Leveraging Conditional Dependence for Efficient World Model Denoising

Shaowei Zhang[1,2], **Jiahan Cao**[1,2], **Dian Cheng**[1,2], **Xunlan Zhou**[3], **Shenghua Wan**[1,2], **Le Gan**[1,2], **and De-Chuan Zhan**[†1,2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Artificial Intelligence, Nanjing University, China
[3]School of Intelligence Science and Technology, Nanjing University, China
{zhangsw, caojh, chengd, wansh}@lamda.nju.edu.cn, wyattzhouxl@smail.nju.edu.cn,
{ganl, zhandc}@nju.edu.cn

Abstract

Effective denoising is critical for managing complex visual inputs contaminated with noisy distractors in model-based reinforcement learning (RL). Current methods often oversimplify the decomposition of observations by neglecting the conditional dependence between task-relevant and task-irrelevant components given an observation. To address this limitation, we introduce *CsDreamer*, a model-based RL approach built upon the world model of ***C**ollider-**s**tructure **R**ecurrent **S**tate-**S**pace **M**odel (CsRSSM)*. CsRSSM incorporates colliders to comprehensively model the denoising inference process and explicitly capture the conditional dependence. Furthermore, it employs a decoupling regularization to balance the influence of this conditional dependence. By accurately inferring a task-relevant state space, CsDreamer improves learning efficiency during rollouts. Experimental results demonstrate the effectiveness of CsRSSM in extracting task-relevant information, leading to CsDreamer outperforming existing approaches in environments characterized by complex noise interference. [1]

1 Introduction

Reinforcement Learning has achieved remarkable success in complex applications such as autonomous driving [1] and conversational interactions [2, 3]. A key factor in these advancements is the agent's ability to accurately perceive and interpret observations from its environment. However, real-world observations are often corrupted by noise and extraneous information, which can severely impair an agent's ability to make sound decisions. This challenge intensifies significantly when dealing with high-dimensional observations like images [4, 5, 6]. Therefore, effectively managing such noisy observations is essential for developing robust and reliable RL agents.

From a generative model perspective [7], at timestep t, an observation o_t is generated by two distinct sets of latent variables: task-relevant variables s_t, which directly influence the agent's rewards or actions, and task-irrelevant variables c_t, which introduce noise and distractions without contributing to the task (Figure 1 (**a**)). Existing model-based RL approaches have sought to extract task-relevant information from observations [8, 9]. Some methods utilize two separate dynamics models to extract task-relevant and task-irrelevant information from observations [10, 11], while others decompose observations into task-relevant and task-irrelevant branches [12] or pursue more fine-grained decomposition [13, 14]. However, these methods oversimplify the decomposition of observations by neglecting the conditional dependence between s_t and c_t given o_t. The importance of this conditional dependence can be illustrated with a simple example: consider random variables

†: Corresponding author.

[1]The code is available at https://github.com/Zhang-Shaowei/CsDreamer.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

A, B, and C where $A + B = C$. If C is unknown, A and B are independent. However, once C is observed (e.g., $C = c$), their joint probability becomes $P(a, b \mid C = c) = \frac{P(a,b)\mathbb{I}(a+b=c)}{P(C=c)}$. This implies that if we determine $A = a$, then B must be $c - a$. Thus, A and B become conditionally dependent given C. This conditional dependence can be leveraged for efficient denoising. For instance, as illustrated in Figure 1 (**b**), previous approaches perform separate inferences for s_t (the walker's state) and c_t (the background) by assuming conditional independence. However, the background information can significantly influence the inference of the walker's state. By capturing the conditional dependence between s_t and c_t, we can significantly enhance the inference process, as the walker's state can be rapidly deduced by comparing differences between the background and the noisy observation. Therefore, capturing this conditional dependence is beneficial and can improve the inference process.

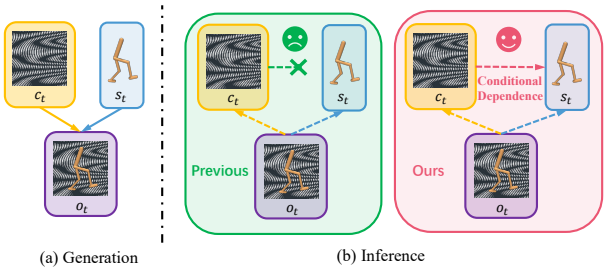


Figure 1: (**a**) Generation process of noisy observations. (**b**) Inference process where we consider the conditional dependence.

Inspired by this, we propose ***C**ollider-structure **R**ecurrent **S**tate-**S**pace **M**odel* (CsRSSM), a world model designed for denoising from a generative modeling perspective. CsRSSM extends Recurrent State-Space Model (RSSM) [15] by modeling the noisy observation generation process as a sequential collider-structure model. By explicitly utilizing the collider structures, CsRSSM exploits the conditional dependence between s_t and c_t given o_t to effectively extract task-relevant information. Specifically, CsRSSM first infers the task-irrelevant variable c_t from o_t and then utilizes both c_t and o_t to infer the task-relevant latent variable s_t. This sequential inference approach utilizes the contextual information provided by task-irrelevant variables to enhance the efficiency of task-relevant information extraction. Additionally, we regularize the model using conditional mutual information to balance the conditional dependence between s_t and c_t during learning. Furthermore, existing methods typically employ masking mechanisms [10, 12] or tailored optimization objectives [16], which depend on prior knowledge or assumptions, to distinguish task-relevant information from task-irrelevant information. Compared with these strategies which may constrain model flexibility and introduce biases, CsRSSM adopts a generative modeling perspective, employing dual latent variables to model the generation of noisy observations and the environment dynamics. It relies solely on the decoupling regularization and the inherent optimization objectives of the generative model, without incorporating any additional prior knowledge or supplementary optimization objectives. Experimental results demonstrate the feasibility and effectiveness of this approach.

Finally, we introduce *CsDreamer*, an extension of the Dreamer framework [17], that employs CsRSSM to effectively extract task-relevant information from noisy observations. By conducting rollouts exclusively within the space of s_t, CsDreamer enhances the agent's ability to efficiently learn policies in high-dimensional visual environments characterized by complex noise distractors. Our contributions are threefold:

- We propose CsRSSM, a novel generative model for environment dynamics that explicitly utilizes conditional dependence to model the noisy observation generation.
- We introduce CsDreamer, which builds upon CsRSSM to effectively denoise observations and train policies within the task-relevant space.
- We conduct extensive experiments demonstrating that our approach outperforms existing state-of-the-art methods in handling noisy observations.

2 Preliminaries

Recurrent State-Space Model. A Recurrent State-Space Model (RSSM) [15] can capture the dynamic relationships in the latent space. Dreamer [18, 19, 17] adopts a world model based on RSSM to construct the dynamics model of the environment perceived by the agent in the latent space. In the real environment, the agent receives an observation input o_t and performs an action a_t, after which it receives the next observation o_{t+1} following the environment's state transition governed by the probability distribution $P(o_{t+1}|o_t, a_t)$. The world model captures the dynamics in observation space based

on a dynamics model in the latent space $\mathcal{Z}$, which primarily comprises the following components: the trajectory history model $h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$, the encoder model $q_\phi(z_t|h_t, o_t)$, the transition model in latent space $p_\phi(\hat{z}_t|h_t)$ and the reward model $p_\phi(\hat{r}_t|h_t, z_t)$, where ϕ represents the parameter vector of the world model. The dynamics model in $\mathcal{Z}$ uses the same action space $\mathcal{A}$ as the real environment. In the dynamics model, h_t records the historical trajectory information, $q_\phi(z_t|h_t, o_t)$ extracts the useful information from the observation, and $p_\phi(\hat{z}_t|h_t)$ represents the transition in the latent space.

3 Related Work

Reinforcement Learning with Noisy Observations. Many recent studies have focused on the reinforcement learning problem with noisy observations. Some studies adopt metric-based representation learning approaches to deal with noisy observations [20, 21, 22, 23]. MIIR [24] adopts information-theoretic principles to learn invariant representations. InfoPower [8] prioritizes action-correlated factors. RePo [6] encourages reward-predictive yet compact encodings. Another line of work relies on data augmentation to reduce distractors [25, 26, 27]. Additionally, structured latent variables have been explored for denoising. SEAR [28] partitions agent- and environment-related components via segmentation, while DEAR [29] applies a similar mask-based approach without reconstructing the entire observation. Other methods decompose observations by training two separate dynamics models to separate task-relevant and task-irrelevant information [10, 11], splitting them into distinct branches [12], or adopting more fine-grained decomposition [13, 14]. However, these decomposition-based approaches typically assume conditional independence between factors given the observation.

Generative Models in Reinforcement Learning. Generative models have played a pivotal role in tackling high-dimensional observations for RL. Some works employ the Variational Autoencoder (VAE) [30, 31] to learn lower-dimensional latent variables from images, facilitating latent-space policy optimization [32, 15]. Flow-based generative models [33, 34] also provide a way to learn flexible distributions with explicit likelihoods, though balancing model complexity with real-time efficiency can be challenging in RL settings. Diffusion models [35, 36] have demonstrated impressive performance in image generation, and their potential for RL has begun to attract attention [37]. However, most of these generative approaches focus primarily on high-fidelity reconstruction or predictive modeling, as opposed to explicitly separating out different observational factors such as background noise or irrelevant distractors.

