every panel shows zero standard error on the spike heights because the surviving sub-game collapses to $|\mathcal{A}| \leq \log_2 M$ players within a couple of refinement rounds and the algorithm switches to exact enumeration. Flat's per-seed minimum recovering budget, by contrast, varies by up to $5\times$ at $L = 300$ (1000–5000 across the 10 seeds), reflecting how strongly the kernel-SHAP estimator depends on which coalitions happen to be drawn. Reporting the typical-seed budget rather than the luckiest is therefore essential to a fair comparison; using a single seed for flat would have given a $5\times$ band of headline numbers at $L = 300$ alone.

C.3 ProcGen CoinRun: detection traces across trajectory lengths

Table 2 reports the headline numbers for both CoinRun mechanism-error variants as a single (recovery, min-budget) pair per (variant, N). Figs. 9 and 10 unpack them into per-step $|\phi|$ traces, one figure per variant. Orange bars are flat $|\phi|/\max$ aggregated as mean $\pm$ 1 SE across the 10 coalition-sampling seeds, with each per-seed trace taken from the smallest budget at which top-K removal recovered the clean return (or, for seeds that never recovered within the 5k cap, from the 5k attempt). Blue bars are recursive $|\phi|/\max$ from the deterministic exact-enumeration run ($M = 100$ per round, $B = 2$). Red ticks mark the injected perturbation set.

Recursive concentrates on the gateway steps at every N. For Variant A (Fig. 9) the perturbation set is exact ($p_A = 1.0$, so every reward-eligible terminal flips), and recursive's blue mass sits exactly on the injected steps at all four lengths: a single bar at step 37 for $N \in \{100, 300\}$ and bars at $\{37, 364\}$ for $N \in \{500, 1000\}$. Recall is therefore 1.0 on Variant A across the sweep and the full clean-buggy gap is closed with ≤ 2 counterfactuals. Variant B (Fig. 10) is harder by design — the perturbation is stochastic ($p_B = 0.7$) and many short-jumps are absorbed by the policy. Recursive's blue mass concentrates on a strict subset of the realised perturbations: at $N = 1000$ only 2 of the 10 realised short-jumps carry non-zero recursive ϕ, yet repairing those two alone restores the entire return. This is the same mechanism-vs-causation gap that the deterministic CenterGoal pixel map in Fig. 11 cleanly isolates: ground truth is mechanistic, recursive picks the gateway steps whose consequences cascade.

Flat is dominated by sampling variance until N exceeds the budget cap. At $N = 100$, flat does eventually localise the causal step on every coalition seed (the orange peak coincides with the blue bar in Figs. 9a and 10a), but only after 620 ± 114 rollouts on Variant A and 810 ± 165 on Variant B — already $> 20\times$ recursive's deterministic cost of 28/40 rollouts. By $N = 300$ the per-seed minimum recovering budget for flat reaches 3800 ± 611 on Variant A and 1711 ± 653 on Variant B, the orange profile becomes visibly noisier (roughly uniform ~ 0.15 floor with the causal spike sitting only $\sim 5\times$ above it), and one of the ten seeds has already exhausted the 5k cap on Variant B. From $N = 500$ onwards the flat profile is essentially flat-noise on Variant A: max-normalised $|\phi|$ is uniform across the horizon and the seeds that did recover are dominated by lucky coalition draws rather than by a separated causal signal. Variant B at $N = 500, 1000$ shows the same pattern with $4/10$ and $2/10$ recovering, respectively, and the budget standard error is comparable to the mean — the few successful seeds are the upper tail of a heavy-tailed distribution, not a stable estimator of where the causal mass lies.

Coalition-seed variance and the failure mode. Recursive's per-seed standard error is zero on these traces because the surviving sub-game collapses to $|\mathcal{A}|$ small enough that $2^{|\mathcal{A}|} \leq M = 100$ and exact enumeration kicks in within one or two rounds; from there the readout is deterministic up to the on-track coupling. Flat has no such collapse: the kernel-SHAP estimator must allocate its budget over all T positions, and the variance grows with T. The most informative diagnostic is therefore not the average min-budget but the recovery rate — the fraction of the 10 coalition seeds that ever recovered within the cap. That rate degrades from $10/10$ at $N = 100$ to $2/10$ at $N = 1000$ on Variant B (and to $0/10$ on Variant A, $N = 500$). The Variant B, $N = 1000$ panel is the cleanest illustration: $K = 9$–10 realised short-jumps are scattered across the 1000-step trajectory, recursive concentrates blame on 2 of them and recovers the full gap in 76 rollouts, while flat's 5k cap recovers on only two seeds with a ± 2250-rollout standard error on those two — a budget on the same order as recursive's, but with a $50\times$ larger price tag and a 20% recovery rate.

C.4 MiniGrid `CenterGoal`

Fig. 6a in the main paper shows the per-step $|\phi|$ pixel map at $L = 100$, where every planted penalty step is visible at single-pixel resolution. Fig. 11 extends the same diagnostic to $n_{\text{cw}} = 10\,000$ (so $T \approx 10^4$ once the BFS-to-goal tail is appended), the longest length at which a single-pixel-per-step layout still fits on a page. The step index i is folded into a 100×100 grid (row $= i \,//\, 100$, column $= i \bmod 100$) so contiguous stretches of the rollout appear as horizontal bands; the planted penalty steps are at the $K = 5$ positions sampled randmly.

The penalty mechanism in this experiment is fully deterministic: `TimePenaltyWrapper` subtracts a fixed -0.1 at exactly the planted set, the policy is a hand-crafted clockwise-then-BFS-to-goal walk, and the environment is reset under a fixed seed. Mechanistic ground truth therefore equals causal

ground truth here: every planted step is causally responsible for the corresponding -0.1 in the return, and recursive's top-K should coincide exactly with the planted set, not merely overlap with it. Fig. 11 confirms this: the only bright pixels in the $10\,000$-step map are the five planted positions, and the rest of the grid is at machine zero. This is the cleanest demonstration of the localisation property at $L \approx 10^4$ the budget RECURSIVE-ϕ needs to recover the entire causal set is essentially the same as at $L = 100$ (Fig. 6c), because pruning collapses the $9\,995$ non-causal steps within the first few rounds and the surviving sub-game is small enough for exact enumeration. None of the flat baselines (FLAT-ϕ, SHAP, LIME) reaches this regime within the largest budget we sweep ($M = 10^5$); their pixel maps at the same length remain noisy across the full grid, which is why the gap-explained curves in Fig. 6b for the baselines flatten near 0 from $L \geq 2\,000$ onwards.

D Extended Related Work

Existing approaches to diagnosing world-model failures are largely indirect. Aggregate metrics measure reconstruction loss or reward prediction error across trajectories [18, 35]; short imagined rollouts from real data quantify how model errors compound over multi-step rollouts [22]. Saliency maps, structural causal models, and Shapley decompositions over state features explain the agent's policy at a given state but do not attribute blame to the learned dynamics over a rollout (see *Explaining RL agents* below). None of these lines answers the targeted question: at which time steps did the world model's prediction errors cause the policy to underperform? Most directly, Li et al. [24] introduce counterfactual Shapley values for trajectories with a flat coalition estimator; we extend their construction with a recursive coarse-to-fine algorithm that scales from $L \leq 300$ to $L \approx 10^6$. Given a fixed policy and a single observed failure trajectory, our method localizes which transitions in the world model are causally responsible for the return gap; we are unaware of prior work that addresses this question for a learned world model under a fixed policy.

Feature attribution. Shapley values [37] provide a principled way to distribute credit among players in a cooperative game. In machine learning, SHAP [25] instantiates this idea for individual predictions, treating input features as players and a trained model as the characteristic value function. LIME [33] fits a sparse local linear surrogate around each prediction rather than invoking the cooperative-game framework. Neither method was designed to attribute credit in multi-step rollouts, where an error at step t propagates through subsequent transitions. Our method treats trajectory time steps as players and uses counterfactual simulation as the characteristic value function, so attributions reflect causal responsibility for the return gap rather than correlational feature importance.

