STRUCTURED WORLD MODELS FOR SAMPLE-EFFICIENT PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-based decision making often involves partially known dynamical systems, where structural knowledge (e.g., stock-flow relations, conservation laws, delays, and mechanistic dependencies) is available while some dynamics remain unknown. Yet general-purpose world models often learn system dynamics largely from data without explicitly encoding this known structure. Starting from supplied System Dynamics (SD) stock–flow structure, we give a constructive, transition-complete procedure for turning it into a decision-ready interface: embedded decision rules are externalized as actions, prediction-relevant delay and pipeline memory are retained, and learning is restricted to unknown structural parameters or residual dynamics. The resulting transition model is not tied to a particular planner; in our discrete-action experiments, we primarily pair it with Monte Carlo tree search. In a realizable setting, our finite-sample analysis shows that identification scales with the structural dimension d and propagates transition error through multi-step rollouts to planning values and action selection under a sufficient action gap. Under limited interaction data, structured models achieve near-oracle decisions in epidemic control and, on the T4 Beer Game at 1k interactions, substantially lower multi-step rollout error and better planning performance than a black-box model; on a ten-node supply network, 1k interactions suffice to reach a return that DreamerV3 attains only after 100k. The planning advantage persists across serial, branching, and multi-source graph structures, while integration studies with existing model-based reinforcement learning algorithms demonstrate that structural knowledge can also be used beyond the native MCTS setting. These results suggest that explicitly preserving known system structure can provide a practical route to more sample-efficient model-based decision making.

1 INTRODUCTION

Many sequential decision problems require anticipating how a dynamical system will evolve under alternative actions. Model-based methods address this need by learning or constructing a predictive world model and using it to evaluate future consequences (Moerland et al., 2023). In many scientific, engineering, and operational systems, however, only part of the dynamics is unknown: structural knowledge such as stock–flow relations, conservation laws, delay structures, or known physical interactions may already be available before learning. The unknown part may be limited to parameters within an established structural family or to local dynamics not captured by the supplied structure. This motivates our central question: *how can existing structural knowledge be organized into a decision-ready world model for sequential decision making when only part of the underlying dynamics must be learned from data?*

Prior work spans both learned world models for model-based decision making and physics-informed or grey-box approaches that combine structural priors with data-driven dynamics (Hafner et al., 2025; Hansen et al., 2024; Karniadakis et al., 2021; Lutter et al., 2019). We focus on a representation and interface-design question that arises when credible system-level structure is already available: which variables are controlled by the agent whose action is being planned, which remain environmental or exogenous, and what dynamic memory must be retained for recursive prediction? In a standard Beer Game SD model, each echelon's ordering rule is an endogenous equation. To use the model for decision making, that rule must be externalized as an action input, while in-transit pipelines must be

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

retained as state for recursive prediction. We refer to the resulting action-conditioned, recursively queryable representation as a *decision-ready structured world model*.

We develop this idea by reorganizing supplied structural knowledge around a focal decision-maker while separating structural modeling, data-driven identification, and downstream planning. In our primary System Dynamics (SD) instantiation (Forrester, 1961), system stocks together with the dynamic memory required for future prediction form the decision state, focal decision rules are externalized as actions, and the remaining feedback, flow, and delay relations define the structural transition backbone. Learning is then concentrated on the remaining unknown components, either low-dimensional parameters within the structural model or, when needed, learned residual dynamics. The resulting one-step transition interface is kept separate from the decision solver. Our main discrete-action experiments query this model with finite-budget Monte Carlo tree search (MCTS), while additional studies examine complementary ways of incorporating structural knowledge into existing model-based reinforcement learning algorithms.

We further show that, when the true dynamics belong to the prescribed structural family, a sufficient identification bound scales with the structural dimension d, and we trace the resulting error through multi-step rollouts and planning values to preservation of the selected action under a sufficient action gap. Empirically, structured models reach near-oracle decisions in epidemic control with the smallest evaluated budget, and on the ten-node Beer Game SD-MCTS with $1{,}000$ interactions reaches a return that DreamerV3 exceeds only after $100{,}000$. The planning advantage persists across serial, branching, and multi-source graph structures, while structural knowledge can also be integrated into DreamerV3 (Hafner et al., 2025) through training-time imagination and into TD-MPC2 (Hansen et al., 2024) through its transition representation.

Contributions. Our contributions are threefold: **(1) Decision-ready structured world modeling for partially known dynamics.** Given supplied structural knowledge and a focal decision-maker, we provide a constructive procedure with an explicit transition-completeness criterion that organizes focal decision variables, environmental mechanisms, exogenous inputs, and prediction-relevant dynamic memory into an action-conditioned, recursively queryable world model. Unknown structural parameters and optional residual dynamics are then learned within this representation. **(2) Finite-sample connection from identification to planning.** In the realizable structured setting, we connect effective parameter dimension to finite-sample identification, multi-step rollout error, planning-value error, and action selection under a sufficient action gap. **(3) Evaluation across finite-data regimes, graph structures, and integration interfaces.** We evaluate the construction under limited interaction data, across serial, branching, and multi-source graph structures, and through two complementary integration case studies with existing model-based reinforcement learning algorithms.

2 Related Work

Model-based world models. Model-based reinforcement learning (MBRL) has developed a broad range of predictive models for planning, value estimation, and policy optimization, from learned latent world models to approaches that incorporate structured or physical inductive biases (Moerland et al., 2023; Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; 2025; Schrittwieser et al., 2020; Chua et al., 2018; Hansen et al., 2022; 2024; Ramesh & Ravindran, 2023). This broader literature suggests a spectrum in how much of the dynamics is learned from data and how much is supplied by prior knowledge. Our setting lies toward the partially known end of this spectrum, where credible system-level structure is available but some parameters or unmodeled dynamics remain unknown.

Physics-informed and grey-box dynamics. Physics-informed and grey-box learning incorporates physical or structural priors into learned dynamics through conservation laws, Hamiltonian or Lagrangian structure, differential-equation constraints, and adaptive network architectures (Karniadakis et al., 2021; Raissi et al., 2019; Greydanus et al., 2019; Lutter et al., 2019; Rayas et al., 2022; Wang et al., 2024). Related model-based RL methods use physics-informed dynamics, jointly learn physical corrections and data-driven residuals, or add robust adaptive control around a learned model (Ramesh & Ravindran, 2023; El Asri et al., 2024; Lutter et al., 2021; Liu & Wang, 2021; Sung et al., 2024). These methods typically assume a specified state–action interface, often for mechanical systems (Ramesh & Ravindran, 2023; El Asri et al., 2024; Lutter et al., 2021). We instead construct

the decision-ready interface from system-level stock–flow knowledge, including externalized decision rules and prediction-relevant dynamic memory.

System Dynamics and decision models. System Dynamics (SD) provides one such system-level representation through stocks, flows, feedback relations, and delays, and has long been used for simulation and policy analysis (Forrester, 1961). Related work has also optimized policy rules or parameters inside SD models, reformulated SD equations for white-box optimization, and wrapped existing SD models as reinforcement-learning environments (Duggan, 2008; Vierhaus et al., 2017; Klu et al., 2023). Building on this structural representation, we reorganize the decision state, focal action inputs, exogenous drivers, and remaining system mechanisms into an action-conditioned, recursively queryable one-step world model. The resulting transition model is kept separate from the downstream solver so that the same structural representation can support different decision procedures.

3 Problem Formulation

We consider a discounted Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ (Sutton & Barto, 2018), where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition kernel, R is the reward function, and $\gamma \in (0, 1)$ is the discount factor. A policy $\pi(a_t \mid s_t)$ seeks to maximize the expected discounted return $J(\pi) = \mathbb{E}_{\tau \sim (\pi, P)}[\sum_{t=0}^{\infty} \gamma^t r_t]$.

We focus on partially known dynamical systems in which credible structural knowledge is available while part of the transition dynamics must still be identified from data. Let G denote the supplied structural knowledge, such as stock–flow topology, conservation relations, delay structure, or known mechanistic interactions, and let e_t denote exogenous inputs outside the current decision boundary. During planning, e_t is observed, known, or forecast; when needed for a Markov representation, it can be included in an augmented state. For the vector-valued state representations used below, we write the true one-step dynamics as

$$s_{t+1} = F_G(s_t, a_t, e_t; \theta^\star) + \Delta^\star(s_t, a_t, e_t) + \xi_t, \tag{1}$$

where F_G is the transition component specified by G, $\theta^\star$ denotes unknown quantities within the structural family, $\Delta^\star$ captures systematic dynamics not represented by that component, and ξ_t denotes stochastic disturbance. This decomposition separates what is supplied by prior structural knowledge from what remains to be learned. When $\Delta^\star = 0$, the problem reduces to structured identification within the supplied transition family.

For such structural knowledge to serve as a world model for sequential decision making, the representation must be decision-ready. The state s_t must contain the information required for future prediction, including relevant stock, delay, pipeline, or memory variables; focal control variables must enter explicitly through the action a_t rather than remain embedded in an internal decision rule; and each one-step prediction must produce a state that can be recursively reused in subsequent model calls. Task-relevant feasibility relations may also be supplied as part of the structural knowledge, while the general formulation remains independent of a particular enforcement mechanism or downstream planner.

These requirements define the central modeling problem addressed in this work: to make supplied structural knowledge the transition backbone of a decision-ready structured world model. The state, action variables, exogenous inputs, and retained memory are organized so that the resulting transition is action-conditioned and recursively queryable, while learning is concentrated on the components not fixed by G. Section 5 analyzes the realizable special case $\Delta^\star = 0$ with a correctly specified low-dimensional structural family $\theta^\star \in \mathbb{R}^d$; additional assumptions required by individual results are introduced there.

4 Method

We instantiate the partially known dynamics introduced in Section 3 as a decision-ready structured world model. As illustrated in Figure 1, supplied structural knowledge defines an action-conditioned dynamical backbone, transition data identify the remaining unknown components, and downstream planners query the resulting one-step model for decision making. We first construct the structured

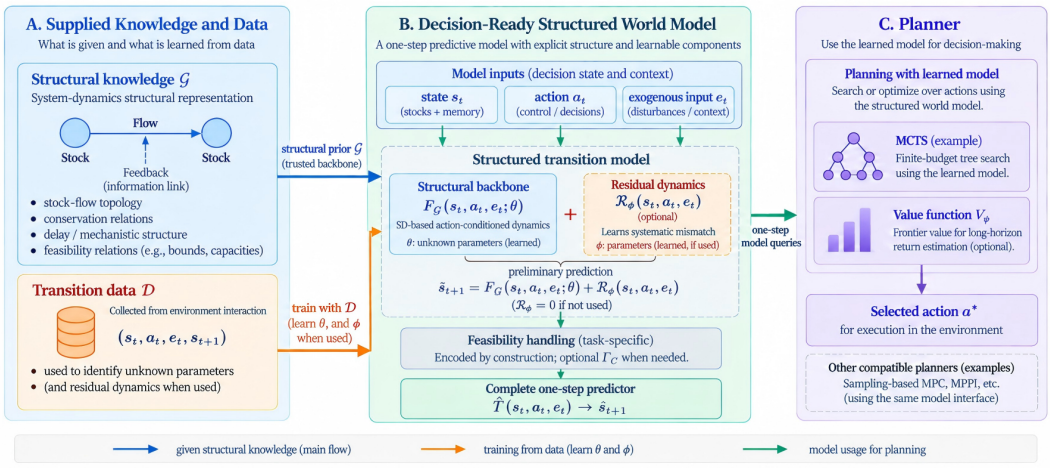

Figure 1: Overview of the SD-based structured world-model framework. Supplied structural knowledge defines the dynamical backbone, transition data identify unknown components, and downstream planners query the resulting one-step predictor for decision making.

transition from System Dynamics (SD) knowledge, then learn unknown parameters and, when needed, residual dynamics, and finally expose the resulting predictor through a planner-independent interface.

4.1 Constructing a Decision-Ready SD World Model

System Dynamics represents system evolution through stocks, flows, feedback relations, delays, and auxiliary equations (Forrester, 1961). We reorganize the supplied SD knowledge into an action-conditioned one-step transition model that can be repeatedly queried by a sequential decision algorithm.

Let G denote the supplied structural knowledge, including stock–flow topology, conservation relations, known delay structures, and other trusted mechanistic relations. We construct the decision state s_t from system stocks together with any dynamic memory required for future prediction, expose variables controlled by the focal agent as the action a_t, and represent drivers outside the current decision boundary as exogenous inputs e_t, assumed known or forecast during planning. Given (s_t, a_t, e_t), the supplied structure defines the one-step update $s^G_{t+1} = F_G(s_t, a_t, e_t; \theta)$, where θ denotes numerical quantities in the structured model, with unknown values identified from data when needed.

For a discrete-time stock x_i, the structural update may, for example, take the form

$$x_{i,t+1} = x_{i,t} + h\left(\sum_{j\in\mathcal{I}_i} q_{ji,t} - \sum_{k\in\mathcal{O}_i} q_{ik,t}\right), \qquad q_t = Q_G(s_t, a_t, e_t; \theta), \tag{2}$$

where h is the simulation step, $\mathcal{I}_i$ and $\mathcal{O}_i$ denote the incoming and outgoing flow sets, and Q_G collects the flow and auxiliary relations defined by the supplied structure. Algebraic auxiliary variables can be recomputed at each model call, whereas delay chains, in-transit pipelines, and other persistent variables are included in s_t whenever they affect future evolution.

Existing SD models may also contain endogenous rules that automatically generate control variables from the current system state. When such a variable is controlled by the focal agent, we externalize the corresponding rule and expose the variable as a_t. Physical, operational, and non-focal-agent mechanisms remain inside F_G. This separates who chooses the action from how the system responds to that action. Appendix A.2 states the full construction as a six-step procedure.

After this externalization, the model must remain transition-complete: for every admissible (s_t, a_t, e_t), the structural equations determine the next decision state together with all dynamic memory required

for subsequent model calls. The same one-step transition can therefore be queried recursively by a planner. Supplied feasibility relations, such as conservation, non-negativity, capacity, and admissible-flow constraints, are enforced by construction whenever possible; any additional feasibility handling required after learnable components are introduced is described in Section 4.2.

4.2 Learning the Unknown Dynamics

The structural construction in Section 4.1 specifies the part of the transition supplied by prior knowledge. The remaining uncertainty is represented through unknown structural parameters θ and, when parameter adjustment alone is insufficient, a learnable residual R_ϕ. We write the resulting hybrid prediction as $\tilde{s}_{t+1} = F_G(s_t, a_t, e_t; \theta) + R_\phi(s_t, a_t, e_t)$, where R_ϕ captures systematic transition components not represented by the structured backbone. When the supplied feasibility relations are already preserved by construction, no additional operation is required. Otherwise, task-specific feasibility handling can be applied as $\hat{s}_{t+1} = \Gamma_{\mathcal{C}}(\tilde{s}_{t+1})$, where $\Gamma_{\mathcal{C}}$ reduces to the identity when no additional correction is needed. We denote the resulting complete one-step predictor, including feasibility handling when required, by $\widehat{\mathcal{T}}(s_t, a_t, e_t)$.