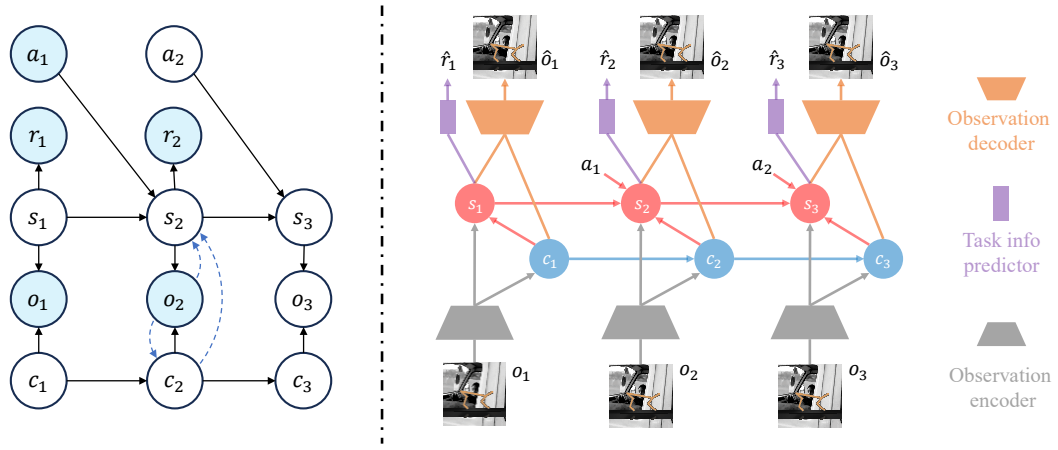
Disentanglement in Reinforcement Learning. Several studies concentrate on disentanglement within reinforcement learning to derive efficient representations for behavioral learning. TED [38] and CMID [39] introduce disentanglement techniques into feature learning process for RL. Our method draws inspiration from them while focusing on decoupling between two sets of latent variables rather than disentangling the original feature into individual feature dimensions. Some causal reinforcement learning approaches target the disentanglement of different state components [40, 41, 16], while they pay less attention to the observation generation process or employ counterfactual and intervention mechanisms [42, 43]. Instead, we treat noisy observations from a generative model perspective and focus on conditional dependence between task-relevant and task-irrelevant variables during inference. Although Cao et al. [44] also notice the conditional dependence, they combine two separate generative models rather than utilizing a unified model directly.

4 Method

In this section, we first introduce the underlying assumption regarding the structure of noisy observations and then present the ***C**ollider-**s**tructure **R**ecurrent **S**tate-**S**pace **M**odel* (CsRSSM) from a generative modeling perspective in Section 4.1. Subsequently, we introduce a decoupling regularization according to the characteristics of the network in Section 4.2 to facilitate subsequent reinforcement learning. We present the overall loss objective for the CsRSSM world model in Section 4.3 and introduce the policy learning in Section 4.4. The overall framework of the CsRSSM world model is shown in Figure 2.

4.1 Collider-Structure Recurrent State-Space Model

In real-world scenarios, agent observations are often contaminated by noise and irrelevant information, which degrades decision-making performance. From a generative modeling perspective, we propose the following assumption to elucidate the relationship between latent variables and the generation of observations in such environments.



(a) (b)

Figure 2: Framework of the CsRSSM world model. For brevity, content related to historical information in Eq. (3) is omitted. (**a**) Probabilistic graphical model of CsRSSM. Shaded nodes represent observed variables such as actions and observations. Among the unshaded nodes, a_2 and o_3 are future variables that have not yet been observed at timestep $t = 2$, while the remaining unshaded nodes are latent variables. The black solid lines represent the generative model, and the blue dotted lines illustrate the inference process, which accounts for conditional dependence. (**b**) The training framework of the world model.

Assumption 4.1. ***(Collider structure assumption)*** *There exist two distinct sets of latent variables that jointly determine the generation of the observation $o_t \in \mathcal{O}$: the task-relevant latent variable $s_t \in \mathcal{S}$, which is associated with the agent's actions and rewards, and the task-irrelevant latent variable $c_t \in \mathcal{C}$, which pertains solely to interference.*

This assumption posits that the observations an agent perceives are a composite of two underlying factors and acknowledges the conditional dependence between the latent variables s_t and c_t given the observation o_t. This nuanced modeling ensures that the interplay between task-relevant and task-irrelevant factors is accurately captured, thereby enhancing the agent's ability to make informed and robust decisions despite the presence of noise and irrelevant information. Under Assumption 4.1, we can obtain the following generative process

$$p(s_t, c_t, o_t) = p(c_t)p(s_t)p(o_t|c_t, s_t), \tag{1}$$

which is similar to the former methods [10, 12, 13]. Due to the presence of collider structures, conditional independence typically does not hold during inference; that is, $p(c_t, s_t|o_t) \neq p(c_t|o_t)p(s_t|o_t)$. Instead, we recognize the dependence between s_t and c_t and employ the chain rule of probability rather than the arbitrary factorization based on conditional independence to factorize the inference probability. There are two ways to implement the probability chain rule: either by inferring c_t first or inferring s_t first. In scenarios where agents receive noisy observations, the provision of additional task-irrelevant information can aid in inferring task-relevant information given the observations of an agent, as indicated by a reduction in conditional entropy: $\mathcal{H}(s_t|c_t, o_t) \leq \mathcal{H}(s_t|o_t)$. Therefore, we utilize

$$p(c_t, s_t|o_t) = p(c_t|o_t)p(s_t|c_t, o_t), \tag{2}$$

where the main difference lies in the inclusion of c_t for the inference of s_t. We first use the observation o_t to infer c_t, the task-irrelevant information, and then utilize both c_t and o_t to infer s_t, the task-relevant information.

To better capture the temporal information in noise, we model the task-irrelevant variables c_t as time-varying. Constant noise can be regarded as a special case of time-varying noise where c_t remains constant. In practice, the noise at the next moment is primarily influenced by the noise at the current moment. Hence, we posit the following assumption.

Assumption 4.2. *The transitions of the task-irrelevant latent variables c_t satisfy the Markov property, i.e., $c_t \sim p(c_t|c_{<t}) = p(c_t|c_{t-1})$.*

Remark 4.3. *Assumption 4.2 is analogous to the Markov property in Reinforcement learning, with the distinction that transitions of c_t are independent of the agent's actions.*

Consider the sequences $\{o_t, a_t, r_t, x_t\}_{t=1}^{T}$, where t is the timestep, a_t is the action taken by the agent, r_t is the reward signal, x_t is the episode continuation flag and T is the length of the sequence. Under Assumption 4.1 and Assumption 4.2, we propose the CsRSSM world model, which leverages the conditional dependence of the collider structures to model the noisy observation generation process from a generative modeling perspective:

$$\begin{cases} \text{Task-irrelevant history:} & h_t^c = f_\phi(h_{t-1}^c, c_{t-1}) \\ \text{Task-relevant history:} & h_t^s = f_\phi(h_{t-1}^s, s_{t-1}, a_{t-1}) \\ \text{Task-relevant transition:} & p_\phi(\hat{s}_t | h_t^s) \\ \text{Task-irrelevant transition:} & p_\phi(\hat{c}_t | h_t^c) \\ \text{Task-irrelevant encoder:} & q_\phi(c_t | h_t^c, h_t^s, o_t) \\ \text{Task-relevant encoder:} & q_\phi(s_t | h_t^c, h_t^s, c_t, o_t) \\ \text{Reconstruction model:} & p_\phi(\hat{o}_t | h_t^s, s_t, h_t^c, c_t) \\ \text{Reward model:} & p_\phi(\hat{r}_t | h_t^s, s_t), \\ \text{Continuation predictor:} & p_\phi(\hat{x}_t | h_t^s, s_t) \end{cases} \tag{3}$$

where ϕ is the parameter vector of the world model. Analogous to the RSSM [15], our approach incorporates historical information. We denote the task-irrelevant history as h_t^c and the task-relevant history as h_t^s. The transition models provide the transition priors: $p_\phi(\hat{s}_t | h_t^s)$ predicts the next task-relevant state based on the current task-relevant history, while $p_\phi(\hat{c}_t | h_t^c)$ predicts the next task-irrelevant state from the current task-irrelevant history. The two observation encoders operate similarly to the transition models but incorporate observations as inputs.