Explaining RL agents. A growing body of work seeks to explain the behavior of trained RL agents. Perturbation-based saliency maps highlight which observation regions drive an agent's action choice [13]. Approaches that learn a causal graph during RL use it to generate counterfactual explanations of agent actions [26]. Shapley-based methods apply the cooperative-game framework to attribute an agent's action choices, expected return, or predicted return to state features [5]. All three lines target the policy's action selection at a state and surface either correlations or local causal links at a given state, rather than attributing blame to the learned dynamics over a rollout. We instead attribute return loss to specific transitions of the world model, asking which transitions, not which features, caused the failure.

Model-based RL and world models. Model-based RL agents learn a world model to reduce sample complexity and enable planning [17, 18, 35, 39]. World models have achieved strong results across domains: Alonso et al. [2] demonstrate that a diffusion-based world model achieves state-of-the-art Atari 100k performance among agents trained entirely within a world model; for a broad survey of world models in robot learning, see Hou et al. [19]. A well-known failure mode is *model exploitation*: errors in the learned dynamics compound over multi-step rollouts and steer the policy toward spurious high-reward regions [22]. Recent work trains more robust world models by targeting data collection at environments where world model error is highest, guided by reward-free curricula [34]. Güzel et al. [14] train agents entirely inside a learned world model with an adaptive autocurriculum over imagined initial states, generalizing to novel settings without real-environment interaction. These lines all aim to build world models that work better on average; none localizes which transitions in a specific failure episode are causally responsible for the return gap. Most closely, Buesing et al. [8] counterfactually simulates alternative actions under shared random seeds to score policies inside a learned model. Prior work either mitigates model error uniformly, by limiting rollout length or reweighting model-based and model-free updates, or counterfactually scores policies rather than

transitions; neither line localizes which specific transitions caused the underperformance. Holding the policy fixed, we identify which specific transitions in the world model's rollout are causally responsible for the return gap, providing targeted feedback for model improvement.

Sim-to-real transfer. Sim-to-real transfer methods train policies in simulation and deploy them in the real world, relying on domain randomization over visual or dynamic parameters [27, 31, 41] to bridge the simulator-real gap. Teacher–student pipelines extend domain randomization by distilling privileged-state policies into visual ones under aggressive rendering randomization, enabling zero-shot transfer for humanoid loco-manipulation [46]. A related line uses structured benchmarks with explicit causal factors to study transfer under controlled interventions [1]. A separate line learns the simulator entirely: Ha and Schmidhuber [15] train policies inside a learned world model, Wu et al. [45] extend this to physical robots, and Bruce et al. [7], Yang et al. [47] scale learned interactive simulators to internet-scale video data. Domain randomization and learned simulators both target aggregate transfer: robustness over a distribution of environments, not diagnosis of any specific failure episode. Our method is complementary: given a specific failure episode, it pinpoints which simulator transitions caused the return gap, supporting targeted simulator improvement rather than aggregate robustness.

Safe and robust RL. Safe RL incorporates constraints alongside reward maximization [11]; Huang et al. [21] integrate Lagrangian-based safety constraints into a world model, either via per-step online planning (OSRP) or background policy optimization over imagined rollouts (BSRP). Robust RL trains policies that are tolerant to a family of model perturbations on transitions [29, 44] or on observations [48]. Both safe and robust RL are prescriptive and online: they aim to prevent failures during deployment by optimizing over constraints or perturbation families, whereas our method is a post-hoc diagnostic applied to a fixed policy after an observed failure. SafeDreamer specifically is distinguished from our work by its goal: it adapts the policy at deployment time to satisfy constraints, while we hold the policy fixed and attribute the return gap to specific simulator transitions. Our diagnostic is complementary: by identifying exactly which transitions drove a failure, it can directly inform what perturbation families robust RL should model, or which transition types require additional safety data.

E Theory Details

In this section, we provide detailed discussion on assumptions, definitions and proofs.

E.1 World Model Transfer Gap and Causal Attribution

Below is the formal version of the properties that a causal attribution for the world model transfer gap should satisfy (Def. 3.2).

Definition E.1 (Attribution for World Model Transfer Gap, Formal)**.** A world-model transfer gap attribution function ϕ maps a trajectory, a sequence of dynamics selectors, a world model, and a policy to a real vector denoting each step's attribution to the transfer gap,

$$\phi : \tau \times \{\mathrm{wm}, \mathrm{env}\}^T \times \mathcal{M}_{\mathrm{wm}} \times \Pi \times [T] \mapsto \mathbb{R}^T \tag{7}$$

where $\{\mathrm{wm}, \mathrm{env}\}^T$ is the sequence of dynamics selectors determining whether at time step t, $\{f_S, r\}$ is from $\mathcal{M}_{\mathrm{wm}}$ or $\mathcal{M}_{\mathrm{env}}$. Each desideratum below is a mapping $D_i : \Omega \times \mathbb{C} \times \Phi \to \{0, 1\}$, where Ω is the space of SCMs (each $\mathcal{M} \in \Omega$ packaging a world-model/environment pair $(\mathcal{M}_{\mathrm{wm}}, \mathcal{M}_{\mathrm{env}})$ with a trajectory τ and a policy π), $\mathbb{C}$ is the space of causal measures, and Φ is the space of attribution functions. The attribution function ϕ is causal if, for every step $t \in [T]$, $D_i(\mathcal{M}, c, \phi) = 1$ holds for $i = 1, \ldots, 4$:

$\mathbf{D}_1$ **Causal Admissibility**: Steps that do not cause the performance gap have zero credit:

$$c = 0 \implies \phi = \mathbf{0}. \tag{8}$$

$\mathbf{D}_2$ **Causal Power**: Steps that cause the performance gap are assigned non-zero credit:

$$\exists\, \mathbf{u}, z : \; c \neq 0 \implies \phi \neq \mathbf{0}. \tag{9}$$

$\mathbf{D}_3$ **Causal Normality**: Steps that are more likely to affect the outcome have more credit:

$$c_{\mathcal{M}} = c_{\mathcal{M}'} \;\wedge\; P^{\mathcal{M}}_{\bar{c}}(Z) \not\equiv P^{\mathcal{M}'}_{\bar{c}}(Z) \;\wedge\; P^{\mathcal{M}}_{+}(z) \geq P^{\mathcal{M}'}_{+}(z) \implies \phi_{\mathcal{M}} > \phi_{\mathcal{M}'}. \tag{10}$$

$\mathbf{D}_4$ **Causal Effect Scaling**: Steps affecting the outcome more are assigned more credit:

$$c_{\mathcal{M}} \geq c_{\mathcal{M}'} \ \wedge \ P^{\mathcal{M}}(Z) \equiv P^{\mathcal{M}'}(Z) \ \wedge \ P^{\mathcal{M}}(c_Z) \not\equiv P^{\mathcal{M}'}(c_Z) \implies \phi_{\mathcal{M}} > \phi_{\mathcal{M}'}. \tag{11}$$

Quantification: all premises universally quantified unless marked $\exists$. Notation: c, $c_{\mathcal{M}}$ abbreviate $c(\mathcal{M}, \mathbf{u}, X, Y, z)$ — the local discrepancy between $\mathcal{M}_{\text{wm}}$ and $\mathcal{M}_{\text{env}}$ at step t as seen through its downstream effect on the return Y; ϕ, $\phi_{\mathcal{M}}$ abbreviate $\phi_t(\mathcal{M})$, the per-step attribution at t; $P \equiv P'$ denotes distributional equality; $P^{\mathcal{M}}_{\not c}(Z) \coloneqq P^{\mathcal{M}}(Z \mid c \neq 0)$; $P^{\mathcal{M}}_{+}(z) \coloneqq \text{sign}(c_{\mathcal{M}}) \cdot P^{\mathcal{M}}(z)$.

Proof for Thm. 3.3. We show that the ϕ-values under our transfer gap attribution setting with NTE $f(\mathbf{z}) = Y^{\tau}_{\mathcal{M}_{\text{env}}} - Y_{\sigma_{\mathbf{z}}}$ from Eqs. (5) and (6) satisfies each of $\mathbf{D_1}$–$\mathbf{D_4}$ in Def. E.1.