Parameter-only and residual learning. When the supplied structural family is sufficiently expressive, we set $R_\phi \equiv 0$ and estimate only the unknown quantities θ within F_G, preserving the supplied topology, conservation relations, delay structure, and mechanistic equations. When parameter adjustment alone cannot capture systematic transition discrepancies, R_ϕ provides an additional data-driven correction. The additive form above is used for notation; in implementation, the residual may instead be attached to selected components of the structured model.

Where one-step prediction is used, the learnable components are fitted from replay transitions $\mathcal{D} = \{(s_t, a_t, e_t, s_{t+1})\}$ using the one-step prediction objective

$$\mathcal{L}_{\text{model}}(\theta, \phi) = \mathbb{E}_{(s_t, a_t, e_t, s_{t+1}) \sim \mathcal{D}} \left[\| s_{t+1} - \hat{s}_{t+1} \|_2^2 \right]. \tag{3}$$

Task-specific multi-step or host-algorithm objectives use the same structured transition construction and are detailed in the corresponding experimental appendices.

Here R_ϕ captures systematic discrepancy predictable from the model inputs, whereas the disturbance ξ_t introduced in Section 3 represents irreducible stochastic variation. Section 5 analyzes the parameter-only special case $R_\phi \equiv 0$.

4.3 Planning with the Structured World Model

The learned predictor $\widehat{\mathcal{T}}$ provides a planner-independent one-step transition interface. In our discrete-action experiments, we instantiate this interface with finite-budget Monte Carlo tree search (MCTS). At each decision step, MCTS allocates a finite simulation budget using a standard UCB/PUCT-style tree policy (Kocsis & Szepesvári, 2006; Browne et al., 2012; Silver et al., 2016; 2017), while successor states are generated recursively as $\hat{s}_{t+k+1} = \widehat{\mathcal{T}}(\hat{s}_{t+k}, a_{t+k}, e_{t+k})$. Recursive model queries therefore carry the structural and feasibility relations encoded in the learned world model into imagined trajectories.

Because the search depth is finite, a simulation reaching frontier depth ℓ is evaluated by combining the explicitly accumulated return with an estimate of the continuation beyond the explored trajectory:

$$\hat{G}_{t,\ell} = \sum_{k=0}^{\ell-1} \gamma^k \hat{r}_{t+k} + \gamma^\ell \widehat{C}(\hat{s}_{t+\ell}), \tag{4}$$

where $\widehat{C}$ denotes a task-dependent continuation estimate. In our experiments, it is instantiated either by a short rollout under a fixed default policy or, when a learned value estimator is available, by $\widehat{C}(\hat{s}_{t+\ell}) = V_\psi(\hat{s}_{t+\ell})$ (Bertsekas & Tsitsiklis, 1995; Silver et al., 2017). The resulting estimate is backed up through the search tree, with model and value parameters fixed during each planning call.

The same transition interface can equally be queried by other model-based planners, such as model-predictive solvers (Qin & Badgwell, 2003; Kamthe & Deisenroth, 2018; Hewing et al., 2020; Hansen et al., 2022).

5 Theoretical Analysis

We analyze the realizable, parameter-only case of the structured transition introduced in Section 4. Writing this transition as $\mathcal{P}$ and suppressing a known or forecast exogenous-input sequence for notation, we have

$$s_{t+1} = \mathcal{P}(s_t, a_t; \theta^\star) + \xi_t, \qquad \theta^\star \in \mathbb{R}^d, \tag{5}$$

where the true dynamics belong to the prescribed structural family. The analysis assumes standard identifiability, sub-Gaussian noise, and Lipschitz regularity conditions (Ljung, 1999; Vershynin, 2018; Asadi et al., 2018); full assumptions and proofs are deferred to Appendix B. We connect these identification and Lipschitz-propagation ingredients through the decision-ready interface to planning-value and action-selection guarantees.

Finite-sample structured identification. Under the identifiability condition with information lower bound $\mu > 0$ (Ljung, 1999) and σ-sub-Gaussian transition noise, a sufficient sample size for estimating $\theta^\star$ to accuracy $\|\hat{\theta} - \theta^\star\|_2 \le \epsilon_\theta$ with probability at least $1 - \delta$ scales as

$$N = \mathcal{O}\left(\frac{\sigma^2}{\mu}\frac{d + \log(1/\delta)}{\epsilon_\theta^2}\right). \tag{6}$$

Thus, when the supplied structural family is valid, estimation complexity is governed by the effective structural dimension d rather than by an unrestricted representation of the transition map. In our experiments, $d = 6$ for epidemic control and $d = 3$ for the Graph Beer Game, whose black-box counterpart has 130,128 trainable parameters.

Multi-step rollout error. If $\mathcal{P}$ is L_θ-Lipschitz in its parameters, the one-step model error satisfies $\eta_N \le L_\theta \|\hat{\theta} - \theta^\star\|_2$. If the transition is additionally L_s-Lipschitz in the state (Asadi et al., 2018), the expected H-step rollout discrepancy satisfies

$$\varepsilon_H^{\text{roll}} \le L_s^H \varepsilon_0^{\text{roll}} + \eta_N \sum_{j=0}^{H-1} L_s^j. \tag{7}$$

Hence, the propagation of estimation error depends jointly on model accuracy and dynamical sensitivity: it remains bounded for contractive dynamics ($L_s < 1$), grows at most linearly at $L_s = 1$, and may grow geometrically when $L_s > 1$.

Planning-value error. For the planning-value analysis, we specialize the generic frontier evaluation in Section 4.3 to the learned continuation-value case $\widehat{C} = V_\psi$. Let $\hat{Q}_H^\pi$ denote the H-step bootstrapped value computed from the learned model with terminal estimator V_ψ. If the reward and continuation value are Lipschitz and $\|V_\psi - V^\pi\|_\infty \le \epsilon_V$, then

$$\left|\hat{Q}_H^\pi(s,a) - Q_H^\pi(s,a)\right| \le L_r \sum_{k=0}^{H-1} \gamma^k \varepsilon_k^{\text{roll}} + \gamma^H \left(L_V \varepsilon_H^{\text{roll}} + \epsilon_V\right). \tag{8}$$

The bound separates model-induced reward and terminal-state errors from the discounted continuation-value approximation error $\gamma^H \epsilon_V$. The rollout-based frontier evaluation used in our main experiments is the special case in which a fixed, model-independent rollout policy extends the model horizon to H' with zero terminal value, so the bound applies to the corresponding H'-step value with $\epsilon_V = 0$.

Decision stability. For exact maximization over a finite candidate action set, write $Q = Q_H^\pi$ and $\hat{Q} = \hat{Q}_H^\pi$, and let $a^\star$ be the unique maximizer of $Q(s,\cdot)$. Define the action gap as $\Delta_Q(s) = Q(s,a^\star) - \max_{a \ne a^\star} Q(s,a)$. If

$$\sup_a \left|\hat{Q}(s,a) - Q(s,a)\right| < \frac{\Delta_Q(s)}{2}, \tag{9}$$

then $\arg\max_a \hat{Q}(s,a) = a^\star$.

6 EXPERIMENTS

We evaluate the proposed structured world-model framework through three empirical questions. **Q1:** Under limited interaction data, does structural parameterization improve predictive accuracy and downstream decision performance relative to black-box transition learning? **Q2:** How does the same modeling construction perform across serial, branching, and multi-source graph structures? **Q3:** Can structural knowledge be incorporated into existing model-based reinforcement learning algorithms while retaining their core learning or planning procedures?

6.1 EXPERIMENTAL SETUP

We evaluate three task families within the partially known grey-box setting of Section 3. In epidemic control, the SEIHR stock–flow structure and clinical transition mechanisms are supplied, while transmission- and intervention-related parameters are identified from data. The Graph Beer Game, built on the classical Beer Game setting (Sterman, 1989; Oroojlooyjadid et al., 2022), supplies inventory, backlog, shipment, order-delay, and network structure, enabling evaluation under delayed feedback and supply-chain amplification effects associated with information distortion (Lee et al., 1997), as well as varying graph topology. In Hopper-Hop from the DeepMind Control Suite (Tassa et al., 2018), an approximate mechanistic transition backbone is supplemented by learned residual dynamics and integrated into an existing continuous-control algorithm.

For the epidemic-control and Graph Beer Game world-model comparisons, structured and black-box models use the same downstream MCTS configuration and matched real-environment interaction budgets to isolate the effect of the model class. Epidemic-control results use 10 random seeds, Graph Beer Game and DreamerV3 use five, and Hopper-Hop uses three. DreamerV3 paired return differences are reported with 95% bootstrap confidence intervals. Full protocols, additional ablations, OOD evaluations, and Cartpole results are provided in the appendix.

6.2 FINITE-DATA MODEL LEARNING AND DECISION PERFORMANCE

Finite-data decision performance. In epidemic control, we compare a context-augmented black-box transition model (A1'), a learned structured model (A2), and a true-parameter structured reference (A4) under the same finite-budget MCTS planner. A2 matches the true-parameter reference from the smallest evaluated budget onward (-37.33 versus -37.42 at 52 weekly decisions), with a cross-seed standard deviation below 0.25 at every budget. A1' remains 5–11 return points lower at all five budgets, including -42.63 at 1,040 decisions, with roughly 18–36 times larger cross-seed variation. Structural parameterization thus reaches near-oracle decisions with the least data we evaluate, consistent with the identification argument in Section 5.

Multi-step prediction and planning. On the ten-node T4 Beer Game at 1,000 environment interactions, the structured model contains three learnable parameters, compared with 130,128 trainable parameters in the black-box residual network. Both share the same replay buffer, update schedule, and MCTS configuration. Figure 2(b) reports rollout RMSE over horizons $H = 1, \ldots, 8$, obtained by recursively predicting along recorded action sequences and comparing the predicted states with the environment trajectory.

The structured model exhibits substantially lower rollout error across the evaluated horizons; at $H = 8$, its RMSE is 0.593, compared with 13.470 for the black-box model. Under the same T4 evaluation setting, SD-MCTS achieves a mean return of -3404.5, compared with -26748.6 for BB-MCTS. The lower rollout error and higher planning return are consistent with the error-propagation pathway analyzed in Section 5. For reference, the true-parameter SD-oracle attains -3253.6, so SD-MCTS at 1k interactions lies within 5% of it. A base-stock rule using the true design demand attains -3040.0; this remaining gap stems from the planner, since MPC on the same structured model yields -2983.6 (Appendix E.5).

As a complementary structural diagnostic, we measure the fraction of states imagined during MCTS expansion and leaf rollouts in which any selected physical quantity becomes negative. The violation rate is 4.52×10^{-4} for SD-MCTS and 0.9396 for BB-MCTS.

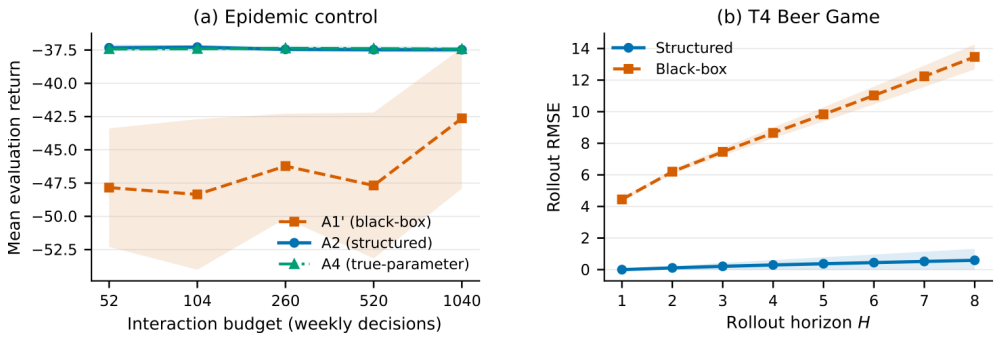

Figure 2: Finite-data model and decision performance. (a) Epidemic-control returns across interaction budgets. (b) T4 Beer Game rollout RMSE at 1,000 environment interactions. Shading indicates one sample standard deviation across seeds.

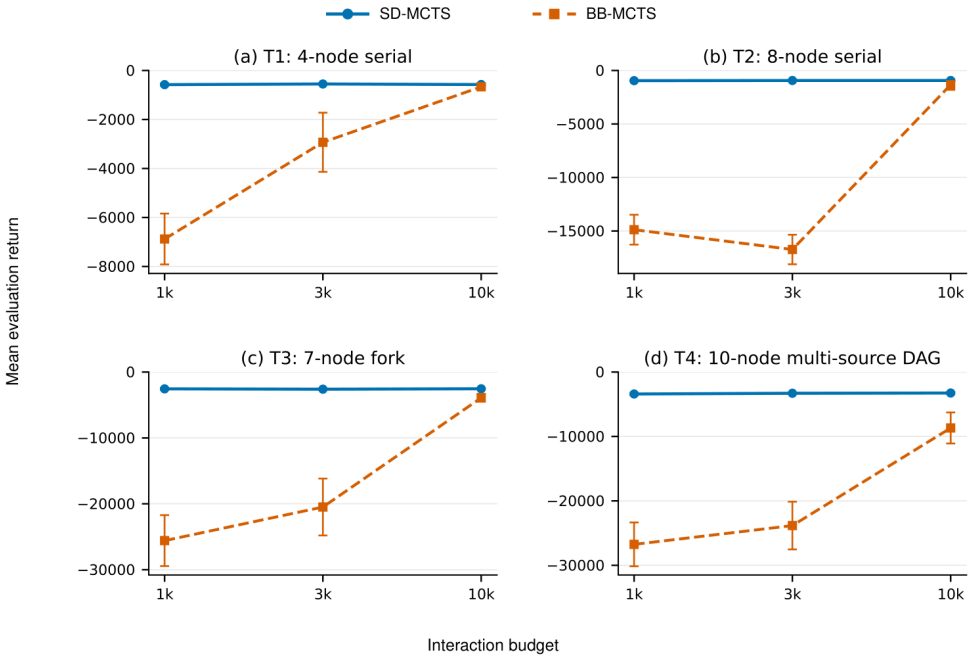

Figure 3: Planning performance across Beer Game graph structures. Mean returns of SD-MCTS and BB-MCTS are shown for T1–T4 at interaction budgets of 1k, 3k, and 10k. Error bars indicate one sample standard deviation across five seeds. Y-axis ranges differ across panels to preserve within-topology resolution.

6.3 Evaluation Across Graph Structures

We instantiate the same graph-based stock–flow construction on a four-node serial chain (T1), an eight-node serial chain (T2), a seven-node fork network (T3), and a ten-node multi-source directed acyclic graph (T4). These environments span increasing graph depth, branching, and multi-source structure while retaining the same basic inventory-flow and delay mechanisms.