The inclusion of historical information introduces conditional dependence, and the posterior distributions of s_t and c_t capture all conditional dependence between task-relevant and task-irrelevant information within the collider structures given the observations. Specifically, for each collider structure, the model first uses the history and the current observation o_t to infer the task-irrelevant information c_t. Subsequently, it utilizes the history along with both o_t and c_t to infer the task-relevant information s_t. We use the features of both c_t and s_t to reconstruct the observation. In contrast, we only use the features of s_t to predict the task information, such as rewards and episode continuation flags. These task information reconstructions facilitate the learning of task-relevant latent variables s_t. Finally, we derive the evidence lower bound (ELBO) loss for the CsRSSM world model from a generative modeling perspective:

$$\begin{aligned} \mathcal{L}_{\text{ELBO}} = -\sum_{t=1}^{T} \Big[& \mathbb{E}[\ln p_\phi(o_t | h_t^s, s_t, h_t^c, c_t) + \ln p_\phi(r_t | h_t^s, s_t) + \ln p_\phi(x_t | h_t^s, s_t)] \\ & - \mathbb{E}[\mathrm{KL}[q_\phi(c_t | h_t^c, h_t^s, o_t) \| p_\phi(c_t | h_t^c)]] - \mathbb{E}[\mathrm{KL}\left[q_\phi(s_t | h_t^c, h_t^s, c_t, o_t) \| p_\phi(s_t | h_t^s)\right]] \Big], \end{aligned} \tag{4}$$

where we maximize the log-likelihood of observations, rewards and continuation flags. The two KL divergence losses serve separately for task-relevant and task-irrelevant information, simultaneously training the priors toward the representations and regularizing the representations toward the priors. The derivation of Eq. (4) can refer to Appendix A.1. Notably, when faced with noisy observations, the presence of conditional dependence mitigates trivial solutions where $c_t = \mathbf{0}$ and $s_t = o_t$, ensuring that both c_t and s_t are effectively utilized.

4.2 Decoupling Regularization in CsRSSM

Directly training the CsRSSM world model may present some issues. Specifically, since the model training primarily relies on the reconstruction objectives, there is a risk of conflating the task-relevant and task-irrelevant variables during the training process. This conflation can hinder the model's ability to accurately disentangle the underlying factors of the observations, ultimately degrading performance and slowing convergence. To mitigate these issues, we propose explicitly decoupling the task-relevant latent variables s_t from the task-irrelevant latent variables c_t. We

measure the information shared between s_t and c_t given o_t using the conditional mutual information, denoted as $I(s_t; c_t|o_t)$. Noting the relationship that the KL divergence between conditional dependence and conditional independence is equivalent to the conditional mutual information, i.e., $\mathrm{KL}[p(c_t|o_t)p(s_t|c_t, o_t)\|p(c_t|o_t)p(s_t|o_t)] = I(s_t; c_t|o_t)$, we achieve decoupling by introducing a regularization loss that minimizes the conditional mutual information between s_t and c_t given the condition variables $\{o_{\leq t}, a_{<t}\}$, denoted as $I(s_t; c_t|h_t^c, h_t^s, o_t)$. Minimizing this mutual information encourages the task-relevant and task-irrelevant variables to capture distinct and non-overlapping aspects of the observations, thereby facilitating more effective learning for CsRSSM. Directly minimizing the conditional mutual information is intractable. By incorporating $q_\xi(s_t|h_t^c, h_t^s, o_t)$, a variational distribution with parameters ξ that approximates the true posterior, we can obtain an upper bound loss for the conditional mutual information:

$$\mathcal{L}_{\mathrm{MI}} = \mathbb{E}[\mathrm{KL}[q_\phi(s_t|c_t, h_t^c, h_t^s, o_t)\|q_\xi(s_t|h_t^c, h_t^s, o_t)]], \tag{5}$$

where $q_\phi(s_t|c_t, h_t^c, h_t^s, o_t)$ is the task-relevant encoder of CsRSSM. The Derivation can refer to Appendix A.2. We can minimize the conditional mutual information for the task-relevant latent variables s_t and the task-irrelevant latent variables c_t by minimizing the upper bound in Eq. (5).

4.3 Overall Objective for CsRSSM

Integrating the discussions in Sections 4.1 and 4.2, we obtain the final loss function for CsRSSM world model

$$\mathcal{L}_{\mathrm{CsRSSM}} = \mathcal{L}_{\mathrm{ELBO}} + \lambda\mathcal{L}_{\mathrm{MI}}, \tag{6}$$

where the second item serves as the regularization of the ELBO loss to balance the conditional dependence between s_t and c_t. The hyperparameter λ controls the weight of the regularization. A larger λ enforces greater separation between s_t and c_t, promoting distinct representations of task-relevant and task-irrelevant information within the latent space, while a smaller λ leverages the conditional dependence to facilitate more efficient inference.

Previous studies operate under the conditional independence assumption for s_t and c_t [10, 12] given an observation. This assumption is a special case of our formulation where the regularization coefficient λ approaches infinity. In such scenarios, $\mathcal{L}_{\mathrm{MI}}$ dominates the loss function, effectively reducing the mutual information between s_t and c_t to zero. This enforces strict independence between the task-relevant and task-irrelevant variables given an observation, aligning with the conditional independence assumption. Our approach generalizes existing methods by introducing a finite λ, enabling a flexible trade-off for the degree of decoupling enforced between s_t and c_t. This extension incorporates the conditional dependence between s_t and c_t for world model denoising. More experimental discussions are provided in Section 5.3.

Moreover, existing methods typically employ masking mechanisms [10, 12] or tailored optimization objectives [16], which depend on prior knowledge or assumptions, to distinguish task-relevant information from task-irrelevant information. However, these strategies may constrain model flexibility and introduce biases. Instead, we adopt a generative modeling perspective, employing dual latent variables to model the generation of noisy observations and the environment dynamics. The model is trained solely based on the inherent optimization objectives of the generative model (the reconstruction errors and the prior regularizations in Eq. (4)), and the decoupling regularization according to the model network (Eq. (5)). It does not incorporate any additional prior knowledge or supplementary optimization objectives.

4.4 Policy Learning in CsDreamer

Finally, we propose *CsDreamer*, which is an extension of Dreamer [17] built upon CsRSSM. The agent's policy is learned during the imagination phase of the CsRSSM. By focusing on task-relevant information, we only utilize s_t to choose the next action. i.e., $a_t \sim \pi_\theta(a_t \mid h_t^s, s_t)$, where π is the policy of the agent. During the imagination phase, the agent interacts solely with the task-relevant transition model, $p_\phi(\hat{s}_t|h_t^s)$, to generate rollout trajectories, with rewards predicted by the reward model $p_\phi(\hat{r}_t|h_t^s, s_t)$. We predict the value utilizing $V_\psi(h_t^s, s_t)$ based on s_t. The agent policy is trained analogously to Dreamer [17]. Focusing on task-relevant information and leveraging the strengths of model-based reinforcement learning ensure efficient and targeted policy development.

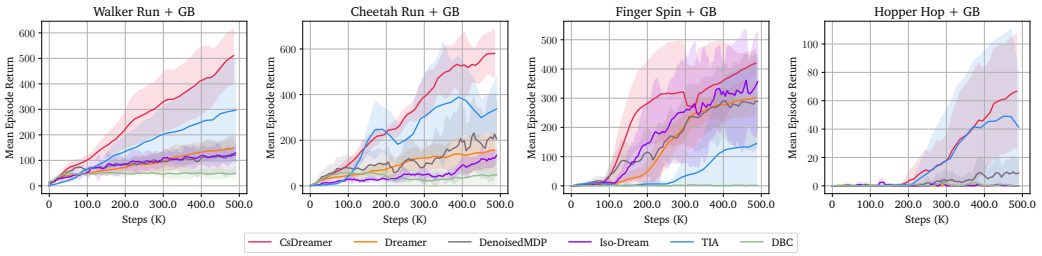


Figure 3: Performance on DMC using gray natural videos as background.

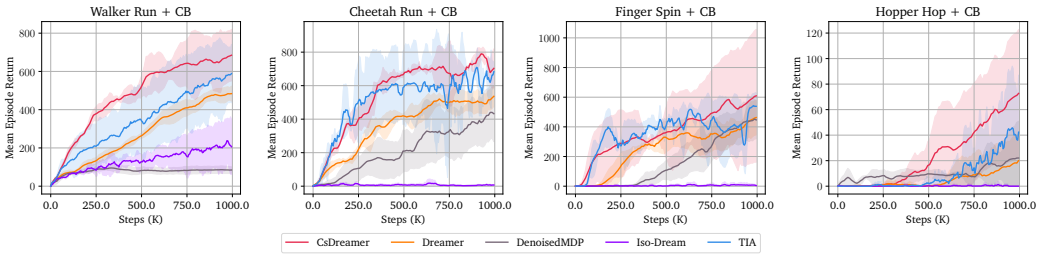


Figure 4: Performance on DMC using colorful natural videos as background.

5 Experiments

We begin by describing the experimental setup in Section 5.1. Subsequently, we address the following research questions: (1) Can CsDreamer enhance the training performance in complex environments with distractors? (Section 5.2); (2) What role does the decoupling regularization loss play in the overall objective, and how do the proposed modules affect performance? (Section 5.3); (3) Do the latent variables in the proposed framework exhibit interpretable semantics upon reconstruction visualization? (Section 5.4).

5.1 Experiment Setup

Benchmarks. We evaluate our model and baselines on visual control tasks. First, we assess performance on four DeepMind Control Suite (DMC) [45] tasks: *Walker Run*, *Cheetah Run*, *Finger Spin* and *Hopper Hop*. To introduce noise distractors, we replace the original backgrounds with two types of task-irrelevant information. The first is a **gray** background composed of natural videos from the Kinetics 400 dataset [46], following the DBC [21] configuration (denoted as **GB**). The second is a **colorful** background derived from DAVIS 2017 videos [47], adhering to the background distractor settings in Distracting Control Suite [48] (denoted as **CB**). These benchmarks require the agent to extract task-relevant information, identify the target entity within the DMC environment, and effectively filter out background distractions. Subsequently, we evaluate these approaches in the more realistic simulated driving environment, CARLA [49]. Here, the agent must extract task-relevant information from visual perception while mitigating distractions such as trees and dynamic sunlight. We also conduct experiments on the complex Atari 100K benchmark [50] in Appendix E.2. Further details on the experimental setup can be found in Appendix C.