Setting. The "players" are the per-step dynamics selectors: a coalition $\mathbf{z} \in \{0,1\}^T$ specifies, for each step t, whether the transition (x_t, r_t, s_{t+1}) in the counterfactual rollout is drawn from the world model $\mathcal{M}_{\text{wm}}$ (when $z_t = 1$) or copied from the observed real-environment trajectory (when $z_t = 0$). The "causal measure" $c \coloneqq c(\mathcal{M}, \mathbf{u}, X_t, Y, z)$ is the functional dependence of the return Y on this selector at step t — equivalently, the local discrepancy between $\mathcal{M}_{\text{wm}}$ and $\mathcal{M}_{\text{env}}$ at t as seen through its downstream impact on the return. The "SCM" $\mathcal{M}$ here packages the pair $(\mathcal{M}_{\text{wm}}, \mathcal{M}_{\text{env}})$ together with the trajectory τ, the policy π, and the exogenous draws $(\mathbf{u}, \mathbf{u}')$ in Eq. (5); the "baseline distribution" $P^{\mathcal{M}}(Z)$ is the corresponding distribution over counterfactual contexts (i.e., the off-track rollouts induced by $\mathbf{u}'$ and π under $\mathcal{M}_{\text{wm}}$).

D1 (Causal Admissibility). Suppose $c = 0$: at step t, the world model is functionally indistinguishable from the real environment for Y across every context $(\mathbf{u}, z)$. Then flipping z_t between 0 and 1 leaves the counterfactual return unchanged, $Y_{\sigma_{\mathbf{z}\cup\{t\}}} = Y_{\sigma_{\mathbf{z}}}$ for every coalition $\mathbf{z}$, so every marginal contribution vanishes: $f(\mathbf{z}\cup\{t\}) - f(\mathbf{z}) = Y_{\sigma_{\mathbf{z}}} - Y_{\sigma_{\mathbf{z}\cup\{t\}}} = 0$. By the Shapley null-player axiom, $\phi_t = 0$, so a step at which the world model agrees with the real environment receives no credit for the transfer gap.

D2 (Causal Power). Suppose there exist exogenous draws $\mathbf{u}$ and a context z such that $c(\mathcal{M}, \mathbf{u}, X_t, Y, z) \neq 0$, i.e., for some configuration of the other steps, swapping $\mathcal{M}_{\text{env}}$ for $\mathcal{M}_{\text{wm}}$ at step t shifts the return. Then there is a coalition $\mathbf{z}^*$ (matching that context) with nonzero marginal contribution $f(\mathbf{z}^* \cup \{t\}) - f(\mathbf{z}^*) \neq 0$. The Shapley weights $w(\mathbf{z}) = \frac{|\mathbf{z}|!(T-|\mathbf{z}|-1)!}{T!}$ are strictly positive for every $\mathbf{z}$, so

$$\phi_t = \sum_{\mathbf{z}: z_t = 0} w(\mathbf{z}) \big[f(\mathbf{z}\cup\{t\}) - f(\mathbf{z}) \big]$$

contains a strictly weighted nonzero term. Under Assumption E.5 (causal steps share a sign, so marginal contributions cannot systematically cancel across coalitions), this entails $\phi_t \neq 0$: any step at which the world model can move the return relative to the real world is given nonzero credit.

D3 (Causal Normality). Consider two world-model/environment pairs $\mathcal{M}, \mathcal{M}'$ with identical local discrepancies at step t ($c_{\mathcal{M}} = c_{\mathcal{M}'}$) but different baseline distributions over counterfactual contexts, with $P^{\mathcal{M}}_{+}(z) \geq P^{\mathcal{M}'}_{+}(z)$ (the baseline under $\mathcal{M}$ assigns weakly higher probability to contexts in which the local discrepancy at t contributes a same-signed effect on the gap). The NTE value $f(\mathbf{z})$ depends on $\mathcal{M}$ through the counterfactual rollout: off-track exogenous draws $\mathbf{u}'$ are propagated through $\mathcal{M}_{\text{wm}}$ at intervened steps and through the observed trajectory at non-intervened ones. Under $\mathcal{M}$, the counterfactual rollout more frequently visits contexts in which the discrepancy at step t produces a positive contribution to the transfer gap, so each marginal contribution $f(\mathbf{z}\cup\{t\}) - f(\mathbf{z})$ weakly exceeds its counterpart under $\mathcal{M}'$. The Shapley value is the expected marginal contribution over uniformly random orderings; since each ordering's marginal contribution under $\mathcal{M}$ dominates that under $\mathcal{M}'$ (by monotonicity of the NTE in baseline probabilities when the causal sign is fixed), we obtain $\phi_t^{\mathcal{M}} > \phi_t^{\mathcal{M}'}$. In words: when the surrounding rollout more often exposes step t's discrepancy, step t earns proportionally more credit for the transfer gap.

D4 (Causal Effect Scaling). Now consider two pairs $\mathcal{M}, \mathcal{M}'$ with identical baseline distributions over counterfactual contexts ($P^{\mathcal{M}}(Z) \equiv P^{\mathcal{M}'}(Z)$) but a strictly larger local discrepancy at step t in $\mathcal{M}$, i.e., $c_{\mathcal{M}} \geq c_{\mathcal{M}'}$. A larger world-model/real-environment discrepancy at t directly inflates the magnitude of every counterfactual return difference: for each coalition $\mathbf{z}$ with $z_t = 0$,

$$\big| Y_{\sigma_{\mathbf{z}}} - Y_{\sigma_{\mathbf{z}\cup\{t\}}} \big|_{\mathcal{M}} \ \geq \ \big| Y_{\sigma_{\mathbf{z}}} - Y_{\sigma_{\mathbf{z}\cup\{t\}}} \big|_{\mathcal{M}'}.$$

Because the baseline distribution — and hence the Shapley weights — are identical across the two games, the Shapley monotonicity property (if player t's marginal contribution in game v weakly exceeds that in game v' for every coalition, then $\phi_t(v) \geq \phi_t(v')$) yields $\phi_t^{\mathcal{M}} > \phi_t^{\mathcal{M}'}$. That is, a step at which the world model deviates more from the real environment is awarded strictly more credit for the transfer gap, as desired. □

The intervention design for the counterfactual simulation rests on the following assumption.

Assumption E.2 (Counterfactual Independence)**.** Exogenous noise is independent across steps: $(S_{t+1})_{S_t, X_t} \perp (S_{t+1})_{S_{t'}, X_{t'}}$ for $(S_t, X_t) \neq (S_{t'}, X_{t'})$, and $(X_t)_{S_t} \perp (X_t)_{S_{t'}}$ for $S_t \neq S_{t'}$.

This assumption does not constrain the MDP, only its SCM representation. Since any distribution can be sampled via inverse CDF with independent uniform noise, any MDP admits an SCM with independent exogenous variables and the same optimal policy; the assumption selects this representation. The consequence is simple: *when parents match the observed trajectory, the observed value is the counterfactual; when parents differ, we resample*. For actions (parent S_t): when $z_t = 0$ and $s_t^{\mathbf{z}} = s_t$, the counterfactual equals the observed x_t; when $s_t^{\mathbf{z}} \neq s_t$, we resample from $\pi(\cdot \mid s_t^{\mathbf{z}})$. When $z_t = 1$, we resample from π_{base}. For transitions and rewards (parents (S_t, X_t)): when $(s_t^{\mathbf{z}}, x_t^{\mathbf{z}}) = (s_t, x_t)$, we reuse (s_{t+1}, y_t); otherwise, we resample. The full pseudo-code is provided in Algo. 2 in Sec. F.

E.2 Correctness of Coarse-to-Fine Estimation

We now characterize the conditions under which RECURSIVESHAPLEY correctly identifies all causally important time steps. The key requirement is that causal responsibility is *sparse* and that the signal from causal steps is not diluted below the pruning threshold when aggregated into blocks.