As shown in Figure 3, across T1–T4 and all interaction budgets of 1k, 3k and 10k, SD-MCTS achieves higher returns than BB-MCTS on every seed; the 95% paired bootstrap intervals exclude zero in all twelve topology–budget cells, and the margins are largest at 1k–3k. Because demand and capacity scale with the topology, we focus on method comparisons within each graph. These results show that

the advantage of the structured construction is retained across the evaluated serial, branching, and multi-source systems. Full topology-by-budget results are reported in the appendix.

To relate these finite-data gains to standard learning-based agents, Table 16 in the appendix reports PPO and pure DreamerV3 trained on T4 under the same environment and evaluation protocol. At 10k interactions, PPO and DreamerV3 reach mean returns of -7536.7 and -15381.6, whereas SD-MCTS reaches -3404.5 with 1k interactions. DreamerV3 surpasses this level only at 100k interactions (-2937.6; -7536.5 at 30k). On this task, the structured model therefore reaches comparable performance with roughly two orders of magnitude fewer interactions, while a fully learned world model attains a higher return once a large interaction budget is available.

6.4 Integration Case Studies with Existing Model-Based RL

Structure-guided DreamerV3. On the T4 Beer Game, we retain DreamerV3's learned world model and actor–critic objectives, but replace imagined actor actions with recommendations from a structural action guide, computed from the inventory position implied by the supplied stock–flow and delay structure, with probability β, linearly annealed from 0.2 to zero over the first half of training. The guide uses the design-demand scale to instantiate a modular imagination-time interface; environment collection and evaluation remain actor-driven.

Across the five evaluated budgets, structure-guided DreamerV3 attains higher mean returns at 3k, 10k, and 30k interactions. The paired improvement over pure DreamerV3 is significant at 3k ($+9124$, 95% CI $[3465, 13920]$) and 30k ($+3036$, 95% CI $[691, 7241]$), while the intervals at 1k, 10k, and 100k include zero (Appendix F.2). Structural guidance therefore accelerates DreamerV3 training in the intermediate-data regime, with no significant difference at 100k interactions.

Structured dynamics in TD-MPC2. On Hopper-Hop, we replace the transition representation with 17 structured state dimensions and 495 residual latent dimensions while retaining TD-MPC2's reward, value, and policy learning components with its MPPI planner (Hansen et al., 2024); the structured core is initialized from supervised Hopper transitions before online TD-MPC2 training (Appendix F.3). Over three seeds, the structured variant achieves mean returns of 61.14 and 206.90 at 100k and 200k interactions. For context, the public TD-MPC2 reference trace, not a paired local run, reports 13.63 and 166.63 at those interaction counts (Hansen et al., 2024). The structured–residual representation therefore operates inside TD-MPC2 while maintaining competitive control performance.

Together, these case studies show that supplied structural knowledge can be incorporated through different interfaces without replacing the host algorithm's core learning or planning machinery.

7 Discussion and Conclusion

We organized supplied structural knowledge of partially known dynamical systems into an action-conditioned, recursively queryable world model that concentrates learning on the remaining unknown dynamics. In the realizable low-dimensional case, our analysis connects structural parameterization to finite-sample identification and its downstream consequences for multi-step prediction and planning accuracy. Empirically, structured modeling improves finite-data decision performance in epidemic control and yields substantially lower multi-step prediction error and stronger planning performance in the Graph Beer Game. The same construction retains its advantage across the evaluated serial, branching, and multi-source graph structures, while the DreamerV3 and TD-MPC2 studies show that supplied structure can also be incorporated into existing model-based RL systems through different interfaces. Together, these results support retaining reliable structural knowledge in the world model and focusing data-driven learning on the remaining unknown dynamics in finite-data settings. The gains are largest at small budgets; fully learned models narrow the gap and can overtake the structured model once interaction data are plentiful.

The framework assumes credible supplied structure together with an adequate decision boundary and state representation, and the present theory focuses on correctly specified low-dimensional structural families. An important next step is to handle uncertainty or incompleteness in the structure itself, including more automatic extraction of decision-ready interfaces and joint identification of structural and residual components. More broadly, for partially known dynamical systems, world-model design can preserve what is already known and concentrate learning on what remains uncertain.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research ideation and execution, including feedback on conceptual framing, methodology, experimental design, and the development and checking of mathematical arguments and proofs. Generative AI tools were also used for literature retrieval and discovery, drafting parts of the manuscript, and editing the paper for clarity and readability.

All AI-assisted technical content was reviewed by the authors. Mathematical claims and proofs were checked against the underlying derivations, experimental interpretations were checked against the reported results, and literature-related content and citations were verified against the original sources. All final methodological decisions, mathematical claims, experimental configurations, reported results, citations, and conclusions were reviewed by the authors.

REPRODUCIBILITY STATEMENT

We provide supplementary materials to support the reproducibility and verification of both the theoretical and empirical results. The main paper specifies the structured world-model formulation, learning and planning procedures, primary experimental settings, and evaluation measures. The appendix provides the complete assumptions and proofs for the theoretical analysis; common experimental protocols including interaction budgets, random seeds, evaluation procedures, and statistical reporting conventions; model-learning and optimization settings; planning configurations; and task-specific environment, model, and implementation details. It also reports complete budget-wise results, supplementary comparisons, ablations, and additional evaluations. An anonymous supplementary code package accompanying the submission provides the code and experiment materials associated with the reported studies. Together, the main paper, appendix, and supplementary code package are intended to provide the information and artifacts needed to inspect, verify, and reproduce the reported methodology and results.

REFERENCES

Kavosh Asadi, Dipendra Misra, and Michael L Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 264–273. PMLR, 2018. URL https://proceedings.mlr.press/v80/asadi18a.html.

D.P. Bertsekas and J.N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pp. 560–564 vol.1, 1995. doi: 10.1109/CDC.1995.478953.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Jim Duggan. Equation-based policy optimization for agent-oriented system dynamics models. *System Dynamics Review*, 24(1):97–118, 2008. doi: 10.1002/sdr.393.

Zakariae El Asri, Olivier Sigaud, and Nicolas Thome. Physics-informed model and hybrid planning for efficient dyna-style reinforcement learning. *Reinforcement Learning Journal*, 2:693–713, 2024. URL https://rlj.cs.umass.edu/2024/papers/Paper99.html.

Jay W. Forrester. *Industrial Dynamics*. MIT Press, Cambridge, MA, 1961.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman

Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15353–15363, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy 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.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Control Tasks Through World Models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 8387–8406. PMLR, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020. doi: 10.1146/annurev-control-090419-075625.

Sanket Kamthe and Marc Peter Deisenroth. Data-efficient reinforcement learning with probabilistic model predictive control. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1701–1710. PMLR, 2018. URL http://proceedings.mlr.press/v84/kamthe18a.html.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-Informed Machine Learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Emmanuel Klu, Sameer Sethi, DJ Passey, and Donald Martin Jr. Sdgym: Low-code reinforcement learning environments using system dynamics models. *arXiv preprint arXiv:2310.12494*, 2023. URL https://arxiv.org/abs/2310.12494.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

Hau L. Lee, V. Padmanabhan, and Seungjin Whang. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558, 1997. doi: 10.1287/mnsc.43.4.546.

Xin-Yang Liu and Jian-Xun Wang. Physics-informed dyna-style model-based deep reinforcement learning for dynamic control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2255):20210618, 2021. doi: 10.1098/rspa.2021.0618.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BklHpjCqKm.

Michael Lutter, Johannes Silberbauer, Joe Watson, and Jan Peters. Differentiable physics models for real-world offline model-based reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4163–4170, 2021. doi: 10.1109/ICRA48506.2021.9561805.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118, 2023. doi: 10.1561/2200000086. URL https://doi.org/10.1561/2200000086.

Afshin Oroojlooyjadid, MohammadReza Nazari, Lawrence V. Snyder, and Martin Takáč. A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management*, 24(1):285–304, 2022. doi: 10.1287/msom.2020.0939.

S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003. doi: 10.1016/S0967-0661(02)00186-7.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Adithya Ramesh and Balaraman Ravindran. Physics-informed model-based reinforcement learning. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 26–37. PMLR, 2023. URL https://proceedings.mlr.press/v211/ramesh23a.html.

Anirudh Rayas, Rajasekhar Anguluri, and Gautam Dasarathy. Learning the structure of large networked systems obeying conservation laws. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. doi: 10.52202/068431-1064.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017. doi: 10.1038/NATURE24270. URL https://doi.org/10.1038/nature24270.

John D. Sterman. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science*, 35(3):321–339, 1989. doi: 10.1287/mnsc.35.3.321.

Minjun Sung, Sambhu Harimanas Karumanchi, Aditya Gahlawat, and Naira Hovakimyan. Robust model based reinforcement learning using $\mathcal{L}_1$ adaptive control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=GaLCLvJaoF.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, 2018.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*, 2018. URL https://arxiv.org/abs/1801.00690.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 2018.

Ingmar Vierhaus, Armin Fügenschuh, Robert Gottwald, and Stefan Grösser. Using white-box nonlinear optimization methods in system dynamics policy improvement. *System Dynamics Review*, 33(2):138–168, 2017. doi: 10.1002/sdr.1583.

Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. Piratenets: Physics-informed deep learning with residual adaptive networks. *Journal of Machine Learning Research*, 25(402):1–51, 2024. URL http://jmlr.org/papers/v25/24-0313.html.

A Notation and Supplementary Method Details

A.1 Notation

For ease of reference, Table 1 summarizes the core notation used in the general framework and theoretical analysis. Symbols specific to individual experimental environments are defined locally in the corresponding appendices.

Table 1: Core notation used in the structured world-model framework and theoretical analysis.

Symbol	Definition
Decision problem	
$\mathcal{S}, \mathcal{A}$	State and action spaces.
s_t, a_t	Decision state and focal-agent action at step t.
e_t	Exogenous input supplied from outside the focal decision boundary at step t.
P	True environment transition kernel.
R, γ, π	Reward function, discount factor, and policy, respectively.
Structured world model	
G	Supplied structural knowledge, such as stock–flow topology, conservation relations, delay structure, and trusted mechanistic dependencies.
F_G	Structural transition backbone induced by the supplied structural knowledge G.
$\theta, \theta^\star$	Structural parameter vector and its true value, respectively.
$\Delta^\star$	Systematic dynamics not represented by the supplied structural family.
ξ_t	Stochastic transition disturbance at step t.
R_ϕ	Learned residual dynamics parameterized by ϕ, used to represent systematic transition components not captured by the structural backbone.
$\mathcal{C}, \Gamma_{\mathcal{C}}$	Task-specific feasible set and optional feasibility-enforcement operator, respectively.
$s^G_{t+1}, \tilde{s}_{t+1}, \hat{s}_{t+1}$	Successor state produced by the structural backbone alone, combined structured–residual prediction before optional feasibility handling, and final predicted successor state, respectively.
$\widehat{\mathcal{T}}$	Complete learned one-step transition interface, including task-specific feasibility handling when required.
Learning and identification	
$\mathcal{D}$	Transition dataset used to fit the learnable components of the world model.
$\mathcal{P}(s, a; \theta)$	Prescribed structured transition family used in the realizable theoretical analysis.
$\hat{\theta}$	Estimated structural parameter vector.
d	Effective dimension of the unknown structural parameter vector.

Continued on next page.

Table 1 continued.

Symbol	Definition
N	Number of transition observations in the finite-sample identification analysis.
μ	Lower bound associated with local identifiability of the structural parameterization.
σ	Sub-Gaussian scale of the transition noise in the finite-sample analysis.
δ	Failure probability in the finite-sample estimation guarantee.
ϵ_θ	Target accuracy for structural parameter estimation.
η_N	Uniform one-step systematic model error after estimation from N transition observations.
Planning and theoretical analysis	
H	Finite rollout or planning horizon used in the theoretical analysis.
ℓ	Frontier depth reached by an individual finite-budget search simulation.
$\widehat{C}$	Continuation estimate associated with a search-frontier state.
V_ψ, V^π	Learned continuation-value estimator and corresponding true continuation value under policy π, respectively.
$Q_H^\pi, \hat{Q}_H^\pi$	True and learned-model H-step bootstrapped planning values, respectively.
$\varepsilon_k^{\text{roll}}$	Expected rollout-state discrepancy after k model steps.
L_θ, L_s	Lipschitz constants of the structured transition with respect to structural parameters and state, respectively.
L_r, L_V	Lipschitz constants of the reward and continuation-value functions, respectively.
ϵ_V	Uniform approximation error of the learned continuation-value estimator.
$\Delta_Q(s)$	Action gap between the best and second-best true planning values at state s.

A.2 Constructing a Decision-Ready Structured World Model

Section 4 describes the components of the proposed structured world model and their roles in model-based decision making. Here we make the construction procedure explicit. Starting from supplied structural knowledge G, a specification of the focal decision-maker, and, when learning is required, transition data $\mathcal{D}$, the goal is to construct a recursively queryable one-step transition interface $\widehat{\mathcal{T}}(s_t, a_t, e_t)$.

Step 1: Specify the focal decision boundary. The construction first identifies the variables controlled by the focal decision-maker. An existing structural model may contain endogenous decision rules that automatically generate such control variables from the current system state. When a variable is assigned to the focal decision-maker, the corresponding internal rule is externalized and the variable is exposed as part of the action a_t. Decision rules associated with other actors or environmental mechanisms may remain endogenous to the structural model. This step determines who selects the action while retaining the system mechanisms that govern how the environment responds.

Step 2: Partition the model variables. The variables in the supplied model are then organized according to their role in sequential prediction. Variables that must persist across decision steps are included in the decision state s_t, variables controlled by the focal decision-maker form a_t, and drivers supplied from outside the current decision boundary are represented by e_t. Purely algebraic auxiliary variables that can be recomputed from the current model inputs need not be stored in the decision state.

Step 3: Ensure recursive state closure. The decision state is augmented until it contains all information required for subsequent model calls. In particular, delay chains, in-transit pipelines, or other dynamic memory variables are included whenever they affect future evolution. The resulting representation should allow the successor state produced at one model call to serve directly as the state input to the next:

$$(s_t, a_t, e_t) \longrightarrow s_{t+1} \longrightarrow s_{t+2} \longrightarrow \cdots . \tag{10}$$

Accordingly, any prediction-relevant information that cannot be reconstructed from the current state must be incorporated into s_t rather than left as unrepresented internal history.

Step 4: Construct the structural transition backbone. After the decision boundary and state representation have been fixed, the remaining trusted mechanisms define the structural transition backbone

$$s_{t+1}^{G} = F_G(s_t, a_t, e_t; \theta). \tag{11}$$

These mechanisms may include stock–flow relations, conservation relations, known delays, mechanistic dependencies, and endogenous rules that remain outside the focal decision. The parameter vector θ collects numerical quantities within this structural family whose values may be known or may need to be identified from data. Thus, preserving the structural backbone does not require all of its numerical parameters to be known in advance.

Step 5: Assign the learnable components. Learning is concentrated on the parts of the transition that remain uncertain after the structural backbone has been specified. When the supplied structure is sufficiently expressive and the remaining uncertainty consists only of unknown numerical quantities, learning reduces to estimating θ and the residual is omitted, $R_\phi \equiv 0$. When parameter adjustment alone cannot represent a predictable systematic discrepancy, a learned residual component $R_\phi(s_t, a_t, e_t)$ may be added. The residual is intended to represent systematic dynamics that are not captured by the supplied backbone; missing prediction-relevant state or incorrectly imposed structural relations instead require revision of the model specification itself.