Baselines. To evaluate the effectiveness of our proposed CsDreamer framework, we compare it against several state-of-the-art model-based RL algorithms known for their strong performance in high-dimensional observation environments. Our primary baselines include Dreamer [17], TIA [10], Iso-Dream [12] and Denoised MDP [13]. Dreamer is a classic work in model-based RL but does not specifically address environments with noisy distractors. TIA targets settings with significant distractions by employing separate dynamics models for task-relevant and task-irrelevant features and utilizing masking techniques to isolate essential information. Iso-Dream incorporates separate task-relevant and task-irrelevant branches within a single world model to focus on aspects of the environment that the agent can influence. Denoised MDP decomposes observations into more fine-grained components, enhancing performance in complex environments with noisy distractors. Additionally, we compare our method with DBC [21], a model-free RL method, in the DMC environment with gray natural videos as background.

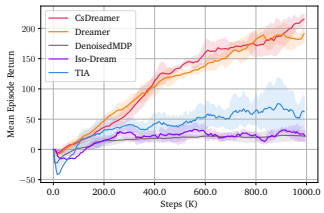


Figure 5: Performance on the autonomous driving simulator CARLA.

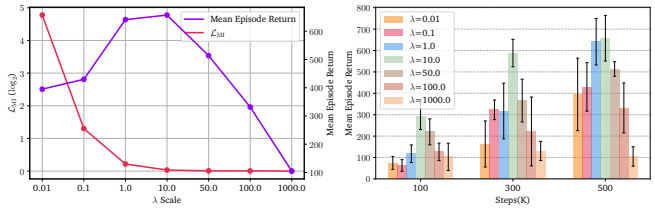


Figure 6: Ablation study across different λ on *Cheetah Run* using gray driving car as background. **Left** is the relationship between mutual information upper bound and mean episode return at 500K timesteps. **Right** is the mean episode return over varying steps.

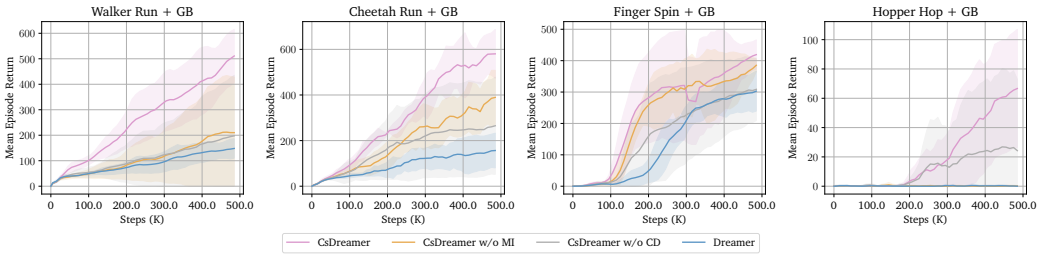


Figure 7: Ablation Performance on DMC using gray natural videos as background.

5.2 Performance in Visual Inputs with Complex Noisy Distractors

Figure 3 presents the performance of various methods on DMC with gray natural video backgrounds. Solid lines represent mean episode returns, while shaded areas represent the 95% confidence intervals. All methods show some performance in the *Walker Run* and *Cheetah Run* tasks. While most approaches performed well on the *Finger Spin* task, the model-free method, DBC, lags behind. Only CsDreamer, TIA, and DenoisedMDP demonstrate substantial performance on the more complex *Hopper Hop* task. Notably, CsDreamer consistently outperforms all baseline methods across all four benchmark tasks. Figure 4 presents the performance on DMC with colorful backgrounds. The results also show that CsDreamer consistently outperforms all baseline methods across the four tasks. Iso-Dream exhibits the lowest performance in every task, likely due to its difficulty in managing complex and colorful background distraction. In the *Cheetah Run* and *Finger Spin* tasks, TIA shows competitive performance, approaching that of CsDreamer. In addition, Dreamer also delivers promising results within this benchmark, which may be attributed to the characteristics of the dataset used in Distracting Control Suite [48]. Specifically, it introduces background variations with multiple non-continuous images, making it easier for Dreamer to capture and disregard background-related features compared to scenarios using original videos as backgrounds, as shown in Figure 3.

We also evaluate these algorithms on the CARLA simulator, a more realistic benchmark for autonomous driving. The results are shown in Figure 5. All the methods exhibit similar performance at the first 200K timesteps. Although the interference-handling baselines such as TIA and DenoisedMDP perform well in the DMC environment with noisy background distractors, they fail to achieve comparable performance in the CARLA environment. This discrepancy arises from CARLA's substantial complexity, where these methods struggle to effectively distinguish task-relevant information from irrelevant distractions in first-person views and fail to extract task-relevant features from the highly intricate and volatile input observation. In contrast, CsDreamer demonstrates superior adaptability, outperforming all baselines and highlighting its strength in addressing more realistic and complex reinforcement learning tasks. Notably, Dreamer achieves impressive performance in the CARLA environment, slightly trailing behind CsDreamer. This strong performance is attributable to Dreamer's approach of encoding all environmental information into its latent space, which ensures the preservation of critical task-relevant features essential for autonomous driving. However, the inclusion of irrelevant information may slightly hinder its efficiency and robustness compared to CsDreamer.

5.3 Ablation Study

The Role of Regularization. Figure 6 illustrates the effect of varying the coefficient λ in Eq. (6). As λ increases, the upper bound of mutual information progressively decreases, signaling a reduced conditional dependence between task-relevant and task-irrelevant variables. Meanwhile, the mean

episode return exhibits a non-monotonic trend, initially increasing and then decreasing. A small λ weakens the constraint imposed by the mutual information upper bound, heightening the risk of confounding task-relevant with task-irrelevant information, as previously discussed. Conversely, a large λ minimizes the association between these variables. The near independence of c_t and s_t, given the observation o_t, results in poorer performance, thereby validating the analysis presented in Section 4.3.

Effectiveness of Different Modules. We conduct ablation studies to evaluate the contributions of the conditional dependence and decoupling regularization modules in CsDreamer. Specifically, we compare the performance of **CsDreamer** and **Dreamer** against:

- **CsDreamer w/o CD**: CsDreamer using the CsRSSM framework without conditional dependence for s_t and c_t given o_t in collider structures.
- **CsDreamer w/o MI**: CsDreamer with the CsRSSM framework excluding decoupling regularization.

The implementation details can refer to Appendix D.3. As shown in Figure 7, **CsDreamer w/o CD** achieves mean episode returns comparable to or higher than those of **Dreamer**, demonstrating the effectiveness of structural decomposition. However, it underperforms compared to **CsDreamer w/o MI**, highlighting the importance of conditional dependence. Introducing decoupling regularization, **CsDreamer** performs better than **CsDreamer w/o MI**. It demonstrates that the decoupling regularization can mitigate confusion between task-relevant and task-irrelevant variables, as mentioned in Section 4.2. **CsDreamer** consistently outperforms all other models across all tasks, achieving the best performance. It validates the effectiveness of CsRSSM and the efficient utilization of task-relevant information within the behavior policy of **CsDreamer**, ultimately enhancing overall performance. Additional ablation experiments are provided in Appendix E.5.

5.4 Visualization of The Latent Variables

In this section, we reconstruct the visual input observation to interpret semantic information extracted by s_t and c_t. Figure 8 presents the reconstruction visualization results for *Cheetah Run + GB*. We sample a trajectory and record both ground truth and reconstructed images at selected timesteps. The first row displays the ground truth of the trajectory. The second row shows reconstructions using the model $p_\phi(\hat{o}_t|h_t^s, s_t, h_t^c, c_t)$ from CsRSSM, which concatenates features from both posterior s_t and c_t. To isolate the semantic information captured by c_t, we set all feature dimensions of s_t to zero and concatenate it with posterior c_t as the features for reconstruction, denoted as $p_\phi(\hat{o}_t|\mathbf{0}, \mathbf{0}, h_t^c, c_t)$ in the third row. Conversely, to focus on information from s_t, we set features of c_t to zero and reconstruct using $p_\phi(\hat{o}_t|h_t^s, s_t, \mathbf{0}, \mathbf{0})$, shown in the fourth row. In the fifth row, we obtain the prior $\hat{s}_t$ by imagining within the task-relevant transition model $p_\phi(\hat{s}_t|h_t^s)$ using the first five posterior s_t features and then reconstruct the subsequent observations accordingly.

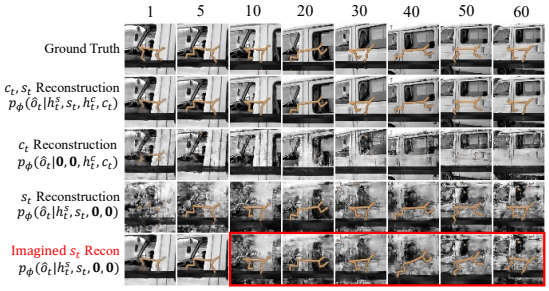


Figure 8: Reconstruction visualization of *Cheetah Run + GB*.