Definition E.3 ((K, δ)-Sparse Attribution)**.** A Shapley value vector $\phi_{1:T}$ is (K, δ)-sparse if at most K indices have $|\phi_t| \geq \delta$ and the remaining indices satisfy $|\phi_t| = 0$.

When time steps are grouped into blocks and treated as composite players, the block-level Shapley value Φ_j in the reduced game need not equal $\sum_{t \in \mathcal{B}_j} \phi_t$ exactly. The discrepancy is captured by the *Shapley interaction index* [12, 28], which generalizes the Shapley value to subsets of players: for $S \subseteq [T]$, the interaction index $I(S)$ measures the $|S|$-order synergy among players in S beyond what their individual Shapley values explain (for $|S| = 1$ it reduces to ϕ_t; for $|S| = 2$ it quantifies pairwise synergy or redundancy). Collapsing a block $\mathcal{B}_j$ into a composite player is algebraically equivalent to aggregating every interaction index supported within $\mathcal{B}_j$, giving the decomposition $\Phi_j = \sum_{t \in \mathcal{B}_j} \phi_t + \sum_{S \subseteq \mathcal{B}_j, |S| \geq 2} c_S \cdot I(S)$, where c_S are combinatorial weights from the interaction decomposition. The higher-order sum vanishes when transitions in $\mathcal{B}_j$ act independently and grows when they interact (e.g., two world-model errors that jointly, but not individually, derail the rollout). The following assumption controls the magnitude of this correction.

Assumption E.4 (Bounded Interaction Effect)**.** For any partition of $\{1, \ldots, T\}$ into contiguous blocks $\mathcal{B}_1, \ldots, \mathcal{B}_N$, let Φ_j denote the Shapley value of block j in the N-player reduced game. There exists $\eta \geq 0$ such that for all blocks j: $\left|\Phi_j - \sum_{t \in \mathcal{B}_j} \phi_t\right| = \left|\sum_{S \subseteq \mathcal{B}_j, |S| \geq 2} c_S \cdot I(S)\right| \leq \eta$.

Although stated as an assumption, the bound η is not restrictive in any absolute sense: each interaction index $I(S)$ is a signed combination of $2^{|S|}$ evaluations of $f(\mathbf{z}) = Y - Y^{\mathbf{z}}$ over coalitions that differ only on S, so every $I(S)$, and the finite combination $\sum_{|S| \geq 2} c_S \cdot I(S)$ that defines the block correction, is automatically bounded by $O(Y_{\max})$ whenever returns are bounded, which holds in any finite-horizon MDP with bounded rewards. Practically, we would like η be small *relative to* the causal gap δ, so that grouping does not blur the distinction between causal and non-causal blocks.

Three structural features of the world-model debugging setting drive η well below δ in practice. First, *sparsity collapses most higher-order terms*: the Shapley interaction index $I(S)$ [43] for $|S| \geq 2$ is non-negligible only when every player in S has a meaningful marginal effect, and under (K, δ)-sparsity the vast majority of subsets $S \subseteq \mathcal{B}_j$ contain only non-causal steps and therefore contribute nothing. Second, the *on-track coupling* in CTFSIM reuses observed transitions for non-causal steps, so intervening on a non-causal step leaves the counterfactual rollout unchanged; its marginal contribution is zero in every coalition, which forces every $I(S)$ containing it to vanish exactly and eliminates all interaction mediated by non-causal players. Third, when two causal steps do fall in the same block, their effects typically propagate through disjoint segments of the rollout separated by stretches of

recoverable dynamics, so the marginal effect of each is approximately independent of whether the other is intervened on and the pairwise interaction $I(\{t, t'\})$ is small. Together these give $\eta = 0$ exactly when all within-block interaction indices vanish, and $\eta \ll \delta$ generically in the sparse-causality regime.

However, a small η alone is not sufficient: even when $\Phi_j \approx \sum_{t \in \mathcal{B}_j} \phi_t$, a block containing two causal steps with opposite-sign attributions (e.g., one world-model error that hurts the agent and another that accidentally helps it) could have $|\Phi_j| \approx 0$ despite containing causally important steps, causing false pruning. We therefore additionally require that causal attributions do not cancel within blocks.

Assumption E.5 (Sign Consistency)**.** All causal time steps (those with $|\phi_t| \geq \delta$) have the same sign: either $\phi_t \geq \delta$, or $\phi_t \leq -\delta$ for all such t.

Sign consistency ensures that when multiple causal steps fall in the same block, their contributions *reinforce* rather than cancel: $|\sum_{t \in \mathcal{B}_j} \phi_t| \geq \max_{t \in \mathcal{B}_j} |\phi_t|$. This assumption is natural in the world-model debugging setting because the errors we aim to surface are *systematic* mispredictions, not statistical noise or small perturbations from which the policy quickly recovers. Systematic transition errors tend to *compound* [22] along a rollout: once the world model pushes the agent off the ground-truth manifold, subsequent mispredicted transitions typically accumulate the performance deficit rather than offset it, so the ϕ_t of the responsible steps share a sign. The threshold δ in the (K, δ)-sparsity definition deliberately filters out the opposite regime: small, recoverable, or noise-like perturbations have $|\phi_t| < \delta$ and never enter the causal set, even if they happen to have opposite signs.

Theorem E.6 (Correctness of RECURSIVESHAPLEY)**.** *Let $\phi_{1:T}$ be (K, δ)-sparse (Def. E.3) and suppose Assumption E.4 holds with parameter η and Assumption E.5 holds. At each round r, let $\hat{\Phi}_j$ denote the estimated block-level Shapley value and suppose $|\hat{\Phi}_j - \Phi_j| \leq \alpha$ with probability at least $1 - \beta$ (over coalition sampling, achievable by exact calculation or optimal proposal distribution Q_n^* from Li et al. [24]). If the pruning threshold satisfies*

$$\alpha + \eta < \epsilon < \frac{(\delta - \alpha - \eta) \cdot B}{T}, \tag{12}$$

where $r_0 = \min\{r : B^{r+1} \geq T/B\}$ is the first round at which blocks have size $\leq B$, and $\delta > \alpha + \eta$, then with probability at least $(1 - \beta)^R$ where $R = \lceil \log_B T \rceil$:

(a) ***No false pruning.*** *No block containing a causal time step (with $|\phi_t| \geq \delta$) is pruned at any round.*

(b) ***Localization.*** *After R rounds, the active set contains at most K time steps, all of which are causal.*

(c) ***Approximation error.*** *The final estimates satisfy $|\hat{\phi}_t - \phi_t| \leq \alpha + \eta$ for causal steps and $\hat{\phi}_t = 0$ for pruned (non-causal) steps.*

Proof. Consider a block $\mathcal{B}_j$ at round r containing a causal step t^* with $|\phi_{t^*}| \geq \delta$. If $\mathcal{B}_j$ contains additional causal steps, sign consistency (Assumption E.5) guarantees they share the same sign as ϕ_{t^*}, so $|\sum_{t \in \mathcal{B}_j} \phi_t| \geq |\phi_{t^*}| \geq \delta$ (causal contributions reinforce, they do not cancel). By Assumption E.4 and the reverse triangle inequality, $|\Phi_j| \geq |\sum_{t \in \mathcal{B}_j} \phi_t| - \eta \geq \delta - \eta$, and by estimation accuracy, $|\hat{\Phi}_j| \geq \delta - \eta - \alpha$ with probability $\geq 1 - \beta$. The per-step pruning criterion checks $|\hat{\Phi}_j|/|\mathcal{B}_j| \geq \epsilon$. At round r, block sizes are $|\mathcal{B}_j| = T/B^{r+1}$ (up to rounding), so the per-step signal is $(\delta - \eta - \alpha) \cdot B^{r+1}/T$. At the beginning ($r = 0$), blocks are large (size $> B$) and the per step signal is diluted. Thus, the biggest threshold we can set should be smaller than this diluted per-step value to avoid premature pruning. Setting $r = 0$ and we arrive at Condition (12).