Step 6: Enforce feasibility when required and expose the transition interface. The structural and learned components produce the combined successor prediction

$$\tilde{s}_{t+1} = F_G(s_t, a_t, e_t; \theta) + R_\phi(s_t, a_t, e_t). \tag{12}$$

When the supplied structure or parameterization already preserves the required feasibility conditions, no additional operation is needed. Otherwise, a task-specific operator $\Gamma_{\mathcal{C}}$ may be applied to obtain the final successor state. The resulting complete predictor is exposed to the downstream decision procedure through the common interface

$$\hat{s}_{t+1} = \widehat{\mathcal{T}}(s_t, a_t, e_t). \tag{13}$$

The planner therefore queries the same one-step interface regardless of whether the underlying implementation uses a parameter-only structural model, a structured–residual model, or additional feasibility handling.

The parameter-only case $R_\phi \equiv 0$ corresponds to the realizable structured setting analyzed in Section 5. The concrete roles of the structural backbone, learned components, and downstream decision procedure in each experimental study are summarized in Appendix A.3; task-specific model definitions, optimization settings, and planning configurations are reported in the corresponding experimental appendices.

A.3 Framework Instantiations and Structural Integration Interfaces

The construction in Appendix A.2 describes how supplied structural knowledge can be organized into a decision-ready one-step transition interface. The experimental studies use these structural components in several related forms. Epidemic control, the Graph Beer Game, and Cartpole-Swingup instantiate parameter-only structured transitions with $R_\phi \equiv 0$, while the Hopper-Hop study incorporates a structured–residual transition into TD-MPC2. The DreamerV3 study uses a different integration interface: its learned world model is retained, and structural knowledge instead guides action selection during training-time imagination.

The first three studies provide the most direct parameter-only instantiations of the structured transition construction and are therefore closest in model form to the realizable structured setting analyzed in Section 5. Hopper-Hop exercises the broader structured–residual case in which learned dynamics supplement a mechanistic backbone. DreamerV3 illustrates a complementary use of supplied structure through an auxiliary integration interface without replacing the host learned world model. Common evaluation conventions are summarized in Appendix C, while task-specific model definitions, optimization settings, and planning configurations are provided in Appendices D, E, F, and G.

Table 2: Relationship between the general structured world-model construction and the experimental studies.

Study	Supplied structure	Learned or retained component	Structural interface	Decision procedure
Epidemic control (A2)	SEIHR compartment structure and clinical transition mechanisms	Six-dimensional physical and intervention-related parameter vector	Parameter-only structured transition, $F_G(\cdot;\theta)$ with $R_\phi \equiv 0$	Finite-budget MCTS
Graph Beer Game (SD)	Graph routing, inventory–backlog stock–flow relations, shipment and order delays, capacities, and action structure	Three shared demand-profile parameters	Parameter-only structured transition, $F_G(\cdot;\theta)$ with $R_\phi \equiv 0$	MCTS in the primary comparison; supplementary MPC
Cartpole-Swingup	Damped Newton–Euler dynamics with supplied gravity, actuator, and damping terms	Cart mass, pole mass, and pole length	Parameter-only structured transition, $F_G(\cdot;\theta)$ with $R_\phi \equiv 0$	Finite-budget MCTS
Hopper-Hop / TD-MPC2	Approximate mechanistic state and transition structure	17-dimensional structured component together with 495-dimensional learned residual latent dynamics	Structured–residual transition integrated into the TD-MPC2 transition representation	Retained TD-MPC2 MPPI planner
DreamerV3	Graph Beer Game structural knowledge used to construct a structural action guide	Original DreamerV3 recurrent world model and actor–critic learning are retained	Structural guidance of imagined actions; the host learned transition model is not replaced	DreamerV3 actor and RSSM imagination

B Supplementary Theoretical Analysis

This appendix provides the detailed assumptions and derivations underlying the theoretical results in Section 5. The analysis focuses on the realizable structured setting in which the systematic dynamics are contained within the prescribed structural family. We first state the complete theoretical setting and then derive the chain from finite-sample parameter estimation to multi-step rollout error, planning-value error, and action-selection stability. A brief discussion of residual model mismatch is provided at the end.

B.1 Theoretical Setting and Assumptions

The general framework introduced in Sections 3 and 4 allows the true transition to contain systematic dynamics that are not fully represented by the supplied structural model:

$$s_{t+1} = \mathcal{P}(s_t, a_t; \theta^\star) + \Delta^\star(s_t, a_t) + \xi_t, \tag{14}$$

where $\mathcal{P}$ denotes the prescribed structural transition family, $\theta^\star \in \mathbb{R}^d$ is an unknown physical parameter vector, $\Delta^\star$ denotes systematic model mismatch, and ξ_t denotes stochastic transition noise. Exogenous inputs are omitted here for notational simplicity; the analysis may be understood conditionally on a fixed exogenous-input sequence.

The main theoretical results consider the realizable special case

$$\Delta^\star \equiv 0, \qquad s_{t+1} = \mathcal{P}(s_t, a_t; \theta^\star) + \xi_t. \tag{15}$$

Thus, the systematic component of the dynamics belongs to the supplied structural family and learning reduces to estimating the low-dimensional parameter vector $\theta^\star$.

For the rollout analysis, we write the conditional-mean transition as

$$\mathcal{T}^\star(s, a) = \mathcal{P}(s, a; \theta^\star), \qquad \widehat{\mathcal{T}}(s, a) = \mathcal{P}(s, a; \hat{\theta}). \tag{16}$$

Accordingly, the rollout bounds below characterize model-induced prediction error in the systematic transition component; irreducible realization noise is accounted for in the finite-sample estimation problem but is not treated as a deterministic model bias.

We use the following assumptions.

T1. Transition observations and noise. The identification data contain N transition observations $\{(s_i, a_i, s_i')\}_{i=1}^N$ satisfying

$$s_i' = \mathcal{P}(s_i, a_i; \theta^\star) + \xi_i, \tag{17}$$

where the observations are independently sampled from the identification distribution and ξ_i is conditionally zero-mean and σ-sub-Gaussian. This independent-sampling assumption corresponds to the generative-simulator setting used for the finite-sample bound.

T2. Local identifiability. The structural parameterization is locally identifiable around $\theta^\star$. In particular, the information matrix associated with the local sensitivity of $\mathcal{P}$ satisfies

$$\lambda_{\min}(\mathcal{I}(\theta^\star)) \geq \mu > 0. \tag{18}$$

The constant μ measures the amount of information available for distinguishing nearby parameter values.

T3. Parameter regularity. The transition map is sufficiently smooth with respect to θ in a neighborhood of $\theta^\star$, and the corresponding local estimation problem satisfies the regularity conditions required for a consistent nonlinear least-squares or equivalent parametric estimator. These conditions determine the problem-dependent constant C_θ appearing below.

T4. Transition Lipschitz continuity. Over the operating region relevant to planning, the structural transition is L_θ-Lipschitz in its parameters and L_s-Lipschitz in the state:

$$\|\mathcal{P}(s, a; \theta_1) - \mathcal{P}(s, a; \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|_2, \tag{19}$$

$$\|\mathcal{T}^\star(s_1, a) - \mathcal{T}^\star(s_2, a)\| \leq L_s \|s_1 - s_2\|. \tag{20}$$

T5. Reward and continuation-value regularity. For the planning-value analysis, the reward is L_r-Lipschitz in the state uniformly over the compared actions, and the continuation value V^π is L_V-Lipschitz:

$$|r(s_1, a) - r(s_2, a)| \leq L_r \|s_1 - s_2\|, \tag{21}$$

$$|V^\pi(s_1) - V^\pi(s_2)| \leq L_V \|s_1 - s_2\|. \tag{22}$$

The terminal estimator V_ψ satisfies

$$\|V_\psi - V^\pi\|_\infty \leq \epsilon_V. \tag{23}$$

T6. Action-gap condition. The decision-stability result is stated for a finite action set. At the state under consideration, the true planning value has a unique maximizing action, so that the corresponding action gap is strictly positive.

Optional feasibility operator. If a task-specific feasibility operator $\Gamma_\mathcal{C}$ is applied to the predicted transition, the rollout bounds below remain unchanged whenever $\Gamma_\mathcal{C}$ is non-expansive:

$$\|\Gamma_\mathcal{C}(x) - \Gamma_\mathcal{C}(y)\| \leq \|x - y\|. \tag{24}$$

This condition ensures that feasibility enforcement does not enlarge an existing prediction discrepancy. It is not required when feasibility is already preserved by construction.

B.2 Finite-Sample Structured Identification

We first formalize the statistical consequence of restricting the unknown dynamics to the low-dimensional structural parameter vector $\theta^\star \in \mathbb{R}^d$.

Let $\hat{\theta}$ denote a consistent estimator obtained from the N transition observations in Assumption T1.

Proposition 1 (Finite-Sample Structured Identification). Under Assumptions T1–T3, there exists a problem-dependent constant $C_\theta > 0$ such that, with probability at least $1 - \delta$,

$$\|\hat{\theta} - \theta^\star\|_2 \le C_\theta \frac{\sigma}{\sqrt{\mu}} \sqrt{\frac{d + \log(1/\delta)}{N}}. \tag{25}$$

Consequently, a sufficient sample size for achieving $\|\hat{\theta} - \theta^\star\|_2 \le \epsilon_\theta$ with probability at least $1 - \delta$ is

$$N = \mathcal{O}\left(\frac{\sigma^2}{\mu} \frac{d + \log(1/\delta)}{\epsilon_\theta^2}\right). \tag{26}$$

Proof. Under local identifiability, the population estimation objective has positive curvature around $\theta^\star$, with its smallest information direction bounded below by μ. The stochastic score at $\theta^\star$ is an average of independent sub-Gaussian transition-noise terms. Standard concentration for a d-dimensional sub-Gaussian score therefore gives, with probability at least $1 - \delta$,

$$\|\nabla \widehat{\mathcal{L}}(\theta^\star)\|_2 \le C_0 \sigma \sqrt{\mu} \sqrt{\frac{d + \log(1/\delta)}{N}}, \tag{27}$$

for a constant C_0 determined by the local sensitivity of the structural parameterization.

Combining this concentration bound with the local curvature implied by Assumption T2 and the regularity conditions in Assumption T3 yields the usual local parametric estimation bound

$$\|\hat{\theta} - \theta^\star\|_2 \le C_\theta \frac{\sigma}{\sqrt{\mu}} \sqrt{\frac{d + \log(1/\delta)}{N}}, \tag{28}$$

where C_θ absorbs constants arising from the local geometry of the parameterization. Rearranging the inequality to require the right-hand side to be at most ϵ_θ gives Eq. (26). □

The central quantity in this result is the effective structural dimension d. When the supplied structural family is valid, the estimator searches over this parameter space rather than an unrestricted representation of the full transition map. The proposition therefore formalizes the dimensional reduction provided by structured identification; it does not require a separate transition model to be learned independently for every state–action pair.

B.3 From Parameter Error to Multi-Step Rollout Error

We next propagate the parameter-estimation error into the trajectories used by model-based planning.

Define the one-step systematic model error by

$$\eta_N := \sup_{s,a} \left\| \widehat{\mathcal{T}}(s,a) - \mathcal{T}^\star(s,a) \right\|. \tag{29}$$

By the parameter Lipschitz condition in Eq. (19),

$$\eta_N \le L_\theta \|\hat{\theta} - \theta^\star\|_2. \tag{30}$$

Combining this relation with Proposition 1 directly transfers the finite-sample parameter-estimation rate to one-step transition accuracy.

Let $\{s_{t+k}\}$ denote a rollout under the true systematic transition and $\{\hat{s}_{t+k}\}$ the corresponding rollout under the learned transition, starting from possibly different initial states but following the same candidate action sequence. Define

$$e_k := \mathbb{E}\left[\|\hat{s}_{t+k} - s_{t+k}\|\right]. \tag{31}$$

Proposition 2 (Multi-Step Rollout Error). Under Assumption T4,

$$e_{k+1} \le L_s e_k + \eta_N, \tag{32}$$

and therefore

$$e_H \le L_s^H e_0 + \eta_N \sum_{j=0}^{H-1} L_s^j. \tag{33}$$

Equivalently,

$$e_H \le \begin{cases} L_s^H e_0 + \dfrac{1 - L_s^H}{1 - L_s}\eta_N, & 0 \le L_s < 1, \\ e_0 + H\eta_N, & L_s = 1, \\ L_s^H e_0 + \dfrac{L_s^H - 1}{L_s - 1}\eta_N, & L_s > 1. \end{cases} \tag{34}$$

Proof. For a common action a_{t+k} at rollout step k,

$$\begin{aligned} \|\hat{s}_{t+k+1} - s_{t+k+1}\| &= \left\|\widehat{\mathcal{T}}(\hat{s}_{t+k}, a_{t+k}) - \mathcal{T}^\star(s_{t+k}, a_{t+k})\right\| && (35) \\ &\le \left\|\widehat{\mathcal{T}}(\hat{s}_{t+k}, a_{t+k}) - \mathcal{T}^\star(\hat{s}_{t+k}, a_{t+k})\right\| \\ &\quad + \|\mathcal{T}^\star(\hat{s}_{t+k}, a_{t+k}) - \mathcal{T}^\star(s_{t+k}, a_{t+k})\| && (36) \\ &\le \eta_N + L_s\|\hat{s}_{t+k} - s_{t+k}\|. && (37) \end{aligned}$$

Taking expectations yields Eq. (32). Iterating the recursion gives Eq. (33), and evaluating the geometric sum gives the three cases in Eq. (34). □

The result separates two factors that determine long-horizon prediction quality. Structural parameterization affects the magnitude of the one-step estimation error η_N, whereas the propagation of that error is governed by the dynamical sensitivity L_s. In particular, structural knowledge does not by itself imply contractive dynamics. When $L_s < 1$, the accumulated effect of one-step error remains bounded; when $L_s = 1$, it grows at most linearly; and when $L_s > 1$, geometric amplification is possible.

If a non-expansive feasibility operator satisfying Eq. (24) is applied to both transitions, the same argument continues to hold because the operator cannot increase the discrepancy between the two predicted successor states.

B.4 From Rollout Error to Planning-Value Error

We next connect rollout-state error to the values assigned to candidate actions.

Consider an H-step bootstrapped planning value under a fixed rollout policy π:

$$Q_H^\pi(s, a) = \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^k r(s_k, a_k) + \gamma^H V^\pi(s_H)\right]. \tag{38}$$

Let $\hat{Q}_H^\pi(s, a)$ denote the corresponding quantity computed using the learned transition model and the terminal estimator V_ψ:

$$\hat{Q}_H^\pi(s, a) = \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^k r(\hat{s}_k, a_k) + \gamma^H V_\psi(\hat{s}_H)\right]. \tag{39}$$

The comparison is made under a common candidate action sequence; equivalently, the rollout action draws are coupled across the true and learned transitions for a fixed, model-independent rollout policy.