The second row demonstrates that combining features from s_t and c_t facilitates effective reconstruction. In the third row, the cheetah entity is nearly imperceptible, leaving only background automobile details. Conversely, the fourth row exhibits a highly blurred background while the cheetah remains clear. This indicates that c_t effectively captures task-irrelevant information, whereas s_t captures task-relevant information. In the fifth row, the cheetah in the first 30 frames closely matches the ground truth, and the background is significantly blurred. These observations suggest that performing imagination in S space allows the model to focus on task-relevant dynamics. These results illustrate why CsDreamer achieves outstanding performance. Additional visualization experiments are provided in Appendix E.8.

6 Conclusion and Future Work

In this work, we adopt a generative modeling perspective and exploit the conditional dependence between task-relevant and task-irrelevant latent variables for both observation generation and environment dynamics in scenarios with complex noisy distractors. By effectively separating task-relevant features from irrelevant background interference, CsDreamer enhances the agent's decision-making

capability. Our experiments confirm its superior performance in noisy environments. We also acknowledge the limitations, including the assumption of Markovian, action-independent noise and a simplified binary partition of latent variables, which may not capture the complexity of real-world scenarios. The model's capability carries some societal implications. While it can improve safety in applications like autonomous navigation and robotics, it also presents some risks. These include overreliance, where a critical safety signal could be erroneously ignored, and dual-use potential, where the technology could be repurposed for invasive surveillance. Our future work will address these challenges on two fronts. Technically, we will focus on modeling more complex non-Markovian dynamics and exploring richer, more structured latent representations. Concurrently, we will prioritize research into enhancing model transparency to mitigate overreliance and will investigate technical safeguards, such as built-in privacy-preserving mechanisms, to deter misuse. We believe this dual focus on technical advancement and ethical considerations is essential for the responsible development of robust autonomous agents.

Acknowledgments and Disclosure of Funding

We would like to thank Yucen Wang and the anonymous reviewers for their helpful discussions and support. This work was supported by the National Science Foundation of China (62476123) and the Young Scientists Fund of the National Natural Science Foundation of China (PhD Candidate) (624B200197).

References

[1] Noriaki Hirose, Dhruv Shah, Kyle Stachowicz, Ajay Sridhar, and Sergey Levine. SELFI: autonomous self-improvement with reinforcement learning for social navigation. *CoRR*, 2024.

[2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

[3] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, 2025.

[4] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019.

[5] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

[6] Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. Repo: Resilient model-based reinforcement learning by regularizing posterior predictability. In *Advances in Neural Information Processing Systems*, 2023.

[7] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *CoRR*, 2016.

[8] Homanga Bharadhwaj, Mohammad Babaeizadeh, Dumitru Erhan, and Sergey Levine. Information prioritization through empowerment in visual model-base RL. In *International Conference on Learning Representations*, 2022.

[9] Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, 2022.

[10] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi S. Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, 2021.

[11] Yucen Wang, Shenghua Wan, Le Gan, Shuai Feng, and De-Chuan Zhan. AD3: implicit action is the key for world models to distinguish the diverse visual distractors. In *International Conference on Machine Learning*, 2024.

[12] Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. In *Advances in Neural Information Processing Systems*, 2022.

[13] Tongzhou Wang, Simon S. Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. In *International Conference on Machine Learning*, 2022.

[14] Yuren Liu, Biwei Huang, Zhengmao Zhu, Hong-Long Tian, Mingming Gong, Yang Yu, and Kun Zhang. Learning world models with identifiable factorization. In *Advances in Neural Information Processing Systems*, 2023.

[15] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

[16] Biwei Huang, Chaochao Lu, Liu Leqi, José Miguel Hernández-Lobato, Clark Glymour, Bernhard Schölkopf, and Kun Zhang. Action-sufficient state representation learning for control with structural constraints. In *International Conference on Machine Learning*, 2022.

[17] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, 2023.

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

[19] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[20] Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021.

[21] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

[22] Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. In *Advances in Neural Information Processing Systems*, 2021.

[23] Jianda Chen and Sinno Jialin Pan. Learning representations via a robust behavioral metric for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

[24] Shuo Wang, Zhihao Wu, Jinwen Wang, Xiaobo Hu, Youfang Lin, and Kai Lv. How to learn domain-invariant representations for visual reinforcement learning: An information-theoretical perspective. In *International Joint Conference on Artificial Intelligence*, 2024.

[25] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, 2020.

[26] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *Advances in Neural Information Processing Systems*, 2021.

[27] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

[28] Kevin Gmelin, Shikhar Bahl, Russell Mendonca, and Deepak Pathak. Efficient RL via disentangled environment and agent representations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning*, 2023.

[29] Ameya Pore, Riccardo Muradore, and Diego Dall'Alba. DEAR: disentangled environment and agent representations for reinforcement learning without reconstruction. *CoRR*, 2024.

[30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[31] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[32] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

[33] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

[34] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

[35] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[37] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[38] Mhairi Dunion, Trevor McInroe, Kevin Sebastian Luck, Josiah P. Hanna, and Stefano V. Albrecht. Temporal disentanglement of representations for improved generalisation in reinforcement learning. In *International Conference on Learning Representations*, 2023.

[39] Mhairi Dunion, Trevor McInroe, Kevin Sebastian Luck, Josiah Hanna, and Stefano V. Albrecht. Conditional mutual information for disentangled representations in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

[40] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, 2020.

[41] Andrew Bennett, Nathan Kallus, Lihong Li, and Ali Mousavi. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

[42] Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, 2022.

[43] Lars Buesing, Theophane Weber, Yori Zwols, Nicolas Heess, Sébastien Racanière, Arthur Guez, and Jean-Baptiste Lespiau. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019.

[44] Haiyao Cao, Zhen Zhang, Panpan Cai, Yuhang Liu, Jinan Zou, Ehsan Abbasnejad, Biwei Huang, Mingming Gong, Anton van den Hengel, and Javen Qinfeng Shi. Rethinking state disentanglement in causal reinforcement learning. *CoRR*, 2024.

[45] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, 2018.

[46] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, 2017.

[47] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, 2017.

[48] Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite - A challenging benchmark for reinforcement learning from pixels. *CoRR*, 2021.

[49] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *Annual Conference on Robot Learning*, 2017.

[50] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020.

[51] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *CoRR*, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe that we propose to leverage conditional dependence to implement efficient world model denoising.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mainly describe the limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide a full set of assumptions for each theoretical result in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information to reproduce the results in the paper, and we also provide the code in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code in the supplementary materials, and give the instructions to run the code in the README.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We give all the training and test details in the experiment part and the related appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use the confidence intervals and the error bars in our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information in Appendix D.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive and negative societal impacts in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite them explicitly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Detailed Derivations

A.1 Derivation of ELBO Loss in CsRSSM

We use $o_{1:T}$, $a_{1:T}$, $r_{1:T}$ and $x_{1:T}$ for observation sequence, action sequence, reward sequence and episode continuation flag sequence. Similarly, we also use $s_{1:T}$ and $c_{1:T}$ for task-relevant and task-irrelevant latent variables. After introducing the variational posterior $q(s_{1:T}, c_{1:T}|o_{1:T}, a_{1:T}) = \prod_t q(c_t|o_{\leq t}, a_{<t})q(s_t|c_t, o_{\leq t}, a_{<t})$, we can obtain the variational bound for the dynamics models $p((o_{1:T}, r_{1:T}, x_{1:T}), s_{1:T}, c_{1:T}|a_{1:T}) = \prod_t p(s_t|s_{t-1}, a_{t-1})p(c_t|c_{t-1})p(o_t|s_t, c_t)p(r_t|s_t)p(x_t|s_t)$ in CsRSSM using Jensen's inequality