For part (b), blocks containing only non-causal steps have $|\sum_{t \in \mathcal{B}_j} \phi_t| = 0$, so $|\Phi_j| \leq \eta$ and $|\hat{\Phi}_j| \leq \eta + \alpha < \epsilon \cdot |\mathcal{B}_j|$ by the gap condition. These blocks are pruned, and only blocks overlapping with causal steps remain. After $R = \lceil \log_B T \rceil$ rounds, each remaining block has size 1, yielding at most K active steps.

For part (c), at the final round the per-step estimate is $\hat{\phi}_t = \hat{\Phi}_j$ for singleton blocks, so $|\hat{\phi}_t - \phi_t| \leq |\hat{\Phi}_j - \Phi_j| + |\Phi_j - \phi_t| \leq \alpha + \eta$. A union bound over R rounds gives the overall success probability $(1 - \beta)^R$. □

Remark E.7 (Relaxing sign consistency). When Assumption E.5 is violated where some world-model errors help the agent while others hurt it, the algorithm remains sound but may require a more conservative (smaller) pruning threshold or additional rounds. One practical relaxation is to run the algorithm twice, once with the original value function $f(\mathbf{z}) = Y - Y^{\mathbf{z}}$ and once with $f^+(\mathbf{z}) = |Y - Y^{\mathbf{z}}|$, taking the union of active sets from both runs. The absolute-value variant detects blocks with large causal activity regardless of sign, at the cost of losing the signed direction of attribution.

Remark E.8 (Computational cost and approximation error). Since pruning reduces the active set to $O(K)$ blocks within the first few rounds, the effective per-round cost is $O(2^K)$ simulations for exact enumeration (rather than $O(2^T)$ for the flat estimator), giving a total cost of $O(2^K \cdot \log_B T)$ complexity. In practice, when $2^K \leq M$, the algorithm switches to exact enumeration in later rounds, eliminating estimation error ($\alpha = 0$) for the final sub-game and tightening the bound in (c) to $|\hat{\phi}_t - \phi_t| \leq \eta$.

F Algorithm Details

In this section, we provide detailed pseudocode for major algorithms discussed in the main text.

Algorithm 2 CTFSIM: Counterfactual Trajectory Simulation

Input: Coalition $\mathbf{z} \in \{0,1\}^T$, trajectory $\tau = (s_{1:T}, x_{1:T}, y_{1:T})$, policies π, π_{base}
Output: $(s^{\mathbf{z}}_{1:T}, y^{\mathbf{z}}_{1:T}, Y^{\mathbf{z}})$: counterfactual states, rewards, and total return

$s^{\mathbf{z}}_1 \leftarrow s_1$
for $t = 1$ to T **do**
  if $z_t = 0$ **and** $s^{\mathbf{z}}_t = s_t$ **then**
    $x^{\mathbf{z}}_t \leftarrow x_t$ {On-track: reuse observed}
  else if $z_t = 0$ **then**
    $x^{\mathbf{z}}_t \sim \pi(\cdot \mid s^{\mathbf{z}}_t)$ {Off-track: sample from π}
  else
    $x^{\mathbf{z}}_t \sim \pi_{\text{base}}(\cdot \mid s^{\mathbf{z}}_t)$ {Intervention}
  end if
  if $(s^{\mathbf{z}}_t, x^{\mathbf{z}}_t) = (s_t, x_t)$ **then**
    $(s^{\mathbf{z}}_{t+1}, y^{\mathbf{z}}_t) \leftarrow (s_{t+1}, y_t)$ {Reuse observed}
  else
    $s^{\mathbf{z}}_{t+1} \sim P(\cdot \mid s^{\mathbf{z}}_t, x^{\mathbf{z}}_t)$
    $y^{\mathbf{z}}_t \sim r(s^{\mathbf{z}}_t, x^{\mathbf{z}}_t, \cdot)$
  end if
end for
$Y^{\mathbf{z}} \leftarrow \sum_{t=1}^{T} \gamma^{t-1} y^{\mathbf{z}}_t$ {Discounted return}

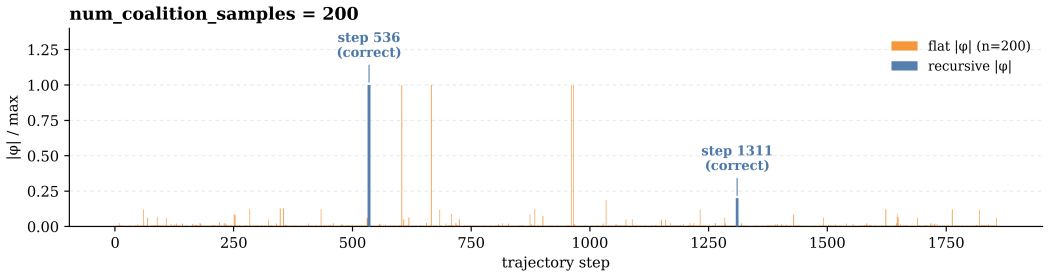
return $(s^{\mathbf{z}}_{1:T}, y^{\mathbf{z}}_{1:T}, Y^{\mathbf{z}})$

Algorithm 3 FLAT-ϕ: Single-Round Counterfactual Shapley Estimation

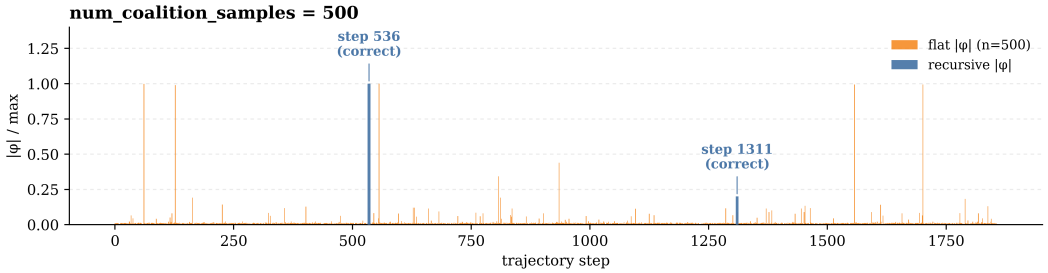
Input: Trajectory τ of length T; policy π; baseline π_{base}; environment P; coalition budget M
Output: Per-step attributions $\hat{\phi}_{1:T}$

$\hat{\phi}_{1:T} \leftarrow \mathbf{0}$
$Y \leftarrow \sum_{t=1}^{T} \gamma^{t-1} y_t$ {Observed return}
Sample M coalitions $\mathbf{z}^{(m)} \sim Q_T^*$ for T players
for $m = 1$ to M **do**
  Simulate counterfactual $\tau^{\mathbf{z}^{(m)}}$ via CTFSIM($\mathbf{z}^{(m)}, \tau, \pi, \pi_{\text{base}}$)
  $Y^{\mathbf{z}^{(m)}} \leftarrow \sum_{t=1}^{T} \gamma^{t-1} y_t^{\mathbf{z}^{(m)}}$; $\delta \leftarrow Y - Y^{\mathbf{z}^{(m)}}$
  Compute importance-weighted kernel weights $\kappa_t(\mathbf{z}^{(m)})$ using Q_T^*
  for $t = 1$ to T **do**
    $\hat{\phi}_t \mathrel{+}= \delta \cdot \kappa_t(\mathbf{z}^{(m)}) \;/\; M$
  end for
end for

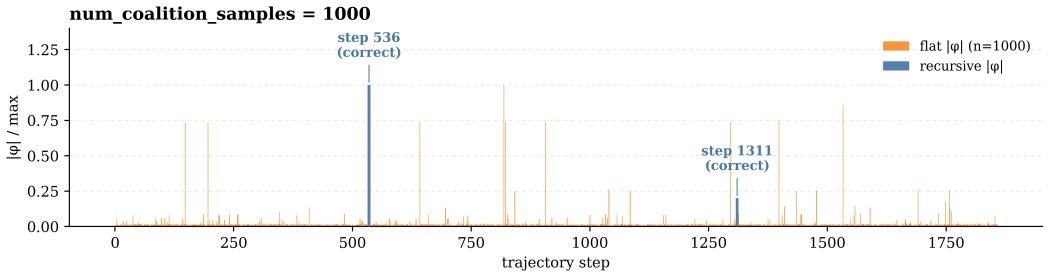
return $\hat{\phi}_{1:T}$



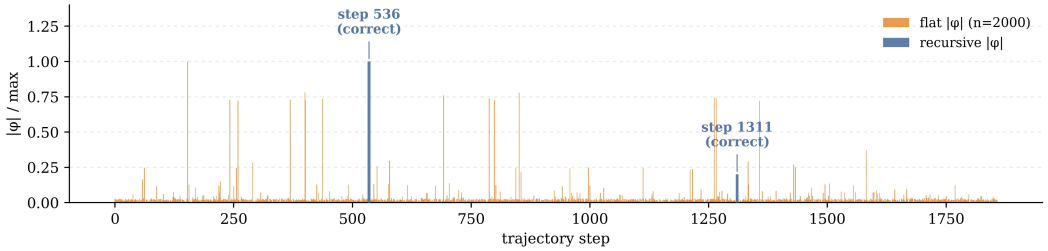
(a) $M = 200$.