Proposition 3 (Planning-Value Error). Under Assumption T5,

$$\left|\hat{Q}_H^\pi(s, a) - Q_H^\pi(s, a)\right| \le L_r \sum_{k=0}^{H-1} \gamma^k e_k + \gamma^H \left(L_V e_H + \epsilon_V\right). \tag{40}$$

Proof. By the triangle inequality,

$$\begin{aligned}&\left|\hat{Q}_H^\pi(s,a) - Q_H^\pi(s,a)\right| \\ &\leq \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^k \left|r(\hat{s}_k, a_k) - r(s_k, a_k)\right|\right] \\ &\quad + \gamma^H \mathbb{E}\left[|V_\psi(\hat{s}_H) - V^\pi(s_H)|\right]. \end{aligned} \tag{41}$$

The reward Lipschitz condition gives

$$\mathbb{E}\left[|r(\hat{s}_k, a_k) - r(s_k, a_k)|\right] \leq L_r e_k. \tag{42}$$

For the terminal term, add and subtract $V^\pi(\hat{s}_H)$:

$$\begin{aligned}&|V_\psi(\hat{s}_H) - V^\pi(s_H)| \\ &\leq |V_\psi(\hat{s}_H) - V^\pi(\hat{s}_H)| + |V^\pi(\hat{s}_H) - V^\pi(s_H)| &(43)\\ &\leq \epsilon_V + L_V \|\hat{s}_H - s_H\|. &(44)\end{aligned}$$

Taking expectations and substituting the resulting inequalities into Eq. (41) proves Eq. (40). □

The bound makes explicit three sources of planning-value error:

1. intermediate rollout-state errors contribute reward distortion through $L_r \sum_{k=0}^{H-1} \gamma^k e_k$;
2. terminal-state prediction error contributes $\gamma^H L_V e_H$;
3. approximation error in the terminal continuation-value estimator contributes $\gamma^H \epsilon_V$.

Thus, the terminal estimator addresses the continuation beyond the explicit planning horizon, while the accuracy of the structured world model controls the model-induced components of the value error.

B.5 Decision Stability under an Action Gap

We finally relate planning-value accuracy to the action selected by the planner.

For this result, write

$$Q = Q_H^\pi, \qquad \hat{Q} = \hat{Q}_H^\pi. \tag{45}$$

Let

$$a^\star = \arg\max_a Q(s,a) \tag{46}$$

be the unique maximizing action, and define the action gap

$$\Delta_Q(s) = Q(s,a^\star) - \max_{a \neq a^\star} Q(s,a) > 0. \tag{47}$$

Corollary 1 (Decision Stability). If

$$\sup_a \left|\hat{Q}(s,a) - Q(s,a)\right| < \frac{\Delta_Q(s)}{2}, \tag{48}$$

then

$$\arg\max_a \hat{Q}(s,a) = a^\star. \tag{49}$$

Proof. Let

$$\epsilon_Q = \sup_a |\hat{Q}(s,a) - Q(s,a)|. \tag{50}$$

For the true maximizing action,

$$\hat{Q}(s,a^\star) \geq Q(s,a^\star) - \epsilon_Q. \tag{51}$$

For any $a \neq a^\star$,

$$\begin{aligned}\hat{Q}(s,a) &\leq Q(s,a) + \epsilon_Q &(52)\\ &\leq Q(s,a^\star) - \Delta_Q(s) + \epsilon_Q. &(53)\end{aligned}$$

Therefore,

$$\hat{Q}(s, a^\star) - \hat{Q}(s, a) \geq \Delta_Q(s) - 2\epsilon_Q. \tag{54}$$

When $\epsilon_Q < \Delta_Q(s)/2$, the right-hand side is strictly positive for every $a \neq a^\star$, proving that $a^\star$ remains the unique maximizing action under $\hat{Q}$. □

Combining Proposition 1, Proposition 2, Proposition 3, and Corollary 1 yields the complete theoretical chain

$$\boxed{d \Longrightarrow \|\hat{\theta} - \theta^\star\|_2 \Longrightarrow \eta_N \Longrightarrow \varepsilon_H^{\mathrm{roll}} \Longrightarrow \left|\hat{Q}_H^\pi - Q_H^\pi\right| \Longrightarrow \text{action preservation under a sufficient gap.}} \tag{55}$$

The first implication reflects the statistical benefit of low-dimensional structural identification; the middle implications describe how model error propagates through prediction and planning; and the final implication states the condition under which the resulting value accuracy is sufficient to preserve the selected action.

B.6 Discussion Beyond the Realizable Case

The realizable analysis isolates the effect of structural parameterization. The general framework also allows systematic dynamics outside the supplied structural family:

$$s_{t+1} = \mathcal{P}(s_t, a_t; \theta^\star) + \Delta^\star(s_t, a_t) + \xi_t, \tag{56}$$

with a learned residual component R_ϕ intended to approximate $\Delta^\star$.

A useful quantity for describing the remaining systematic mismatch is the uniform residual approximation error

$$\epsilon_R = \sup_{s,a} \|R_\phi(s, a) - \Delta^\star(s, a)\|. \tag{57}$$

For the combined learned transition

$$\widehat{\mathcal{T}}_{\mathrm{hyb}}(s, a) = \mathcal{P}(s, a; \hat{\theta}) + R_\phi(s, a), \tag{58}$$

the corresponding one-step systematic prediction error satisfies

$$\begin{aligned} &\left\| \widehat{\mathcal{T}}_{\mathrm{hyb}}(s, a) - \mathcal{T}^\star_{\mathrm{gen}}(s, a) \right\| \\ &\leq \left\| \mathcal{P}(s, a; \hat{\theta}) - \mathcal{P}(s, a; \theta^\star) \right\| + \|R_\phi(s, a) - \Delta^\star(s, a)\| \qquad (59) \\ &\leq L_\theta \|\hat{\theta} - \theta^\star\|_2 + \epsilon_R. \qquad (60) \end{aligned}$$

Hence, if the combined transition continues to satisfy the required state-regularity conditions, residual approximation error enters the subsequent rollout and value-error propagation as an additional source of one-step model error.

This observation does not by itself provide a finite-sample learning guarantee for an unrestricted residual model. Joint estimation of θ and R_ϕ may introduce additional statistical complexity depending on the residual function class, data distribution, and optimization procedure. Accordingly, the formal finite-sample guarantee in Proposition 1 is stated for the realizable low-dimensional structural setting, while Eq. (60) describes how a bounded residual approximation error would enter the later stages of the error decomposition.

C Experimental Protocols and Statistical Reporting

This section summarizes the common experimental conventions used to interpret the results in Section 6. We distinguish three aspects: the amount of real-environment interaction available to each method, the optimization procedures used to fit learnable model components, and the decision procedure used to query the resulting world model. Task-specific state definitions, structural parameterizations, model variants, and complete results are described in the subsequent appendices.

Unless otherwise stated, protocol values are taken from the result-bearing configuration, archived runner, or manifest associated with the reported results.

Table 3: Interaction budgets and statistical reporting conventions.

Experiment	Interaction budget	Seeds	Evaluation	Statistical reporting
Epidemic control	52, 104, 260, 520, or 1040 weekly decisions; each decision advances seven daily transitions	10 seeds	20 evaluation episodes per seed	Mean and sample standard deviation across seeds
Graph Beer Game	1k, 3k, or 10k environment-interaction periods	5 seeds	10 evaluation episodes per seed	Mean and sample standard deviation across seeds
DreamerV3 integration	1k, 3k, 10k, 30k, or 100k environment interactions	5 paired seeds	10 greedy evaluation episodes per seed	Across-seed summaries and paired 95% percentile-bootstrap confidence intervals
Hopper-Hop	Reported checkpoints at 50k, 100k, and 200k environment steps	Seeds 1, 2, and 3	10 evaluation episodes per checkpoint	Mean and population standard deviation across seeds
Cartpole-Swingup	35k shared development interactions plus 5k interactions per target seed	7 development seeds and 3 target seeds	5 evaluation episodes per target seed	Per-seed mean and episode-level standard deviation; overall mean across target seeds

C.1 Evaluation Budgets and Statistical Reporting

Table 3 summarizes the interaction budgets, random seeds, and evaluation conventions used across the experimental suites. The reported interaction budgets count real-environment interactions under the corresponding task protocol. They should therefore be interpreted as empirical data budgets rather than as a one-to-one instantiation of the independent-sample quantity N used in the proposition-specific analysis of Section 5.

For the DreamerV3 integration study, paired differences are first formed between the structure-guided and original DreamerV3 variants at the same random seed. Each seed-level return is the mean over ten greedy evaluation episodes generated with matched evaluation seeds. The reported 95% confidence intervals are percentile-bootstrap intervals obtained from 10,000 resamples of the five seed-level paired differences.

For Graph Beer Game comparisons, the paired 95% percentile-bootstrap intervals for the twelve SD-MCTS versus BB-MCTS topology–budget cells are computed from the five seed-level return differences using 10,000 resamples.

The Cartpole-Swingup experiment is used only as supplementary continuous-control and planning-ablation evidence. Its protocol is drawn from the archived experimental runner and summary files associated with the reported supplementary results.

C.2 Model Learning and Optimization

Table 4 reports the main optimization settings required to interpret the learnable components of each experiment. The table focuses on the model-learning procedures relevant to the claims in the main text; task-specific model definitions, auxiliary objectives, and experimental-arm details are given in the corresponding task appendices.

These settings reflect the different roles of learning across the experiments. The epidemic and Beer Game studies primarily test low-dimensional structured parameter identification against higher-dimensional black-box transition learning. Hopper-Hop instead instantiates the more general hybrid

Table 4: Main optimization settings for learnable world-model and value components.

Experiment	Learnable component	Optimization	Update protocol
Epidemic control	Black-box transition model; structured physical parameters; value estimator	Black-box: Adam, 3×10^{-3}. Structured estimator: Adam, 5×10^{-2}. Value estimator: Adam, 10^{-3}	Black-box and structured estimators are updated for 12 optimization steps every 8 collected transitions; batches contain up to 32 and 64 transitions, respectively. The value estimator is updated every 16 transitions from recent trajectory data.
Graph Beer Game	Three-parameter structured demand model or black-box state-increment transition model; value estimator	Structured model: Adam, 10^{-2}. Black-box model: Adam, 10^{-3}. Value estimator: Adam, 10^{-3}	Batch size 32, replay capacity 5000, and five model updates per episode. Training begins with a budget-dependent random-interaction warmup.
DreamerV3 integration	Original DreamerV3 world-model and actor–critic components	Optax-based optimization, learning rate 4×10^{-5}	Batch configuration 8×32, training ratio 512, and replay capacity 100,000. The structural guide changes imagined action selection but does not introduce a separate transition-model optimizer.
Hopper-Hop	Structured–residual transition representation and retained TD-MPC2 learning components	Adam, learning rate 3×10^{-4}	Batch size 128. The structured transition core additionally receives online supervised dynamics updates; task-specific auxiliary losses are detailed with the Hopper integration results.
Cartpole-Swingup	Three physical dynamics parameters and a frontier value estimator	System identification: Adam, 5×10^{-3}. Value estimator: AdamW, 10^{-3}	System identification uses 800 epochs with a 10-step multi-step objective. The value estimator uses batch size 256 and 500 training epochs; a shared value prior is learned on development seeds and then fine-tuned on each target seed.

setting in which a mechanistic transition backbone is combined with learned residual dynamics. Cartpole-Swingup provides an additional parameter-only structured case.

C.3 Planning and Decision Configuration

The world model and the downstream decision procedure are configured separately. Table 5 summarizes the main decision procedures used in the reported experiments. Within each core world-model comparison, structured and black-box models are evaluated under the same task-specific decision procedure.

Supplementary Beer Game comparisons using alternative planners are described together with the model–planner factorial results, where the relevant planner configuration is reported alongside each comparison. Likewise, the Cartpole one-step value-lookahead variant is introduced with the corresponding supplementary planning ablation rather than as part of the common protocol.

D Epidemic-Control Experiments

This section provides the complete experimental specification and supplementary results for the epidemic-control study summarized in Section 6.2. The experiment instantiates the parameter-identification setting of Section 4.2: the compartmental structure is supplied to the model, while a small set of physical parameters is estimated from interaction data. We first describe the formal experimental configuration and the model variants, and then report the complete finite-budget results and parameter-recovery diagnostics.

Table 5: Planning and decision configurations used in the experimental suites.

Experiment	Decision procedure	Main configuration	Role of the learned model
Epidemic control	Finite-budget MCTS	64 simulations per decision, $\gamma = 0.99$, and rollout-based frontier evaluation with depth 4	Candidate interventions are evaluated by recursively querying the learned epidemic transition model.
Graph Beer Game	Sequential per-node MCTS	100 simulations per node, $\gamma = 0.99$, with $4n$ sequential rollout substeps for an n-node graph, corresponding to four joint decision periods	Structured and black-box transition models are queried under the same MCTS configuration within each topology.
DreamerV3 integration	DreamerV3 actor and RSSM imagination	Imagination length 15; no external MCTS planner	The learned DreamerV3 transition model is retained. Structural information enters only through training-time guidance of imagined actions; environment collection and final evaluation remain actor-driven.
Hopper-Hop	TD-MPC2 MPPI planner	256 sampled trajectories, 32 elites, 24 policy trajectories, horizon 3, six planning iterations, and action repeat 2	The host planner queries the structured–residual transition representation while retaining the original reward, value, and policy-learning components.
Cartpole-Swingup	Finite-budget MCTS with frontier value estimation	200 simulations per decision, 11 discretized action branches, UCB coefficient 1.5, $\gamma = 0.99$, and action repeat 2	The parameter-only structured model supplies imagined transitions and the value network evaluates search frontiers. Tree depth is determined by expansion under the simulation budget rather than by a separately fixed numeric depth.

D.1 Formal Experimental Setting and Structured Parameterization

The formal epidemic-control experiments use a deterministic SEIHR compartmental system with population

$$N = 1{,}000{,}000.$$

The initial compartment counts are

$$(S_0, E_0, I_0, H_0, R_0) = (999{,}900, 0, 100, 0, 0).$$

A decision is made once per week, and each weekly decision advances the underlying epidemic dynamics for seven daily transitions. Each episode contains 52 weekly decisions.

The supplied structural model retains the SEIHR stock–flow organization, with susceptible, exposed, infectious, hospitalized, and recovered populations as the principal physical compartments. Intervention actions modulate the transmission process while the remaining compartmental relations evolve according to the supplied epidemic structure. The formal configuration uses the physical values summarized in Table 6.

The weekly objective penalizes epidemic burden, hospital-capacity exceedance, and intervention cost. The formal configuration assigns weight 10 to the infection component, unit weight to the hospital-overflow component, and the action-dependent intervention costs listed in Table 6.