$$
\begin{aligned}
&\ln p((o_{1:T}, r_{1:T}, x_{1:T})|a_{1:T}) \\
=&\ln \left[\mathbb{E}_{q(s_{1:T},c_{1:T}|o_{1:T},a_{1:T})}\left[\frac{p((o_{1:T}, r_{1:T}, x_{1:T}), s_{1:T}, c_{1:T}|a_{1:T})}{q(s_{1:T}, c_{1:T}|o_{1:T}, a_{1:T})}\right]\right] \\
\geq&\mathbb{E}_{q(s_{1:T},c_{1:T}|o_{1:T},a_{1:T})}\left[\ln \frac{p((o_{1:T}, r_{1:T}, x_{1:T}), s_{1:T}, c_{1:T}|a_{1:T})}{q(s_{1:T}, c_{1:T}|o_{1:T}, a_{1:T})}\right] \\
=&\mathbb{E}_{q(s_{1:T},c_{1:T}|o_{1:T},a_{1:T})}\left[\ln \frac{\prod_t p(s_t|s_{t-1}, a_{t-1})p(c_t|c_{t-1})p(o_t|s_t, c_t)p(r_t|s_t)p(x_t|s_t)}{\prod_t q(c_t|o_{\leq t}, a_{<t})q(s_t|c_t, o_{\leq t}, a_{<t})}\right] \\
=&\mathbb{E}_{q(s_{1:T},c_{1:T}|o_{1:T},a_{1:T})}\left[\sum_{t=1}^{T}[\ln p(s_t|s_{t-1}, a_{t-1}) + \ln p(c_t|c_{t-1}) + \ln p(o_t|s_t, c_t) + \ln p(r_t|s_t) + \ln p(x_t|s_t)\right. \\
&\left. - \ln q(c_t|o_{\leq t}, a_{<t}) - \ln q(s_t|c_t, o_{\leq t}, a_{<t})]\right] \qquad (7) \\
=&\sum_{t=1}^{T}\left[\int \prod_{t'=1}^{T} q(c_{t'}|o_{\leq t'}, a_{<t'})q(s_{t'}|c_{t'}, o_{\leq t'}, a_{<t'})\,[\ln p(o_t|s_t, c_t) + \ln p(r_t|s_t) + \ln p(x_t|s_t)\right. \\
&\left. + (\ln p(s_t|s_{t-1}, a_{t-1}) - \ln q(s_t|c_t, o_{\leq t}, a_{<t})) + (\ln p(c_t|c_{t-1}) - \ln q(c_t|o_{\leq t}, a_{<t}))]\,\mathrm{d}s_{1:T}\mathrm{d}c_{1:T}\right] \\
=&\sum_{t=1}^{T}\left[\mathbb{E}_{s_t,c_t}[\ln p(o_t|s_t, c_t)] + \mathbb{E}_{s_t}[\ln p(r_t|s_t)] + \mathbb{E}_{s_t}[\ln p(x_t|s_t)] - \mathbb{E}_{c_{t-1}}[\mathrm{KL}[q(c_t|o_{\leq t}, a_{<t})\|p(c_t|c_{t-1})]]\right. \\
&\left. - \mathbb{E}_{c_t,s_{t-1}}[\mathrm{KL}\left[q(s_t|c_t, o_{\leq t}, a_{<t})\|p(s_t|s_{t-1}, a_{t-1})\right]]\right].
\end{aligned}
$$

By applying the history models in Eq. (3), we can ultimately obtain the $\mathcal{L}_{\mathrm{ELBO}}$ in Eq. (4).

A.2 Derivation of the upper bound for conditional mutual information

By incorporating the variational approximate distribution $q(s_t|h_t^c, h_t^s, o_t)$, the original conditional mutual information is given by:

$$
\begin{aligned}
I(s_t; c_t|h_t^c, h_t^s, o_t) =&\mathbb{E}_{o_t,s_t,c_t}\left[\ln \frac{p(s_t, c_t|h_t^c, h_t^s, o_t)}{p(s_t|h_t^c, h_t^s, o_t)p(c_t|h_t^c, h_t^s, o_t)}\right] \\
=&\mathbb{E}_{o_t,s_t,c_t}\left[\ln \frac{p(s_t|c_t, h_t^c, h_t^s, o_t)q(s_t|h_t^c, h_t^s, o_t)}{p(s_t|h_t^c, h_t^s, o_t)q(s_t|h_t^c, h_t^s, o_t)}\right] \\
=&\mathbb{E}_{o_t,s_t,c_t}\left[\ln \frac{p(s_t|c_t, h_t^c, h_t^s, o_t)}{q(s_t|h_t^c, h_t^s, o_t)}\right] - \mathbb{E}_{o_t,s_t,c_t}\left[\ln \frac{p(s_t|h_t^c, h_t^s, o_t)}{q(s_t|h_t^c, h_t^s, o_t)}\right] \qquad (8) \\
=&\mathbb{E}_{o_t,s_t,c_t}\left[\ln \frac{p(s_t|c_t, h_t^c, h_t^s, o_t)}{q(s_t|h_t^c, h_t^s, o_t)}\right] - \mathbb{E}_{o_t,s_t}\left[\ln \frac{p(s_t|h_t^c, h_t^s, o_t)}{q(s_t|h_t^c, h_t^s, o_t)}\right] \\
=&\mathbb{E}_{o_t,c_t}[\mathrm{KL}[p(s_t|c_t, h_t^c, h_t^s, o_t)\|q(s_t|h_t^c, h_t^s, o_t)]] - \mathbb{E}_{o_t}[\mathrm{KL}[p(s_t|h_t^c, h_t^s, o_t)\|q(s_t|h_t^c, h_t^s, o_t)] \\
\leq&\mathbb{E}_{o_t,c_t}[\mathrm{KL}[p(s_t|c_t, h_t^c, h_t^s, o_t)\|q(s_t|h_t^c, h_t^s, o_t)]].
\end{aligned}
$$

Then we can obtain the upper bound regularization loss in Eq. (5).

B Pseudo Code

The whole algorithm is shown in Algorithm 1. For brevity, we omit the episode continuation flag x_t.

Algorithm 1 CsDreamer

Initialize: Dataset $\mathcal{D}$ collected by random policy, the policy parameters θ, the critic parameters ψ, the variational estimator parameters ξ and the parameters ϕ in the CsRSSM world model
for training step $t_1 = 1...T_1$ **do**
  for update step $t_2 = 1...T_2$ **do**
    // `Dynamics learning`
    Sample minibatch $\{(o_t, a_t, r_t)\}_{t=k}^{k+L} \sim \mathcal{D}$
    Compute loss according to Eq. (6)
    Update CsRSSM world model parameters ϕ and the variational estimator parameters ξ
    // `Behavior learning`
    Infer the task-irrelevant information $c_t \sim q_\phi(c_t|h_t^c, h_t^s, o_t)$
    Infer the task-relevant information $s_t \sim q_\phi(s_t|h_t^c, h_t^s, c_t, o_t)$
    Imagine trajectories $\{(s_t, a_t, r_t)\}_{t=k}^{k+H}$ using $p_\phi(\hat{s}_t|h_t^s), p_\phi(\hat{r}_t|h_t^s, s_t)$ and the policy π
    Update the policy parameters θ and the critic parameters ψ using the imagined trajectories
  end for
  for rollout step $t_3 = 1...T_3$ **do**
    Infer the task-irrelevant information $c_t \sim q_\phi(c_t|h_t^c, h_t^s, o_t)$
    Infer the task-relevant information $s_t \sim q_\phi(s_t|h_t^c, h_t^s, c_t, o_t)$
    Sample action from exploration policy $a_t \sim \pi_{\text{exp}}(a_t|h_t^s, s_t)$
    $r_t, o_{t+1} \leftarrow$ `env.step(`a_t`)`
  end for
  Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, r_t)\}$
end for

C Details about the Benchmark

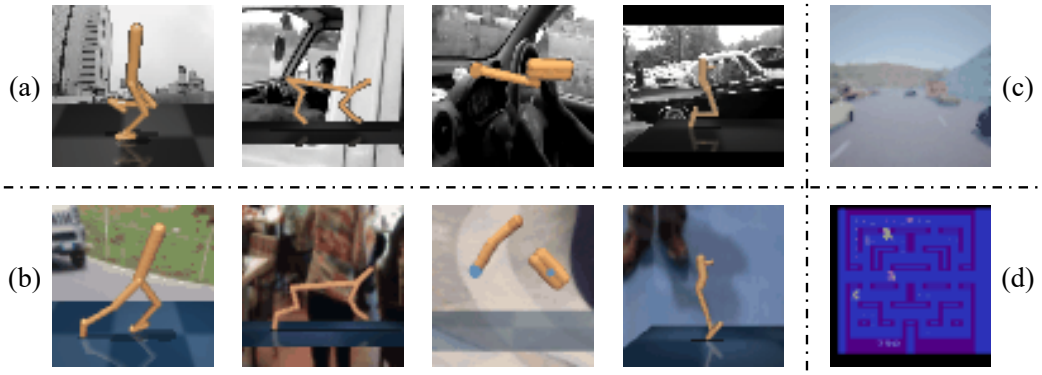


Figure 9: Example observations of the benchmarks. (**a**) The observations in DMC with gray videos in the background and the four tasks from left to right are *Walker Run*, *Cheetah Run*, *Finger Spin* and *Hopper Hop*. (**b**) The observations in DMC with colourful videos in the background and the four tasks from left to right are *Walker Run*, *Cheetah Run*, *Finger Spin* and *Hopper Hop*. (**c**) The observation in CARLA simulator. (**d**) The observation in the game *Alien*, which is one of the 26 games in the Atari 100K benchmark.

DMC with Complex Backgrounds. We introduce the natural video background as the noisy distractor for the widely used robotic locomotion benchmark, DeepMind Control Suite [45]. We select four tasks for evaluation. The *Walker Run* task assesses the stability and coordination of bipedal locomotion, *Cheetah Run* examines high-speed movement and agility, *Finger Spin* evaluates

precise motor control and object manipulation capabilities, and *Hopper Hop* tests dynamic balance and energy management in single-leg hopping. We choose *Walker Run* and *Hopper Hop* instead of *Walker Walk* and *Hopper Stand* for more challenging evaluation. For each task, we adopt two types of noisy distractors. We first introduce the natural video from the Kinetics dataset [46]. We only use the videos in 'driving car' class and set it to gray as that in [21, 10] (Figure 9 (**a**)). Then we utilize the colourful videos of the DAVIS 2017 dataset [47] as that in [48]. We utilize all the 90 train-val videos and adopt the dynamic setting, where the video plays forwards or backwards until the last or first frame is reached at which point the playing direction is reversed, thereby the background motion is always smooth and without 'cuts' (Figure 9 (**b**)). The height and the width of the input observation image are 64×64.