(b) $M = 500$.

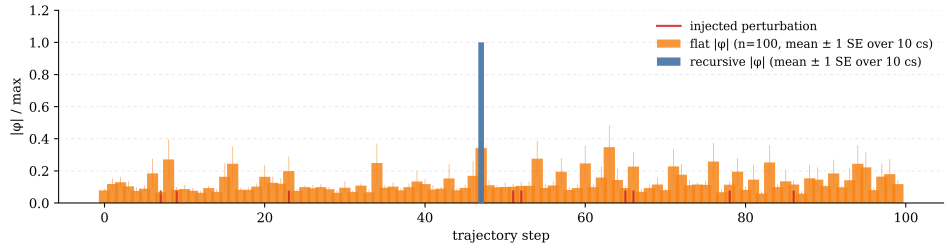


(c) $M = 1000$.

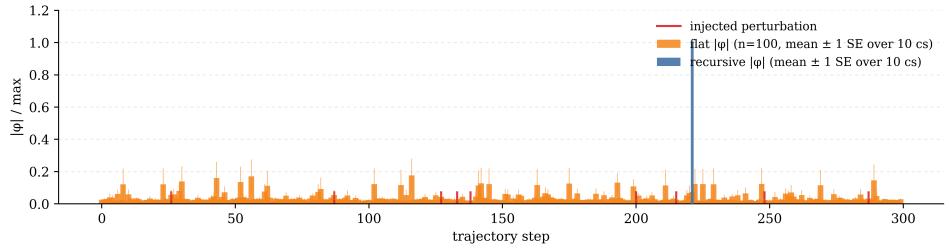


(d) $M = 2000$.

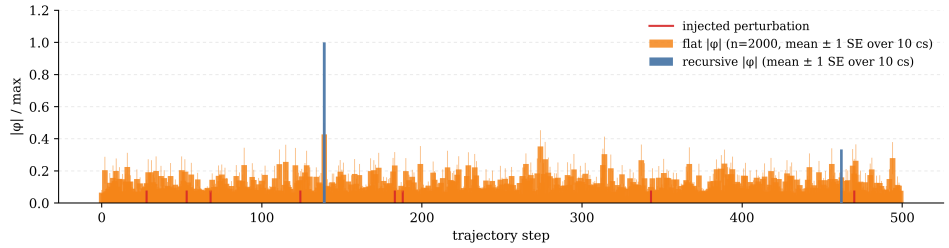
Figure 7: Per-step $|\phi|$ traces (normalised by the per-method max) on the same ~ 1860-step Pong trajectory across coalition budgets $M \in \{200, 500, 1000, 2000\}$. Recursive (blue) places its two largest spikes on the ground-truth causal blackouts at steps 536 and 1311 at every budget; flat (orange) is dominated by sampling variance and at no budget in this range does its top-2 recover both causal sites.



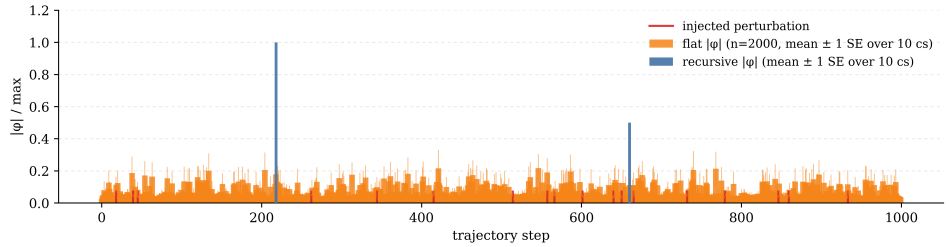
(a) $L = 100$, $|P| = 10$, $K = 1$, gap $= 2$. Min flat budget $\in [500, 1000]$ over 10 cs (10/10 recovered).



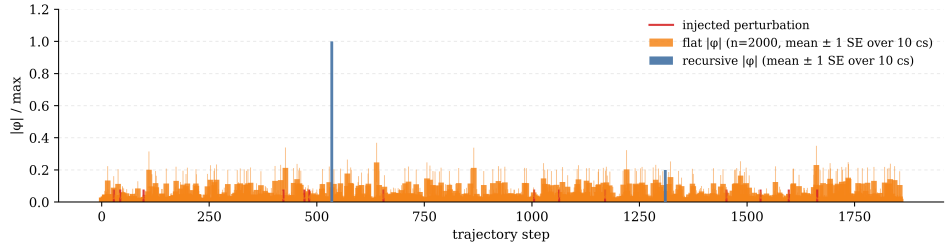
(b) $L = 300$, $|P| = 10$, $K = 1$, gap $= 1$. Min flat budget $\in [1000, 5000]$ over 10 cs (10/10 recovered).



(c) $L = 500$, $|P| = 10$, $K = 1$, gap $= 4$. Flat sweep skipped ($L \geq 500$).

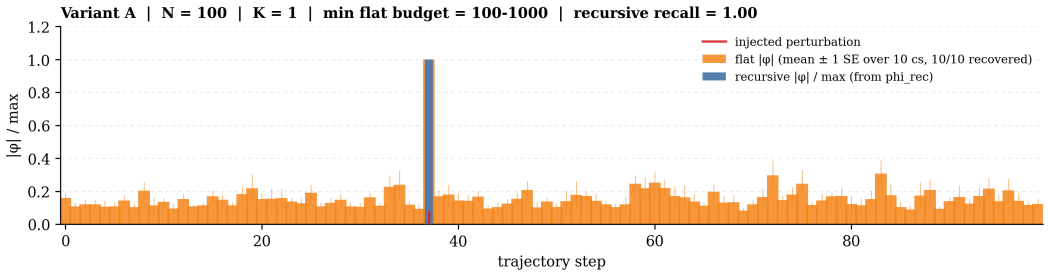


(d) $L = 1000$, $|P| = 20$, $K = 1$, gap $= 3$. Flat sweep skipped.

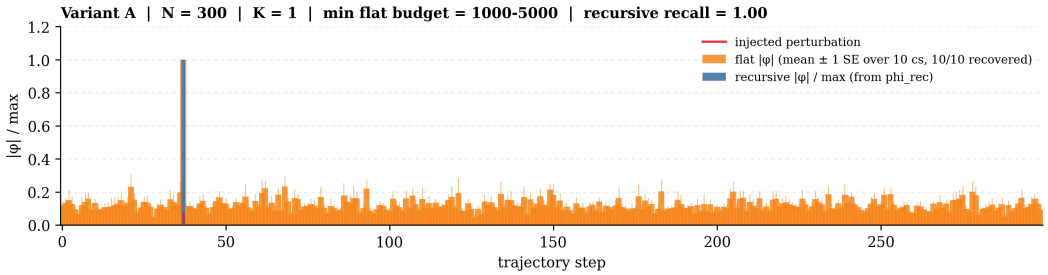


(e) $L = 3000$, $|P| = 20$, $K = 2$, gap $= 3$. Flat sweep skipped.