The primary structured model, A2, retains the SEIHR structure and estimates a six-dimensional parameter vector recorded in the formal result artifacts,

$$\theta_{\mathrm{A2}} = (\beta_0, h, \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)\,,$$

Table 6: Formal epidemic-control configuration used for the reported results.

Quantity	Value
Population N	1,000,000
Initial infectious population	100
Hospital capacity	1,000
External susceptible exposure per day	3,000
Baseline transmission parameter β_0	0.5
Exposed-to-infectious rate σ	0.25
Infectious-outflow rate γ	0.1667
Hospital recovery rate ρ	0.1
Hospitalization fraction h	0.03
Intervention efficacies	[0, 0.2, 0.4, 0.6, 0.75]
Intervention costs	[0, 0.05, 0.12, 0.25, 0.4]
Weekly decision horizon	52
Daily transitions per decision	7
Planning discount	0.99

where the no-intervention efficacy $\epsilon_0 = 0$ is fixed. The extended variant A2-full-θ additionally estimates the three clinical transition parameters,

$$\theta_{\text{A2-full}} = (\beta_0, h, \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \sigma, \gamma, \rho)\,,$$

for an effective dimension of nine. These variants correspond to the parameter-only structured setting $R_\phi \equiv 0$ in Section 4.2.

D.2 Experimental Arms

The formal experiment evaluates eight model and identification variants. Table 7 summarizes the distinction among them. The main-text comparison uses A1', A2, and A4 to contrast black-box learning, low-dimensional structured identification, and a true-parameter structured reference under the same finite-budget planning setting. The remaining arms probe the effect of model class, parameterization, and identification procedure.

All reported evaluation results use frozen model parameters during evaluation. The common evaluation and planning conventions are given in Appendix C.

D.3 Complete Finite-Data Results

Table 8 reports the complete nominal-budget results for all eight arms. Each entry is the mean return and sample standard deviation across ten random seeds.

The complete matrix reinforces the finite-data pattern highlighted in Section 6.2. At the smallest budget of 52 weekly decisions, A2 reaches -37.327 ± 0.244, close to the true-parameter A4 reference at -37.418 ± 0.174, while the context-augmented black-box model A1' obtains -47.840 ± 4.455. The structured variants remain stable across the evaluated budgets, whereas the black-box models improve more gradually as additional interaction data become available.

The supplementary arms further show that the strong performance of the structured representation is not tied to estimating a single prescribed set of physical parameters. Expanding the learned parameter vector in A2-full-θ produces similar returns to A2, while the passive identification variants A5-fixed and A5-receding also remain close to the true-parameter structured references. Across these variants, retaining the SEIHR structure while learning only its uncertain numerical quantities consistently yields strong finite-data decision performance.

D.4 Fitted-Parameter Diagnostics

The structured runs also record the fitted physical parameters. To complement the decision-level results, Table 9 reports parameter recovery for A2 at the smallest interaction budget. Each estimate is aggregated over the same ten random seeds used for the return evaluation.

Table 7: Epidemic-control experimental arms.

Arm	Model family	Main distinction	Experimental role
A1	Black-box transition model	Learns the transition directly from state–action data without the supplied SEIHR parameterization	Unstructured transition-learning baseline
A1'	Context-augmented black-box model	Extends the black-box input with intervention-context variables	Stronger black-box comparator used in the main text
A2	Structured SEIHR model	Estimates a six-dimensional physical parameter vector while retaining the supplied compartmental structure	Primary structured-identification model
A2-full-θ	Structured SEIHR model	Extends A2 by also estimating σ, γ, and ρ	Parameter-dimension ablation
A3	Fixed structured model	Uses fixed true physical parameters under the standard warmup/data-collection protocol	Fixed true-parameter structured reference
A4	True-parameter structured model	Uses the true structured parameters directly for planning without the parameter-learning warmup stage	Main true-model planning reference
A5-fixed	Structured SEIHR model	Estimates the structured parameters from a passive identification stage and then keeps the fitted model fixed	Passive-identification comparison
A5-receding	Structured SEIHR model	Begins from passive identification and subsequently updates the fitted structured model with newly collected data	Receding-identification comparison

Table 8: Complete epidemic-control results across interaction budgets. Entries are mean return $\pm$ sample standard deviation across ten seeds. Higher return is better.

Method	52	104	260	520	1040
A1	-51.678 ± 5.326	-49.839 ± 6.784	-47.874 ± 5.571	-43.103 ± 4.711	-44.968 ± 4.818
A1'	-47.840 ± 4.455	-48.352 ± 5.655	-46.221 ± 3.934	-47.680 ± 5.484	-42.632 ± 5.281
A2	-37.327 ± 0.244	-37.279 ± 0.158	-37.445 ± 0.169	-37.472 ± 0.193	-37.476 ± 0.221
A2-full-θ	-37.373 ± 0.162	-37.328 ± 0.147	-37.483 ± 0.165	-37.444 ± 0.180	-37.383 ± 0.224
A3	-37.503 ± 0.158	-37.405 ± 0.168	-37.348 ± 0.177	-37.376 ± 0.217	-37.381 ± 0.141
A4	-37.418 ± 0.174	-37.404 ± 0.183	-37.362 ± 0.186	-37.389 ± 0.194	-37.426 ± 0.167
A5-fixed	-37.418 ± 0.263	-37.388 ± 0.211	-37.355 ± 0.105	-37.478 ± 0.136	-37.464 ± 0.213
A5-receding	-37.536 ± 0.137	-37.505 ± 0.078	-37.407 ± 0.206	-37.402 ± 0.231	-37.480 ± 0.212

Even at the smallest evaluated budget, the fitted parameters remain close to their configured values. This parameter-level behavior is consistent with the decision results in Table 8: the learned structured model already approaches the true-parameter structured reference under limited interaction data.

At the largest budget, A2 continues to yield fitted values close to the configured physical quantities, with small aggregate errors. The extended A2-full-θ model likewise yields the following fitted values for the additional clinical parameters at 1040 interactions:

$$\sigma = 0.2516 \pm 0.0033, \qquad \gamma = 0.1666 \pm 0.0014, \qquad \rho = 0.0951 \pm 0.0035,$$

compared with configured values 0.25, 0.1667, and 0.10, respectively.

Table 9: Fitted parameter values for A2 at 52 weekly decisions. Estimates are mean ± sample standard deviation across ten seeds; MAE denotes mean absolute error relative to the configured value.

Parameter	True value	Estimate	MAE
β_0	0.500	0.4967 ± 0.0075	0.0047
h	0.030	0.0301 ± 0.0012	0.0009
ϵ_1	0.200	0.1962 ± 0.0127	0.0045
ϵ_2	0.400	0.3975 ± 0.0113	0.0052
ϵ_3	0.600	0.5986 ± 0.0089	0.0048
ϵ_4	0.750	0.7496 ± 0.0074	0.0049

The recorded runs additionally include local Jacobian-based identification diagnostics. These diagnostics indicate strong correlations among some raw transmission-related parameterizations, while the effective physical parameters reported above remain close to their configured values. We use these diagnostics as supplementary evidence about the local estimation geometry; the empirical conclusions in this section are based primarily on the recovered physical parameters and the corresponding decision performance.

E Graph Beer Game Experiments

This section provides the complete Graph Beer Game specification and supplementary results underlying the multi-step prediction analysis in Section 6.2 and the cross-graph evaluation in Section 6.3. The task family is particularly useful for studying structured world models because inventory, backlog, shipment, and order-delay variables must remain part of the decision state in order to support recursive prediction. We first describe the graph-based environment and the structured and black-box transition models, and then report the multi-step fidelity diagnostics, complete topology-by-budget results, and complementary model–planner comparisons.

E.1 Environment and Graph Topologies

The Graph Beer Game represents a supply network as a directed graph. The decision state contains node inventories, retail and edge backlogs, shipment pipelines, order pipelines, source pipelines, the current time, and the current retail-demand variables. Delay and pipeline states are retained explicitly so that the predicted successor state contains the memory required for subsequent model calls.

For a graph with n decision nodes, each node selects one of nine discrete order levels. Let D_v denote the design-demand scale associated with node v. Action index $k \in \{0, \dots, 8\}$ corresponds to an order quantity

$$q_v(k) = 2D_v \frac{k}{8}. \tag{61}$$

The episode horizon is 36 periods. Shipment and order delays are both two periods. At each retail node, external demand is 4 units per period for $t < 5$ and 8 units per period thereafter.

The per-period cost combines inventory holding and backlog penalties,

$$C_t = 0.5 \sum_{v \in \mathcal{V}} \max(I_{v,t}, 0) + \sum_{v \in \mathcal{R}} B_{v,t} + \sum_{e \in \mathcal{E}} B_{e,t}, \tag{62}$$

where $\mathcal{V}$, $\mathcal{R}$, and $\mathcal{E}$ denote the node, retail, and edge sets, respectively. The reward is $r_t = -C_t$.

Across topologies, the same stock–flow and delay mechanisms are instantiated on different directed graphs. Table 10 summarizes the four graph structures used in the experiments.

The resulting state dimensions are 24, 48, 48, and 80 for T1–T4, respectively. All edges use shipment and order delays of two periods, unit backlog cost, and all nodes use a holding-cost coefficient of 0.5.

The design-demand scale D_v also determines the local action range and initial operating scale: the maximum order quantity is $2D_v$, initial inventory is $1.5D_v$, and source capacity is $2.5D_v$. The

Table 10: Graph Beer Game topologies used in the experiments.

Topology	Nodes / edges	Source nodes	Retail nodes	Directed structure
T1	$4/3$	Factory	Retailer	Factory → Distributor → Wholesaler → Retailer
T2	$8/7$	n7	n0	n7 → n6 → n5 → n4 → n3 → n2 → n1 → n0
T3	$7/6$	src	r0–r3	src → w0,w1; w0 → r0,r1; w1 → r2,r3
T4	$10/11$	s0, s1	r0–r4	s0 → w0,w1; s1 → w1,w2; w0 → r0,r1; w1 → r1,r2,r3; w2 → r3,r4

Table 11: Topology-specific demand and capacity scales. Non-source nodes have no additional source-capacity limit.

Topology	Design-demand and initialization scales
T1	All nodes: $D_v = 8$, $O_{\max} = 16$, $I_0 = 12$; Factory source capacity 20
T2	All nodes: $D_v = 8$, $O_{\max} = 16$, $I_0 = 12$; n7 source capacity 20
T3	src: $D_v = 32$, $(O_{\max}, I_0, \text{cap}) = (64, 48, 80)$; w0,w1: $D_v = 16$, $(32, 24)$; r0–r3: $D_v = 8$, $(16, 12)$
T4	s0,s1: $D_v = 20$, $(40, 30, 50)$; w0,w2: $D_v = 12$, $(24, 18)$; w1: $D_v = 16$, $(32, 24)$; r0–r4: $D_v = 8$, $(16, 12)$

topology-specific values are listed in Table 11. Pipelines are initialized at half of their corresponding nominal flows.

Because the topologies differ not only in graph connectivity but also in their demand and capacity scales, the experimental comparisons below focus on methods evaluated within the same topology.

E.2 Structured and Black-Box World Models

The structured world model preserves the supplied graph routing, inventory–backlog stock–flow relations, shipment and order delays, capacity information, cost structure, and discrete order grid. The shared structured variant learns only three quantities describing the demand profile,

$$\theta_{\text{SD}} = (d_{\text{low}}, d_{\text{high}}, t_{\text{step}}), \tag{63}$$

and uses the same three parameters across retail nodes. No learned residual transition is added to this model. Non-negativity is enforced within the structured transition.

The black-box model instead learns the state transition from the current state and a concatenated one-hot representation of the node actions. On T4, the input dimension is

$$80 + 10 \times 9 = 170,$$

and the transition network is

$$170 \to 256 \to 256 \to 80,$$

with ReLU nonlinearities between hidden layers. The network predicts a state increment that is added to the current state, resulting in 130,128 trainable parameters on T4. Negative imagined physical states are retained rather than repaired, allowing the same predictions to be used for the structural-consistency diagnostic below.

Table 12 summarizes the model variants used in the supplementary comparisons.

The contrast between the three-parameter structured model and the 130,128-parameter T4 black-box model illustrates the difference between estimating a low-dimensional unknown component within a supplied transition structure and learning a substantially less constrained transition representation.

E.3 Multi-Step Rollout and Structural Diagnostics

Section 6.2 reports the T4 multi-step prediction comparison at a training budget of 1,000 environment interactions. Here we provide the complete evaluation protocol and horizon-wise results.

Table 12: World-model variants in the Graph Beer Game experiments.

Variant	Learned representation	Supplied structure	Role
SD	Three shared demand-profile parameters	Graph topology, routing, delays, stock–flow update, capacities, costs, and action grid	Primary structured model
BB	Neural state-increment predictor	State and action representation, without the structured transition	Black-box comparison
SD-oracle	No learned world-model parameters	Same structured model with the true demand profile	True-parameter structured reference
SD-node	One $(d_{\mathrm{low}}, d_{\mathrm{high}}, t_{\mathrm{step}})$ triple per retail node	Same structured transition as SD	Shared-versus-node-specific parameterization comparison

Table 13: T4 multi-step rollout RMSE at 1,000 environment interactions. Entries are mean $\pm$ sample standard deviation across five seeds.

Horizon H	Structured model	Black-box model
1	$1.38 \times 10^{-7} \pm 7.71 \times 10^{-9}$	4.445 ± 0.121
2	0.112 ± 0.114	6.200 ± 0.143
3	0.211 ± 0.217	7.459 ± 0.232
4	0.297 ± 0.314	8.659 ± 0.340
5	0.375 ± 0.411	9.831 ± 0.446
6	0.446 ± 0.512	11.030 ± 0.558
7	0.521 ± 0.616	12.240 ± 0.674
8	0.593 ± 0.723	13.470 ± 0.784

For each evaluated trajectory, the learned model is initialized from a recorded environment state and recursively propagated using the action sequence recorded for that same method. Intermediate model states are not replaced by ground-truth environment states. Prediction error is evaluated over horizons $H = 1, \ldots, 8$ using 74 physical state coordinates: inventories, retail and edge backlogs, shipment pipelines, order pipelines, and source pipelines. Time and retail-demand slots are excluded, and errors are computed in the original state units.

RMSE is first computed over each rollout trajectory, then averaged within each seed, and finally aggregated across the five seeds. The complete results are reported in Table 13.

The structured model maintains substantially lower prediction error throughout the evaluated rollout horizon. At $H = 2$, its RMSE is 0.112 compared with 6.200 for the black-box model; at $H = 8$, the corresponding values are 0.593 and 13.470. Under the same T4 setting and interaction budget, SD-MCTS obtains a return of -3404.5 ± 183.1, compared with -26748.6 ± 3402.2 for BB-MCTS. The combination of lower recursive prediction error and stronger decision performance is consistent with the error- propagation pathway analyzed in Section 5.

We additionally evaluate non-negativity during model-based search. A predicted state is counted as violating this diagnostic when at least one selected physical coordinate is below -10^{-6}. The diagnostic is evaluated over states imagined during MCTS expansion and leaf rollouts and is reported per imagined state rather than per coordinate. Each seed contains approximately 14.76 million imagined states under the T4 evaluation.