CARLA. To evaluate on a more real-world control system, we use the CARLA simulator [49] to conduct photo-realistic visual observations. The agent's goal is to drive as far as possible along CARLA's Town04's highway in 1000 timesteps in a first-person way without colliding with other moving vehicles or barriers as the setting in DBC [21]. To increase the task difficulty, we make two modifications as that in Iso-Dream setting [12]. We use one camera which obtains images of 64×64 pixels instead of five to limit the field of view and we include 30 other moving vehicles or obstacles instead of 20 to increase the likelihood of collisions. The example observation is shown in Figure 9 (**c**).

Atari 100K Benchmark. The Atari 100K benchmark [50] comprises 26 distinct Atari games. Within this benchmark, an agent is permitted 100K interaction steps for each game environment. Due to a frameskip setting of 4, this translates to 400K frames. This interaction volume is roughly equivalent to about two hours of real-time gameplay. The game environments within this benchmark are notably complex, making it a practical testbed for evaluating the algorithms' robustness and data efficiency. We keep all implementation details the same as Dreamer [17], and the example observation is shown in Figure 9 (**d**).

D Implementation Details

D.1 Base Method

CsDreamer is implemented based on the classic model-based reinforcement learning method, Dreamer [18, 19, 17]. Given that Dreamer has multiple versions, we first evaluate DreamerV2 [19] and DreamerV3 [17] [2]on DMC using gray natural videos as background.

The performance results are in Figure 10, both DreamerV2 and DreamerV3 demonstrate overall similar performance across the four tasks. Specifically, DreamerV3 shows a slight advantage in the *Finger Spin + GB* tasks, while DreamerV2 performs marginally better in the *Walker Run + GB* and *Hopper Hop + GB* tasks. They have comparable performance in the *Cheetah Run + GB* task. Overall, the performance differences between the two versions are minimal. The minimal performance differences between DreamerV2 and DreamerV3 can likely be attributed to several factors. Firstly, both models share a significant degree of architectural and methodological overlap. Secondly, some of the improvements in DreamerV3 are primarily designed to enhance adaptability across various domains and tasks. However, these enhancements may compromise the model's fundamental capabilities. This trade-off means that while DreamerV3 can perform effectively in a broader range of scenarios, it might not significantly outperform DreamerV2, resulting in similar performance across the evaluated tasks. Considering that DreamerV3 has a much smaller world model loss (on the order of tens) than DreamerV2 (over ten thousand) and in our experiments $\mathcal{L}_{\mathrm{MI}}$ is typically within single digits, we use DreamerV3 as the base method so that $\mathcal{L}_{\mathrm{CsRSSM}}$ and $\mathcal{L}_{\mathrm{MI}}$ have a similar order of magnitude to avoid problems. Unless otherwise specified, in the experiments, **Dreamer** refers to DreamerV3, and **CsDreamer** is based on DreamerV3.

[2]In this paper, we utilize the 'S' size model for DreamerV3 in https://arxiv.org/pdf/2301.04104v1.

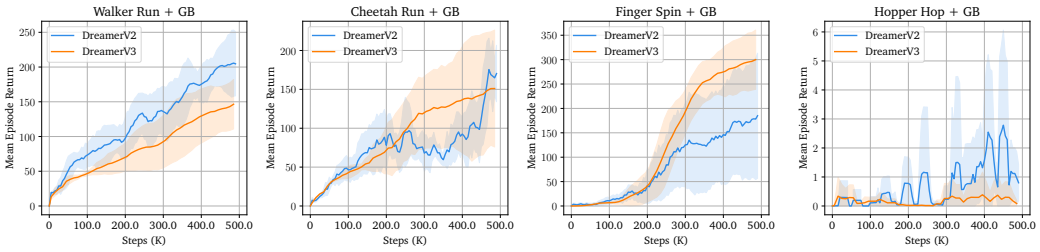


Figure 10: Performance of DreamerV2 and DreamerV3 on DMC using the gray natural videos as background.

D.2 Key Component Implementation

The RSSM world model in Dreamer implements the KL divergence as two separate components in different stop-gradient operator places and loss scales [17]. Similarly, the two KL divergences in Eq. (4) are each composed of two distinct loss terms. Specifically, we have

$$\begin{aligned}\mathbb{E}[\mathrm{KL}[q_\phi(c_t|h_t^c,h_t^s,o_t)\|p_\phi(c_t|h_t^c)]] &= \beta^c_{\mathrm{dyn}}\mathcal{L}^c_{\mathrm{dyn}}(\phi)+\beta^c_{\mathrm{rep}}\mathcal{L}^c_{\mathrm{rep}}(\phi)\\ \mathbb{E}[\mathrm{KL}\left[q_\phi(s_t|h_t^c,h_t^s,c_t,o_t)\|p_\phi(s_t|h_t^s)\right]] &= \beta^s_{\mathrm{dyn}}\mathcal{L}^s_{\mathrm{dyn}}(\phi)+\beta^s_{\mathrm{rep}}\mathcal{L}^s_{\mathrm{rep}}(\phi),\end{aligned} \tag{9}$$

where

$$\begin{aligned}\mathcal{L}^c_{\mathrm{dyn}}(\phi) &= \max\left(1,\mathrm{KL}\left[\mathrm{sg}\left(q_\phi(c_t|h_t^c,h_t^s,o_t)\right)\|p_\phi(c_t|h_t^c)\right]\right)\\ \mathcal{L}^c_{\mathrm{rep}}(\phi) &= \max\left(1,\mathrm{KL}\left[q_\phi(c_t|h_t^c,h_t^s,o_t)\|\,\mathrm{sg}\left(p_\phi(c_t|h_t^c)\right)\right]\right)\\ \mathcal{L}^s_{\mathrm{dyn}}(\phi) &= \max\left(1,\mathrm{KL}\left[\mathrm{sg}\left(q_\phi(s_t|h_t^c,h_t^s,c_t,o_t)\right)\|p_\phi(s_t|h_t^s)\right]\right)\\ \mathcal{L}^s_{\mathrm{rep}}(\phi) &= \max\left(1,\mathrm{KL}\left[q_\phi(s_t|h_t^c,h_t^s,c_t,o_t)\|\,\mathrm{sg}\left(p_\phi(s_t|h_t^s)\right)\right]\right),\end{aligned} \tag{10}$$

and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. In the CsRSSM world model, the task-relevant component utilizes the same network architecture as Dreamer. The primary distinction arises during the computation of the observation posterior, where both the observation o_t and the inferred latent variable c_t are concatenated and provided as input, rather than using only the observation o_t as in Dreamer. Additionally, when reconstructing the observation, the model concatenates the inferred c_t with the variable s_t to reconstruct o_t jointly. The task-irrelevant component employs a network structure similar to that of Dreamer but with some modifications. It does not take actions as input, ensuring that it remains unaffected by action-related information. Moreover, it does not predict task-relevant variables such as rewards, thereby focusing solely on modeling the noise distractors independent of the task-relevant objectives.

For the regularization loss $\mathcal{L}_{\mathrm{MI}}$, we use a variational estimator q_ξ for the variational approximate distribution $q(s_t|h_t^c,h_t^s,o_t)$. The variational estimator takes the history (h_t^c,h_t^s), and the embedding of o_t as the input. The gradients from these variables will be blocked from backpropagation during the training of the variational estimator, i.e., $q_\xi(s_t|\,\mathrm{sg}(h_t^c,h_t^s,o_t))$. This ensures that the learning of the variational estimator does not influence the parameters of the preceding models. Then, we utilize the part network of c_t to get the embedding of (h_t^c,h_t^s,o_t), and we use the embedding to get the variational approximate distribution after a linear layer.

D.3 Module Ablation Implementation

Here, we detail the implementation of the two ablation methods described in Section 5.3. **CsDreamer w/o MI** is implemented by simply removing the decoupling regularization from **CsDreamer**. **CsDreamer w/o CD** adopts the CsRSSM framework with some modifications. Specifically, since **CsDreamer w/o CD** does not account for the conditional dependence between s_t and c_t given o_t in collider structures, we adjust its encoders: the task-irrelevant encoder is defined as $q_\phi(c_t|h_t^c,o_t)$ and the task-relevant encoder is defined as $q_\phi(s_t|h_t^s,o_t)$. The world model is then trained using $\mathcal{L}_{\mathrm{ELBO}}$. Because **CsDreamer w/o CD** ignores all conditional dependence, the $\mathcal{L}_{\mathrm{MI}}$ term is omitted.

D.4 Hyperparameters and Time Cost

Table 1 presents the primary hyperparameters of CsDreamer. Since the behavior policy relies solely on the feature of s_t, and the hyperparameters for s_t in CsRSSM closely resemble those of the latent variables in RSSM of DreamerV3, we adopt the same hyperparameters for the behavior policy as in DreamerV3. The experiments are mainly conducted on NVIDIA RTX 4090 GPUs. With each GPU, we are able to train each environment at a rate of approximately 24K timesteps per hour.