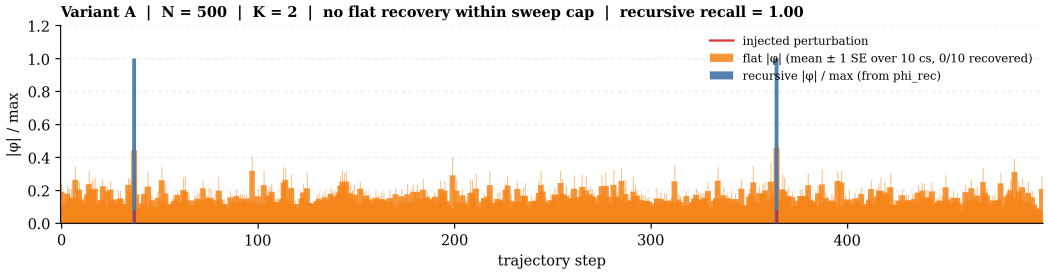
Figure 8: Per-step $|\phi|$ traces across L. Blue: RECURSIVE-ϕ. Orange: FLAT-ϕ at $M = 100$, the same per-round budget recursive uses. Red ticks: injected blackouts; green dashed markers: causal top-K from recursive (recovers the clean return on $10/10$ seeds at every length).



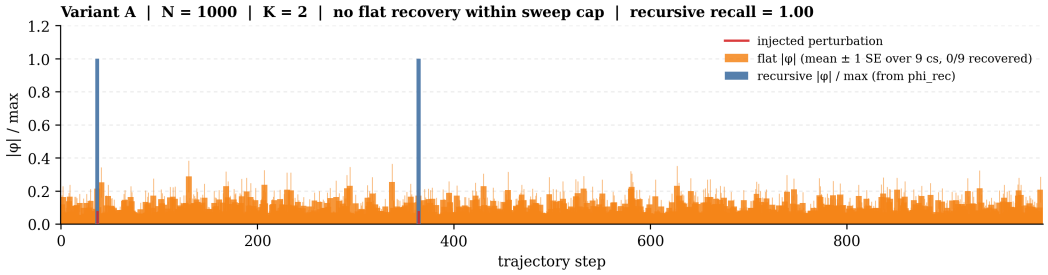
(a) $N = 100$. $K = 1$, min flat budget $\in [100, 1000]$ over 10 cs ($10/10$ recovered).



(b) $N = 300$. $K = 1$, min flat budget $\in [1000, 5000]$ ($10/10$ recovered).

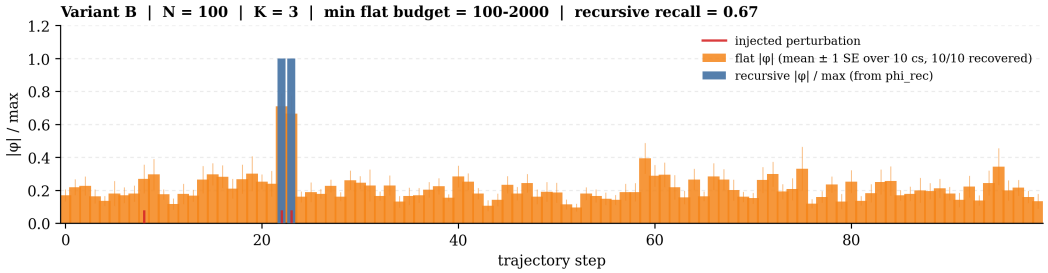


(c) $N = 500$. $K = 2$. No flat recovery within 5k cap on any of the 10 cs.

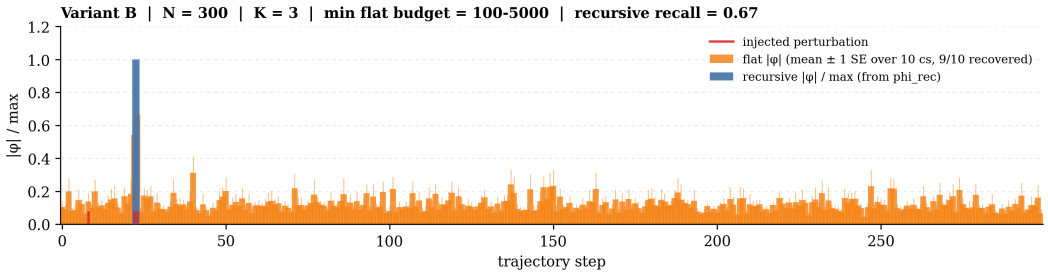


(d) $N = 1000$. $K = 2$. No flat recovery within 5k cap on $9/9$ available cs.

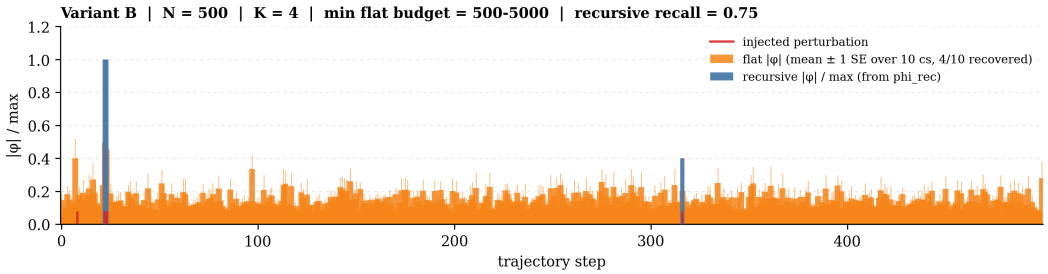
Figure 9: **Variant A (poisonous reward): per-step $|\phi|$ traces across N.** Orange: flat $|\phi|/\max$, mean $\pm$ 1 SE across 10 coalition-sampling seeds at each seed's recovering budget (or 5k if no budget recovered). Blue: recursive $|\phi|/\max$ from the deterministic enumerate-on-survive run. Red ticks at the bottom mark the injected perturbation steps. Recursive recovers 100% of the transfer gap on every panel.



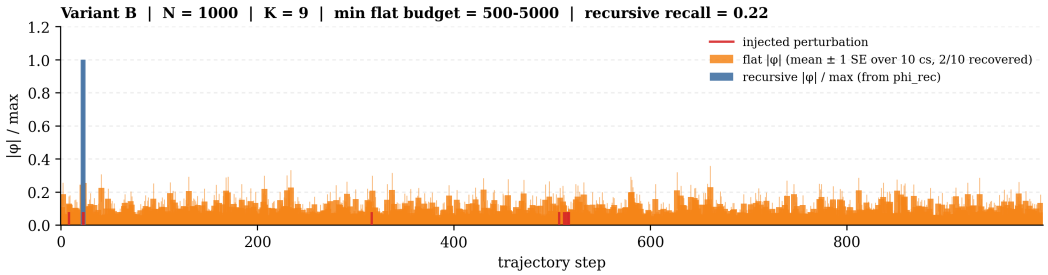
(a) $N = 100$. $K = 3$, min flat budget $\in [100, 2000]$ (10/10 recovered).



(b) $N = 300$. $K = 3$, min flat budget $\in [100, 5000]$ (9/10 recovered).



(c) $N = 500$. $K = 4$, min flat budget $\in [500, 5000]$ (4/10 recovered).



(d) $N = 1000$. $K = 9$ (recursive realisation) / 10 (flat realisation), min flat budget $\in \{500, 5000\}$ (2/10 recovered).

Figure 10: **Variant B (windy day): per-step $|\phi|$ traces across N.** Same conventions as Fig. 9. Recursive recovers 100% of the transfer gap on every panel; the gap between recursive's top-K and the (mechanistic) injected set widens with N: most realised short-jumps are absorbed by the policy and only the gateway events carry non-zero recursive ϕ.