The structured transition therefore remains close to the supplied non-negativity structure during search, whereas negative imagined physical states are frequent under the black-box transition. Together with the rollout results, this diagnostic shows that the structured model preserves both substantially better multi-step predictive fidelity and the tested basic structural condition in the imagined trajectories used for planning.

Table 14: Imagined-state non-negativity diagnostic on T4 at 1,000 interactions. Entries are mean ± sample standard deviation across five seeds.

World model	Violation rate
Structured	0.000452 ± 0.000350
Black-box	0.939603 ± 0.000342

Table 15: Complete Graph Beer Game planning results. Entries are mean return ± sample standard deviation across five seeds. Higher return is better.

Topology / budget	SD-MCTS	BB-MCTS
T1 / 1k	-576.4 ± 45.8	-6877.6 ± 1038.3
T1 / 3k	-549.6 ± 23.3	-2929.7 ± 1206.6
T1 / 10k	-572.3 ± 34.7	-666.1 ± 48.6
T2 / 1k	-948.2 ± 35.9	-14877.0 ± 1396.0
T2 / 3k	-933.8 ± 17.1	-16726.7 ± 1379.4
T2 / 10k	-934.5 ± 30.6	-1328.5 ± 478.8
T3 / 1k	-2550.9 ± 145.3	-25582.9 ± 3866.9
T3 / 3k	-2601.3 ± 120.1	-20485.1 ± 4316.3
T3 / 10k	-2538.3 ± 81.0	-3909.1 ± 581.6
T4 / 1k	-3404.5 ± 183.1	-26748.6 ± 3402.2
T4 / 3k	-3293.3 ± 101.7	-23827.7 ± 3703.1
T4 / 10k	-3253.6 ± 149.3	-8681.2 ± 2409.3

E.4 Complete Topology-by-Budget Results

Table 15 expands the topology comparison in Section 6.3 to all reported interaction budgets. All entries are computed from five independent seeds.

Across all twelve evaluated topology–budget combinations, SD-MCTS achieves a higher mean return than BB-MCTS. The difference is largest at the lower interaction budgets and narrows as additional data are provided to the black-box model, most visibly on T1 and T2 at 10k interactions. The structured model nevertheless retains the higher mean return throughout the evaluated serial, branching, and multi-source graph settings.

Because the four environments use topology-dependent demand and capacity scales, these results are interpreted through within-topology comparisons rather than direct return comparisons between T1–T4. The full matrix therefore complements Figure 3 by showing that the finite-data advantage observed in the main text is retained across all evaluated graph structures and budgets.

Table 16 places the T4 results next to two standard learning-based agents trained under the same environment, interaction budgets, and ten-episode evaluation protocol with five seeds. PPO is the Stable-Baselines3 implementation with an MLP policy, ten epochs per update, learning rate 3×10^{-4}, and $\gamma = 0.99$; it is evaluated with the deterministic policy. Pure DreamerV3 uses the configuration described in Appendix F.2 without structural guidance, and its 30k and 100k entries coincide with the pure-DreamerV3 column of Table 19. At 10k interactions both agents remain far below the return that SD-MCTS attains with 1k interactions, and DreamerV3 exceeds that level only at 100k interactions.

E.5 Model–Planner Factorial Comparisons

The main experiments use MCTS as a common downstream planner in order to compare structured and black-box world models under the same decision procedure. We additionally evaluate several alternative combinations of world model and planner on T4 to examine how the two components contribute to the final decision performance.

For the MCTS variants, each node decision uses 100 simulations over nine actions with discount $\gamma = 0.99$. Search uses a PUCT coefficient of 1.4, progressive widening with $C = 1$ and $\alpha = 0.5$,

Table 16: T4 Beer Game comparison with model-free and latent model-based agents across interaction budgets. Entries are mean return ± sample standard deviation across five seeds. A dash indicates that the budget was not evaluated for that method.

Method	1k	3k	10k	30k	100k
SD-MCTS	-3404.5 ± 183.1	-3293.3 ± 101.7	-3253.6 ± 149.3	–	–
BB-MCTS	-26748.6 ± 3402.2	-23827.7 ± 3703.1	-8681.2 ± 2409.3	–	–
PPO	-21932.8 ± 4906.3	-17201.1 ± 6953.0	-7536.7 ± 1338.5	–	–
Pure DreamerV3	-27843.9 ± 7364.0	-30846.7 ± 8538.1	-15381.6 ± 7962.2	-7536.5 ± 4892.4	-2937.6 ± 485.1

Table 17: Complementary T4 model–planner comparisons. Entries are mean return ± sample standard deviation across five seeds. A dash indicates that the corresponding budget was not evaluated.

Method	1k	3k	10k
SD-MCTS	-3404.5 ± 183.1	-3293.3 ± 101.7	-3253.6 ± 149.3
BB-MCTS	-26748.6 ± 3402.2	-23827.7 ± 3703.1	-8681.2 ± 2409.3
SD-oracle	-3253.6 ± 149.3	-3253.6 ± 149.3	-3253.6 ± 149.3
SD-MPC	-2983.6 ± 141.6	-2989.2 ± 147.8	-2986.4 ± 139.3
BB-MPC	-20866.8 ± 4521.9	-16336.2 ± 5714.8	-11022.7 ± 1040.3
Oracle-MPC	-2986.4 ± 139.3	-2986.4 ± 139.3	-2986.4 ± 139.3
PPO	-21932.8 ± 4906.3	-17201.1 ± 6953.0	-7536.7 ± 1338.5
BaseStock	-3040.0 ± 0.0	-3040.0 ± 0.0	-3040.0 ± 0.0
SD-node	-3385.4 ± 294.2	-3315.9 ± 74.2	-3253.6 ± 149.3

and rollout-based leaf evaluation. On T4, the rollout depth is 40 sequential node-action substeps, corresponding to four complete decision periods.

The MPC variants use random shooting over a four-period horizon with $\gamma = 0.99$. On T4, 1,250 candidate joint plans are sampled at each decision step. These comparisons therefore evaluate alternative downstream planning configurations applied to the corresponding world-model families.

Several patterns are useful for interpreting the main results. First, replacing the structured transition with the black-box transition reduces performance under both MCTS and MPC at the evaluated budgets. The advantage of structured modeling is therefore not specific to the particular MCTS instantiation used in the main comparison.

Second, holding the world-model family fixed while changing the downstream planner produces different absolute returns. In particular, the structured MPC variants perform strongly on T4, illustrating the separation between the transition representation and the decision solver emphasized in Section 4.3. The reported MCTS–MPC comparisons should be read as comparisons between the complete downstream planning configurations rather than as equal-compute planner benchmarks.

Third, SD-MCTS approaches the true-parameter SD-oracle result as the interaction budget increases, while SD-node provides similar performance to the shared three-parameter structured model. These results support the use of a compact shared structured parameterization for the graph experiments without requiring node-specific demand parameters.

Taken together, the Graph Beer Game experiments connect three levels of evidence. The structured model learns a compact unknown component while retaining the supplied stock–flow and delay structure; this produces lower multi-step prediction error and fewer non-negativity violations during imagined search; and the resulting model supports stronger downstream decision performance across the evaluated graph structures and with more than one planning procedure.

F Integration with Existing Model-Based RL

Section 6.4 examines whether structural knowledge can be incorporated into existing model-based reinforcement learning systems without replacing their core learning and planning machinery. We study two complementary interfaces. In DreamerV3, structural knowledge is used to guide actions

Table 18: Structural integration interfaces used in the two model-based RL case studies.

Host algorithm	Structural entry point	Retained components	Structural modification
DreamerV3	Training-time imagined action selection	RSSM, encoder/decoder, reward model, actor, critic/value, slow-value regularization, λ-return learning, and the original imagination and replay-value objectives	A structure-derived action can replace the actor action inside imagined rollouts according to a decaying probability
TD-MPC2	Transition representation	Learned reward model, Q ensemble and target networks, policy, actor objective, and MPPI planner	A 17-dimensional mechanistic state is combined with a 495-dimensional learned residual latent representation and structured transition corrections

only inside training-time imagined rollouts, while the learned recurrent world model and actor–critic pipeline are retained. In TD-MPC2, the transition representation itself is augmented with an explicit mechanistic state and learned residual dynamics while the reward, value, policy, and MPPI planning components remain in place.

This section describes the two integration mechanisms in detail and reports their complete budget-wise results. Common evaluation and statistical conventions are summarized in Appendix C.

F.1 Structural Integration Interfaces

The two case studies insert structural knowledge at different points in the host algorithm. Table 18 summarizes these interfaces.

The DreamerV3 study therefore tests a relatively loose integration interface: structural knowledge modifies the imagined training experience without replacing the host world model. The TD-MPC2 study tests a deeper transition interface that directly instantiates the structured–residual construction in Section 4.2.

F.2 Structure-Guided DreamerV3

Imagination-time structural guidance. The DreamerV3 experiment is conducted on the T4 Graph Beer Game. The original DreamerV3 recurrent state-space model, decoder, reward model, actor, critic, slow-value regularization, and λ-return learning are retained. Structural knowledge is introduced only during imagined training rollouts.

At each imagined state, the current latent feature is decoded and mapped back to physical units. From the reconstructed inventory, backlog, and pipeline variables, we compute a node-wise inventory position. For node v,

$$\begin{aligned}\mathrm{IP}_v = I_v - B_v^{\mathrm{out}} - B_v^{\mathrm{retail}} + P_v^{\mathrm{ship,in}} + P_v^{\mathrm{order,in}} \\ + B_v^{\mathrm{edge,in}} + P_v^{\mathrm{source}},\end{aligned} \tag{64}$$

where the terms collect the corresponding decoded inventory, backlog, and in-transit quantities.

The structural guide uses the same demand scale D_v as the Graph Beer Game environment and defines the target inventory position

$$S_v = (L + \tau + 1.5)D_v = 5.5D_v, \tag{65}$$

with shipment and order delays $L = \tau = 2$. The desired replenishment quantity is

$$q_v^{\mathrm{need}} = \max\left(S_v - \mathrm{IP}_v, 0\right), \tag{66}$$

and the structural action selects the discrete order level whose quantity is closest to q_v^{need}.

Table 19: Complete T4 results for structure-guided and pure DreamerV3. Returns are mean ± sample standard deviation across five seeds. Paired differences are structure-guided minus pure DreamerV3.

Budget	Structure-guided	Pure DreamerV3	Paired difference	95% bootstrap CI
1k	-40728.3 ± 13327.9	-27843.9 ± 7364.0	-12884.5	$[-26020.2,\ 1181.2]$
3k	-21722.7 ± 5146.6	-30846.7 ± 8538.1	$+9123.9$	$[3465.0,\ 13919.6]$
10k	-11381.6 ± 4628.1	-15381.6 ± 7962.2	$+4000.0$	$[-1618.1,\ 9109.1]$
30k	-4500.8 ± 485.0	-7536.5 ± 4892.4	$+3035.7$	$[691.1,\ 7240.8]$
100k	-3363.0 ± 527.8	-2937.6 ± 485.1	-425.4	$[-863.2,\ 37.4]$

Let a_t^{actor} denote the action sampled from DreamerV3's actor and a_t^{struct} the complete ten-node joint action produced by the structural guide. The action used inside imagination is

$$a_t = \begin{cases} a_t^{\text{struct}}, & z_t = 1, \\ a_t^{\text{actor}}, & z_t = 0, \end{cases} \qquad z_t \sim \text{Bernoulli}(\beta(u)), \tag{67}$$

where one Bernoulli draw selects between the two complete joint action vectors. The guide probability follows

$$\beta(u) = 0.2 \max\left(0,\ 1 - \frac{u}{0.5 U_{\text{exp}}}\right), \tag{68}$$

where u is the current training-update index and U_{exp} is the expected number of updates for the corresponding interaction budget. For budgets of 1k, 3k, 10k, 30k, and 100k interactions, U_{exp} is set to

$$2{,}000,\ 6{,}000,\ 20{,}000,\ 60{,}000,\ 200{,}000,$$

respectively.

The action replacement occurs inside the policy used by DreamerV3's RSSM imagination before the corresponding imagined transition is generated. Real-environment data collection continues to use DreamerV3's actor, and final evaluation uses the greedy actor without structural action replacement. Thus, the structural guide affects the imagined training distribution rather than directly controlling the evaluation policy.

Complete budget-wise results. Table 19 reports the complete comparison between structure-guided and pure DreamerV3. Each seed-level score is averaged over ten greedy evaluation episodes. Reported standard deviations are sample standard deviations across five seeds. Paired differences are computed between the two variants at the same seed, and the corresponding 95% confidence intervals use 10,000 percentile-bootstrap resamples of the five seed-level paired differences.

The structure-guided variant produces higher mean returns at 3k, 10k, and 30k interactions. The paired confidence interval lies entirely above zero at 3k and 30k, while the intervals at 1k, 10k, and 100k include zero. In particular, the 30k result highlighted in Section 6.4 corresponds to a paired improvement of 3035.7 with a 95% confidence interval of $[691.1, 7240.8]$.

These results show that a structure-derived action prior can improve DreamerV3 training at selected finite interaction budgets while leaving the learned RSSM and final actor policy intact. The experiment therefore provides an example of modular structural integration through the imagination process rather than through replacement of the learned transition model.

F.3 Structured–Residual Dynamics in TD-MPC2

Integration boundary. The Hopper-Hop experiment incorporates structural knowledge directly into the TD-MPC2 transition representation. The resulting latent state contains an explicit 17-dimensional structured component and a 495-dimensional learned residual component,

$$z_t = \left[z_t^G, z_t^R\right], \qquad z_t^G \in \mathbb{R}^{17}, \quad z_t^R \in \mathbb{R}^{495}, \tag{69}$$

for a total latent dimension of 512.

Table 20: Component boundary for the structured–residual TD-MPC2 integration.

Component	Status	Integration
Observation representation	Modified	Explicit 17-dimensional structured representation combined with a learned residual encoder
Structured state construction	Added	Mechanistic Hopper state in Eq. (70)
Transition dynamics	Modified	Explicit structured transition with learned residual corrections
Residual latent dynamics	Added	495-dimensional learned residual subspace
Reward model	Retained	Original learned TD-MPC2 reward head
Q ensemble and target networks	Retained	Three Q heads with the TD-MPC2 target-network mechanism
Policy and actor objective	Retained	Original TD-MPC2 policy head and Q-based actor objective
MPPI planner	Retained	Original sampling-based planning mechanism with the formal experiment search budget
Replay and training loop	Modified	Original TD-MPC2 optimization augmented with structured-dynamics supervision and a separate structured-model optimizer

The structured state is

$$z_t^G = [H,\, V_x,\, V_z,\, \Theta,\, \Omega,\, A,\, q_{\text{waist}},\, q_{\text{hip}},\, q_{\text{knee}},\, q_{\text{ankle}},\\ \dot{q}_{\text{waist}},\, \dot{q}_{\text{hip}},\, \dot{q}_{\text{knee}},\, \dot{q}_{\text{ankle}},\, c_{\text{toe}},\, c_{\text{heel}},\, z_{\min}]. \tag{70}$$

The structured transition models mechanistic quantities such as root motion, joint evolution, activation, contact, damping, and gravity-related dynamics. Learned residual components provide additional flexibility beyond this mechanistic backbone. The formal configuration includes residual corrections for (V_x, V_z, Ω) and the two contact variables together with the 495-dimensional residual latent dynamics. The transition residual multiplier is 4.0.

After the structured–residual transition, task-specific state handling bounds physical ranges, wraps angular variables, and constrains activation, contact, rate, and height-related state components. The raw Hopper observation has dimension 15, and one agent action is held for two environment steps, corresponding to a model-step duration of 0.04.

Table 20 summarizes which components of the host algorithm are retained and which are modified by the integration.

The integration therefore changes the state and transition representation while retaining the host algorithm's learned reward, Q, policy, and downstream planning components.

Training and planning configuration. The structured transition core receives a separate supervised transition-data initialization before online TD-MPC2 training. Online learning then combines the original TD-MPC2 objectives with structured-state and residual-latent consistency losses and additional supervision of the mechanistic transition.

The enabled structured auxiliary terms use weights

$$\lambda_{\text{struct}} = 1, \qquad \lambda_{\text{res}} = 1,$$

for structured and residual latent consistency,

$$\lambda_{\text{1step}} = 1, \qquad \lambda_{\text{rollout}} = 1,$$

for one-step and rollout structured supervision, and

$$\lambda_{\omega} = 0.25, \qquad \lambda_{\text{contact}} = 0.25, \qquad \lambda_{\text{phase}} = 2.$$

The main and structured-model optimizers use Adam with learning rate 3×10^{-4} and batch size 128.

Table 21: Main Hopper-Hop training and planning settings.

Setting	Value
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	128
Q ensemble size	3
MPPI samples	256
MPPI elites	32
Policy trajectories	24
Planning horizon	3
Planning iterations	6
Action repeat	2
Evaluation episodes / seed	10
Seeds	$1, 2, 3$

Table 22: Complete Hopper-Hop results for the structured–residual TD-MPC2 variant.

Nominal budget	Evaluated step	Seed 1	Seed 2	Seed 3	Mean $\pm$ population SD
50k	50,000	33.67	18.43	2.28	18.13 ± 12.82
100k	100,000	52.93	52.85	77.65	61.14 ± 11.67
200k	200,001	185.62	265.87	169.19	206.90 ± 42.24

The retained TD-MPC2 planner uses MPPI with 256 sampled trajectories, 32 elites, 24 policy trajectories, planning horizon 3, and six planning iterations. The Q ensemble contains three heads. Table 21 summarizes the principal formal-run settings.

The 200k experiment continues from the full 100k trainer state rather than reinitializing the algorithm, preserving the learned reward, value, policy, transition, and optimizer states across the continuation.

Complete Hopper-Hop results. Table 22 reports the structured TD-MPC2 variant at the 50k, 100k, and 200k checkpoints. Each seed value is the mean return over ten evaluation episodes. The final column reports the population standard deviation across the three seed-level means.

For reference, Table 23 reports the bundled official TD-MPC2 result trace at the same nominal 100k and 200k interaction budgets (Hansen et al., 2024). These values are public reference results rather than paired local runs.

The structured–residual variant achieves mean returns of 61.14 at 100k and 206.90 at 200k interactions, compared with public TD-MPC2 reference means of 13.63 and 166.63, respectively. These results show that the structured transition representation can be incorporated into TD-MPC2 while retaining its learned reward, Q, policy, actor objective, and MPPI planner, and can produce competitive control performance at the reported interaction budgets.

Taken together, the two integration studies illustrate complementary ways of using structural knowledge within existing model-based RL systems. DreamerV3 uses structural knowledge to shape the actions encountered during imagined training while preserving its learned recurrent world model, whereas TD-MPC2 incorporates an explicit structured–residual transition representation inside an otherwise retained value, policy, and planning pipeline. The results therefore support the modular role of structural knowledge described in Section 4: it can enter through different algorithmic interfaces rather than requiring a single planner or a single form of world model.

G Additional Experiments and Ablations

This section reports two sets of supplementary experiments that extend the main empirical study in Section 6. We first evaluate the frozen epidemic-control models under several post-training

Table 23: Hopper-Hop comparison with the bundled official TD-MPC2 reference. Values are mean ± population standard deviation across three seeds.

Method	100k	200k
Structured–residual TD-MPC2	61.14 ± 11.67	206.90 ± 42.24
TD-MPC2 public reference	13.63 ± 10.41	166.63 ± 34.89

changes in system conditions. We then report a Cartpole-Swingup study that provides an additional continuous-control instance of parameter-only structured modeling together with a planning ablation.

G.1 Epidemic Distribution Shifts and Decision-Condition Changes

Frozen-model evaluation protocol. For each arm and random seed, the model is first trained for 52 weeks under the nominal epidemic environment. The resulting model is then frozen and evaluated for a further 52-week episode under the modified condition. No additional transition data are collected for model fitting during this evaluation, and no model parameters are updated.

The formal evaluation uses seeds 0–9. We report cumulative 52-week return as mean ± sample standard deviation across the ten seeds. The principal comparison below uses the context-augmented black-box model A1' and the structured model A2, matching the main comparison in Section 6.2. The passive structured variants are reported separately.

We consider four changes in the evaluation environment. Three modify the epidemic dynamics, while the fourth changes the publicly observed hospital capacity without altering the transition dynamics.

Hidden transmission shift. Beginning at evaluation week 8, the true baseline transmission rate is changed according to

$$\beta_0' = \beta_0(1+\delta), \qquad \beta_0 = 0.5, \qquad \delta \in \{0, 0.3, 0.6\}. \tag{71}$$

The occurrence and type of the shift are part of the evaluation context, but its amplitude is not supplied to the learned model. The shifted transmission rate therefore reaches 0.65 and 0.80 for $\delta = 0.3$ and 0.6, respectively.

Known transmission-amplitude shift. We additionally evaluate the same transmission-rate changes while supplying the shift amplitude as part of the intervention context. In this setting, A1' receives the additional intervention features directly, whereas the structured transition continues to use its fitted epidemic parameters. This evaluation uses 16 MCTS simulations per decision and is therefore reported as a separate protocol from the hidden-amplitude experiment, which uses 64 simulations.

Recovery-to-susceptible shift. To introduce a stronger structural change, we activate a transition that is absent in the nominal model. The recovery-to-susceptible rate is set to

$$\omega = \delta, \qquad p_{R\to S} = 1 - \exp(-\omega), \qquad \delta \in \{0, 0.05, 0.10\}. \tag{72}$$

The change is active from the start of evaluation. This creates a mismatch between the frozen nominal model and the true evaluation dynamics.

Hospital-capacity shift. Finally, we change the public hospital-capacity condition from week 8 onward,

$$C' = C(1-\delta), \qquad C = 1000, \qquad \delta \in \{0, 0.5\}. \tag{73}$$

Thus, $\delta = 0.5$ reduces capacity from 1000 to 500. The epidemic transition dynamics remain unchanged; the modification affects the hospital-overflow decision condition and its associated reward term.

Table 24 summarizes the A1' and A2 results across these four evaluations.

Under the hidden transmission shifts, both models experience lower return as the transmission mismatch grows, while A2 maintains a higher mean return than A1' at each evaluated shift magnitude.

Table 24: Frozen-model epidemic evaluations under post-training changes. Entries are cumulative 52-week return, reported as mean ± sample standard deviation across ten seeds. Higher return is better. The known-amplitude block uses 16 MCTS simulations per decision, whereas the other blocks use 64; each block should therefore be interpreted relative to its own $\delta = 0$ baseline.

Evaluation	Shift	A1'	A2
Hidden transmission	$\delta = 0$	-49.603 ± 7.400	-37.870 ± 0.528
	$\delta = 0.3$	-51.190 ± 4.301	-39.477 ± 0.401
	$\delta = 0.6$	-52.725 ± 4.938	-41.229 ± 0.772
Known transmission amplitude	$\delta = 0$	-51.520 ± 2.970	-41.338 ± 2.402
	$\delta = 0.3$	-51.246 ± 3.132	-43.022 ± 2.059
	$\delta = 0.6$	-52.614 ± 2.947	-44.523 ± 1.664
Recovery-to-susceptible	$\delta = 0$	-49.603 ± 7.400	-37.870 ± 0.528
	$\delta = 0.05$	-195.893 ± 44.443	-141.056 ± 6.069
	$\delta = 0.10$	-231.208 ± 69.886	-171.223 ± 7.010
Hospital capacity	$\delta = 0$	-49.603 ± 7.400	-37.870 ± 0.528
	$\delta = 0.5$	-78.153 ± 3.246	-71.277 ± 1.095

Table 25: Supplementary frozen-model OOD results for the passive structured identification variants. Entries are mean return ± sample standard deviation across ten seeds.

Evaluation	Shift	A5-fixed	A5-receding
Hidden transmission	$\delta = 0$	-37.509 ± 0.578	-37.899 ± 0.566
	$\delta = 0.3$	-39.328 ± 0.581	-39.582 ± 0.505
	$\delta = 0.6$	-40.885 ± 0.514	-41.235 ± 0.556
Known transmission amplitude	$\delta = 0$	-41.336 ± 2.385	-41.320 ± 2.399
	$\delta = 0.3$	-43.006 ± 2.124	-43.006 ± 2.124
	$\delta = 0.6$	-44.584 ± 1.667	-44.523 ± 1.664
Recovery-to-susceptible	$\delta = 0$	-37.509 ± 0.578	-37.899 ± 0.566
	$\delta = 0.05$	-142.265 ± 5.784	-141.754 ± 5.631
	$\delta = 0.10$	-168.853 ± 11.317	-168.447 ± 7.668
Hospital capacity	$\delta = 0$	-37.509 ± 0.578	-37.899 ± 0.566
	$\delta = 0.5$	-71.452 ± 1.030	-71.453 ± 1.305

Providing the transmission amplitude as intervention context changes the information available in the second protocol, but A2 again achieves the higher mean return at all three evaluated conditions.

The recovery-to-susceptible experiment produces a substantially larger change in the underlying epidemic process. Activating this previously absent flow causes a marked decrease in return for all methods, with the degradation increasing as the escape rate rises. The structured model nevertheless retains higher mean return than A1' at both nonzero shift levels. These results show that the structured representation continues to provide a useful decision model under the evaluated dynamics changes, while larger structural mismatch also leads to substantial performance loss.

The passive-identification variants exhibit similar behavior, as shown in Table 25.

The close behavior of A2, A5-fixed, and A5-receding across these evaluations is consistent with the nominal results in Appendix D: the benefit of the structured representation is not tied to a single parameter-update procedure.

The hospital-capacity experiment separates model prediction from a change in the decision condition. Because the epidemic transition is unchanged, the structured models retain essentially the same one-step prediction error before and after the capacity reduction. Table 26 reports the normalized compartment-state prediction MSE.

Table 26: One-step normalized compartment-state prediction MSE under the hospital-capacity shift. Entries are mean ± sample standard deviation across ten seeds.

Model	$C = 1000$	$C = 500$
A1'	$(2.76 \pm 1.23) \times 10^{-4}$	$(4.04 \pm 1.00) \times 10^{-4}$
A2	$(2.62 \pm 4.74) \times 10^{-8}$	$(2.54 \pm 4.67) \times 10^{-8}$
A5-fixed	$(1.49 \pm 1.37) \times 10^{-8}$	$(1.50 \pm 1.43) \times 10^{-8}$
A5-receding	$(6.17 \pm 8.87) \times 10^{-8}$	$(6.08 \pm 8.60) \times 10^{-8}$

The drop in return after halving hospital capacity therefore occurs without a corresponding change in the structured transition error. This experiment illustrates the distinction between a change in system dynamics and a change in the public decision condition supplied to the planner.

G.2 CARTPOLE-SWINGUP SUPPLEMENTARY STUDY

The Cartpole-Swingup experiment provides an additional continuous-control example of the parameter-only structured setting and a supplementary planning ablation. The environment is the DMControl Cartpole-Swingup task. The physical observation contains cart position, pole orientation represented by sine and cosine, and the corresponding cart and angular velocities. Environment actions are continuous in $[-1, 1]$.

The structured world model uses a damped Newton–Euler cart-pole transition with numerical integration. Its learnable physical parameter vector is

$$\theta_{\mathrm{CP}} = (m_{\mathrm{cart}}, m_{\mathrm{pole}}, \ell_{\mathrm{pole}}), \tag{74}$$

while gravity, actuator gear, and damping terms are supplied by the mechanistic model. No learned residual transition is used, so this experiment instantiates the parameter-only case $R_\phi \equiv 0$.

Shared value prior and target-seed adaptation. The experiment uses seven development seeds,

$$\{10, 21, 30, 55, 63, 72, 95\},$$

with 5,000 environment interactions per seed, for a total of 35,000 development interactions. These data are used to learn a shared value-network prior. The physical transition parameters themselves are not shared across target seeds.

Evaluation uses three target seeds,

$$\{0, 42, 84\}.$$

For each target seed, an additional 5,000 interactions are collected for target-specific system identification and value-network fine-tuning. The value estimator is a two-hidden-layer MLP with 128 units per layer. Final scores are computed from five evaluation episodes per target seed.

Planning configuration. Planning discretizes the continuous action interval into 11 branches,

$$\{-1.0, -0.8, \ldots, 0.8, 1.0\}.$$

The full planner uses 200 MCTS simulations per decision, UCB coefficient 1.5, discount $\gamma = 0.99$, action repeat 2, and the learned value network to estimate search-frontier continuation. After the simulation budget is exhausted, the root action is selected by child visit count.

We compare this planner with a one-step value lookahead that uses the same structured transition model and target-seed value estimator but evaluates only the immediate successor before applying the continuation value estimate.

Full MCTS achieves the higher mean return on all three target seeds. Relative to one-step value lookahead, the gain is 21.7, 36.0, and 11.5 on seeds 0, 42, and 84, respectively, for an average improvement of 23.1 return points.

This experiment complements the main results in two ways. First, it provides a continuous-control instance in which a low-dimensional physical parameter vector is identified inside an otherwise

Table 27: Cartpole-Swingup planning ablation. Each seed entry is the mean return over five evaluation episodes, with episode-level population standard deviation in parentheses.

Planner	Seed 0	Seed 42	Seed 84	Mean across target seeds
Full MCTS	772.8 (33.5)	817.9 (29.6)	818.2 (18.8)	802.9
1-Step VNet Lookahead	751.0 (7.4)	781.9 (81.3)	806.7 (19.6)	779.9

supplied mechanistic transition model. Second, holding the structured model and value estimator fixed, the planning ablation shows the additional decision benefit observed when the same model is queried through a deeper tree-search procedure rather than only one-step lookahead. The result supports the modular separation between world model and downstream planner used throughout Section 4.