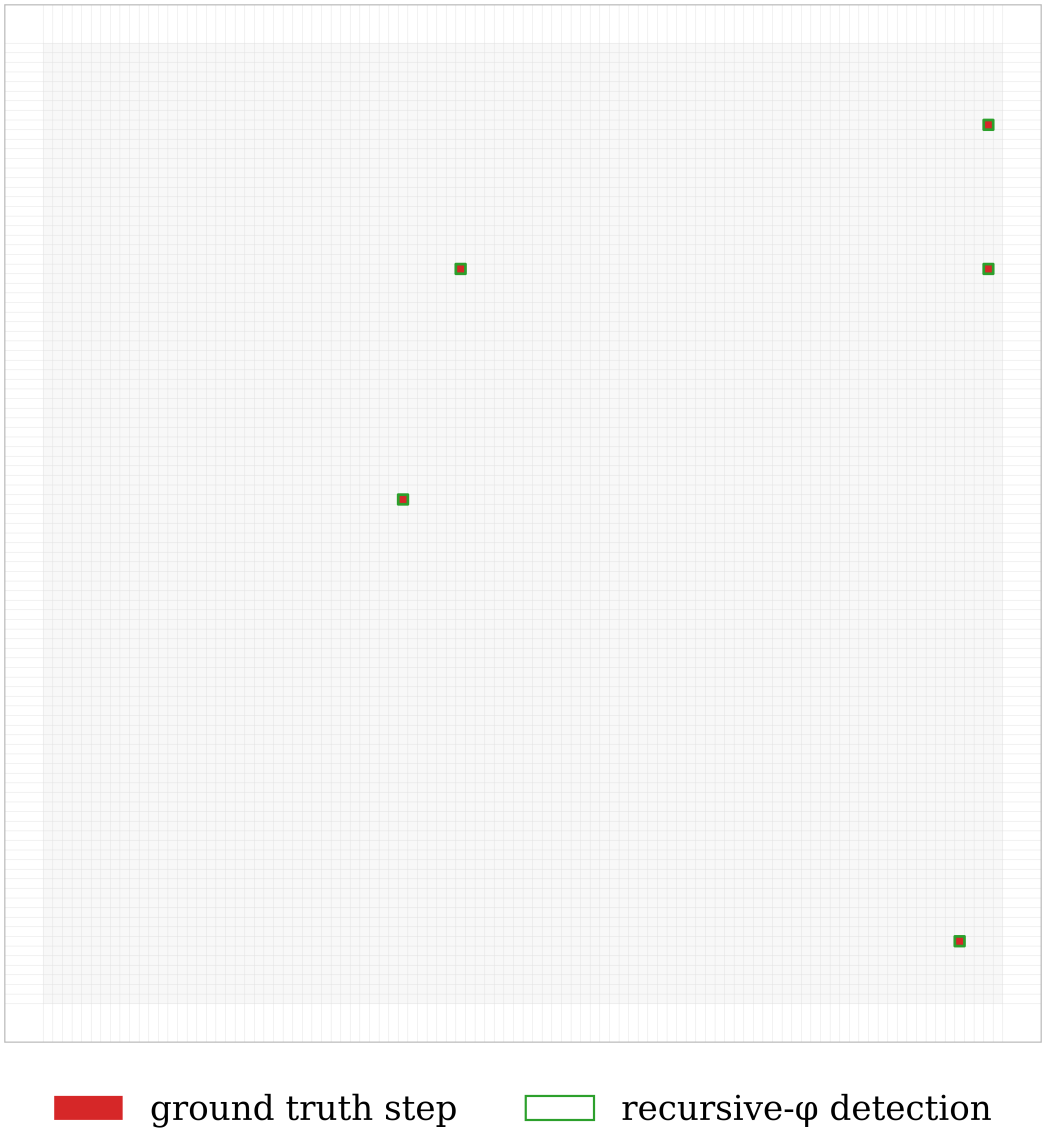


Figure 11: **Per-step $|\phi|$ pixel map on `MiniGrid-CenterGoal5x5-v0` at $n_{\mathrm{cw}} = 10\,000$ ($K = 5$).** Step index i is folded into a 100×100 grid (row $= i \,//\, 100$, column $= i \bmod 100$); a brighter pixel marks a larger $|\phi_t|$. RECURSIVE-ϕ places non-zero mass on exactly the five planted penalty steps and zero everywhere else, so the recovered top-K matches the planted set on $10/10$ coalition seeds whereas other baselines fail across all seeds at this length.

Algorithm 4 RECURSIVE-ϕ: Coarse-to-Fine Counterfactual Shapley Estimation

Input: Trajectory τ of length T; policy π; baseline π_{base}; environment P; coalition budget M; split factor B; pruning threshold ϵ

Output: Per-step attributions $\hat{\phi}_{1:T}$

$\hat{\phi}_{1:T} \leftarrow \mathbf{0}$; $\quad \texttt{active}_{1:T} \leftarrow \mathbf{1}$ {All steps active initially}
$Y \leftarrow \sum_{t=1}^{T} \gamma^{t-1} y_t$ {Observed return}
$R \leftarrow \lceil \log_B T \rceil$ {Maximum number of rounds}
for $r = 0$ to $R - 1$ **do**
  $N_r \leftarrow \min(B^{r+1},\ T)$ {Number of blocks this round}
  Partition $\{1, \dots, T\}$ into N_r contiguous blocks $\mathcal{B}_1, \dots, \mathcal{B}_{N_r}$
  $\mathcal{A} \leftarrow \{j : \exists\, t \in \mathcal{B}_j \text{ s.t. } \texttt{active}_t = 1\}$ {Active blocks}
  if $\mathcal{A} = \emptyset$ **then**
    break
  end if
  $n \leftarrow |\mathcal{A}|$
  if $2^n \leq M$ **then**
    Enumerate all 2^n coalitions $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(2^n)} \in \{0,1\}^n$
    $N_s \leftarrow n$ (only takes averages over n active blocks)
  else
    Sample M coalitions $\mathbf{z}^{(m)} \sim Q_n^*$ for n players
    $N_s \leftarrow M$
  end if
  $\Phi_j \leftarrow 0$ for all $j \in \mathcal{A}$ {Block-level accumulators}
  for $m = 1$ to N_s **do**
    Expand $\mathbf{z}^{(m)}$ to full mask $\bar{\mathbf{z}} \in \{0,1\}^T$: set $\bar{z}_t \leftarrow z_i^{(m)}$ for all $t \in \mathcal{B}_{\mathcal{A}(i)}$ and 0 for other steps
    Simulate counterfactual $\tau^{\bar{\mathbf{z}}}$ via CTFSIM$(\bar{\mathbf{z}}, \tau, \pi, \pi_{\text{base}})$
    $Y^{\bar{\mathbf{z}}} \leftarrow \sum_{t=1}^{T} \gamma^{t-1} y_t^{\bar{\mathbf{z}}}$; $\quad \delta \leftarrow Y - Y^{\bar{\mathbf{z}}}$
    if $2^n \leq M$ **then**
      Compute exact kernel weights $\kappa_i(\mathbf{z}^{(m)}) = \frac{1}{n\binom{n-1}{k-1}}$ if $z_i^{(m)} = 1$, else $\frac{-1}{n\binom{n-1}{k}}$
    else
      Compute importance-weighted kernel weights $\kappa_i(\mathbf{z}^{(m)})$ using Q_n^*
    end if
    for $i = 1$ to n **do**
      $\Phi_{\mathcal{A}(i)} \mathrel{+}= \delta \cdot \kappa_i(\mathbf{z}^{(m)}) \ / \ N_s$
    end for
  end for
  $\hat{\phi}_{1:T} \leftarrow \mathbf{0}$ {Reset; only record active blocks this round}
  for $j \in \mathcal{A}$ **do**
    $\hat{\phi}_t \leftarrow \Phi_j \ / \ |\mathcal{B}_j| \quad$ for all $t \in \mathcal{B}_j$ {Uniform distribution within block}
  end for
  for $j \in \mathcal{A}$ **do**
    if $|\Phi_j| \ / \ |\mathcal{B}_j| < \epsilon$ **then**
      $\texttt{active}_t \leftarrow 0, \quad \hat{\phi}_t \leftarrow 0 \quad$ for all $t \in \mathcal{B}_j$
    end if
  end for
  if $N_r \geq T$ **then**
    break {Reached per-step resolution}
  end if
end for

return $\hat{\phi}_{1:T}$