Table 1: Hyperparameters for CsDreamer

Hyperparameter	**Value**
Action Repeat	4 for CARLA and Atari, and 2 for others
λ	10.0 for Hopper Hop+GB, 0.2 for CARLA and Atari, and 1.0 for others
β^s_{dyn}	0.5
β^s_{rep}	0.1
β^c_{dyn}	0.5
β^c_{rep}	0.1
Discrete latent dimensions of s_t	32
Discrete latent classes of s_t	32
Discrete latent dimensions of c_t	16 for CARLA, 8 for Atari, and 32 for others
Discrete latent classes of c_t	16 for CARLA, 8 for Atari, and 32 for others
GRU recurrent units of s_t	512
GRU recurrent units of c_t	256 for CARLA, 128 for Atari, and 512 for others
Dense hidden units of s_t	512
Dense hidden units of c_t	512
MLP layers	2

E Additional Experiment Results

E.1 Performance Scores

In Table 2, we summarize the final mean episode return and their corresponding standard deviations for various model-based RL methods across multiple environments in the main text, evaluated using four distinct random seeds, each associated with 10 evaluation episodes. The results consistently show that CsDreamer outperforms the other methods in the majority of environments, thereby demonstrating the superior effectiveness of our proposed approach.

Table 2: Final performance across model-based RL methods in different environments.

Environment	**CsDreamer (Ours)**	**Dreamer**	**TIA**	**Denoised MDP**	**Iso-Dream**
Walker Run + GB	$\mathbf{533 \pm 98}$	162 ± 32	293 ± 129	117 ± 81	131 ± 23
Cheetah Run + GB	$\mathbf{547 \pm 159}$	171 ± 84	432 ± 172	215 ± 191	109 ± 12
Finger Spin + GB	$\mathbf{408 \pm 45}$	287 ± 54	123 ± 201	331 ± 146	$\mathbf{415 \pm 118}$
Hopper Hop + GB	$\mathbf{75 \pm 58}$	0 ± 0	$\mathbf{50 \pm 48}$	13 ± 15	0 ± 0
Walker Run + CB	$\mathbf{678 \pm 74}$	474 ± 71	588 ± 135	85 ± 18	211 ± 95
Cheetah Run + CB	$\mathbf{821 \pm 58}$	549 ± 126	758 ± 80	392 ± 195	6 ± 4
Finger Spin + CB	$\mathbf{576 \pm 241}$	466 ± 130	490 ± 136	460 ± 123	4 ± 8
Hopper Hop + CB	$\mathbf{70 \pm 52}$	23 ± 28	42 ± 40	21 ± 19	0 ± 0
CARLA	$\mathbf{235 \pm 118}$	$\mathbf{202 \pm 144}$	44 ± 119	21 ± 28	24 ± 25

E.2 Performance on Atari 100K Benchmark

In order to assess the denoising capabilities of CsDreamer under highly complex visual inputs, we conduct experiments on the Atari 100K benchmark, and the results are presented in Figure 11 and Table 3. For each game, we use at least five seeds. During our experiments, we encounter a random seed that produces performance approximately 20 times higher than the current baseline, which we

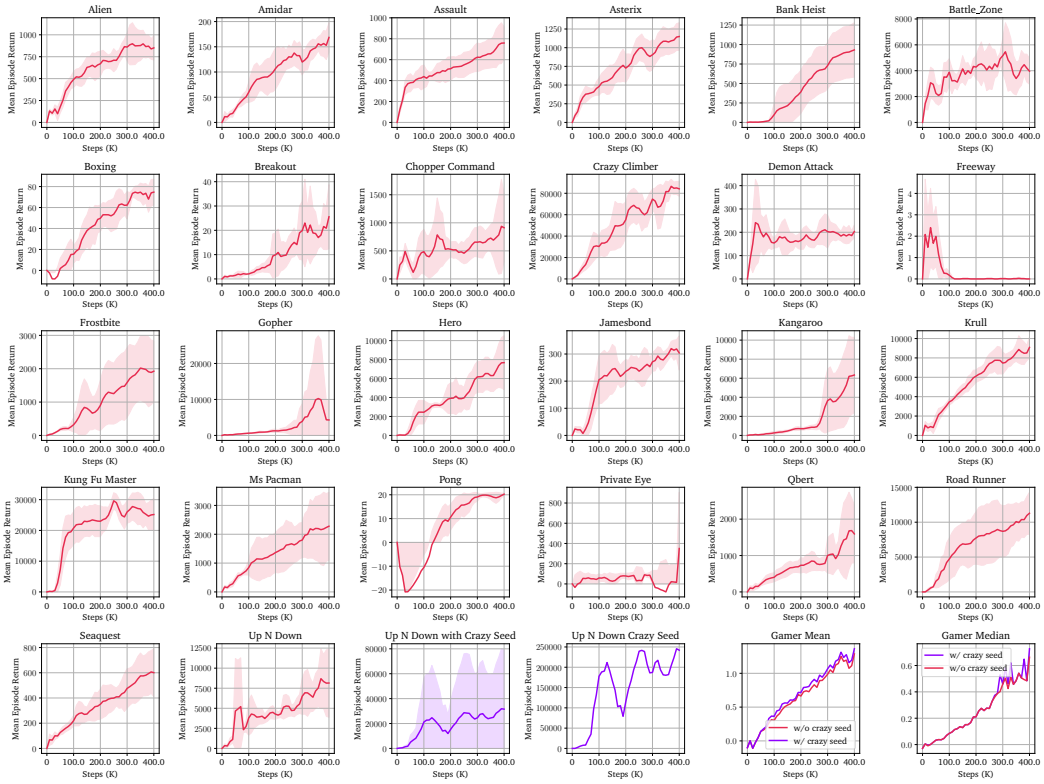


Figure 11: CsDreamer performance on Atari 100K Benchmark. In the *Up N Down* environment, we encounter a random seed that yields performance approximately 20 times higher than the current baseline, which we call the 'crazy seed'. We use 10 seeds in this environment to ensure a more accurate evaluation. The *Up N Down* results reflect the performance of the 9 seeds, excluding the crazy seed, while *Up N Down with Crazy Seed* presents the outcomes incorporating all 10 seeds.

call the 'crazy seed'. To facilitate a more robust and accurate evaluation, we employ a total of 10 seeds in this environment.

In Figure 11, the results labeled *Up N Down* reflect the performance obtained from nine seeds (excluding the crazy seed), whereas *Up N Down with Crazy Seed* encompasses outcomes derived from all 10 seeds. In Table 3, values within parentheses indicate results achieved using the crazy seed, while results outside the parentheses correspond to evaluations without its inclusion. The Dreamer results in Table 3 are sourced directly from the official DreamerV3 paper [17] [3]. The experimental outcomes clearly indicate that CsDreamer outperforms the baseline approach. This improvement demonstrates that leveraging conditional dependencies enables CsDreamer to effectively denoise complex visual inputs, thereby enhancing overall task performance.

E.3 Performance on Standard DMC

We conduct experiments using the standard DeepMind Control (DMC) suite to evaluate performance on clean benchmarks. Figure 12 illustrates performance across four tasks. As depicted, CsDreamer generally achieves performance on par with the original Dreamer algorithm across all four evaluated tasks. These results indicate that the CsDreamer algorithm effectively handles noisy scenarios and maintains strong performance in noise-free environments.

[3] In this paper, we utilize the data in https://arxiv.org/pdf/2301.04104v1.

Table 3: CsDreamer's performance on the Atari 100K Benchmark. Values in parentheses indicate the use of the 'crazy seed'.

Game	Random	Human	Dreamer (official)	CsDreamer
Alien	228	7128	**959**	888 ± 234
Amidar	6	1720	139	$\mathbf{184 \pm 43}$
Assault	222	742	706	$\mathbf{748 \pm 262}$
Asterix	210	8503	932	$\mathbf{1114 \pm 266}$
Bank Heist	14	753	649	$\mathbf{946 \pm 465}$
Battle Zone	2360	37188	**12250**	3960 ± 2537
Boxing	0	12	**78**	$\mathbf{80 \pm 13}$
Breakout	2	30	**31**	$\mathbf{31 \pm 51}$
Chopper Com.	811	7388	420	$\mathbf{864 \pm 480}$
Crazy Climber	10780	35829	**97190**	90196 ± 22281
Demon Attack	152	1971	**303**	207 ± 124
Freeway	0	30	0	0 ± 0
Frostbite	65	4335	909	$\mathbf{2177 \pm 995}$
Gopher	258	2412	3730	$\mathbf{7771 \pm 18563}$
Hero	1027	30826	**11161**	8124 ± 3345
James Bond	29	303	**445**	292 ± 147
Kangaroo	52	3035	4098	$\mathbf{6590 \pm 4797}$
Krull	1598	2666	7782	$\mathbf{9636 \pm 3531}$
Kung Fu Master	258	22736	21420	$\mathbf{24847 \pm 8928}$
Ms Pacman	307	6952	1327	$\mathbf{2170 \pm 1306}$
Pong	-21	15	18	$\mathbf{20 \pm 1}$
Private Eye	25	69571	882	$\mathbf{977 \pm 1813}$
Qbert	164	13455	**3405**	1050 ± 692
Road Runner	12	7845	**15565**	11980 ± 3946
Seaquest	68	42055	**618**	578 ± 252
Up N Down	533	11693	7667	$\mathbf{9800 \pm 13122}(\mathbf{32050 \pm 69863})$
Human Mean	0%	100%	112%	$\mathbf{129\%}(\mathbf{136\%})$
Human Median	0%	100%	49%	$\mathbf{66\%}(\mathbf{73\%})$

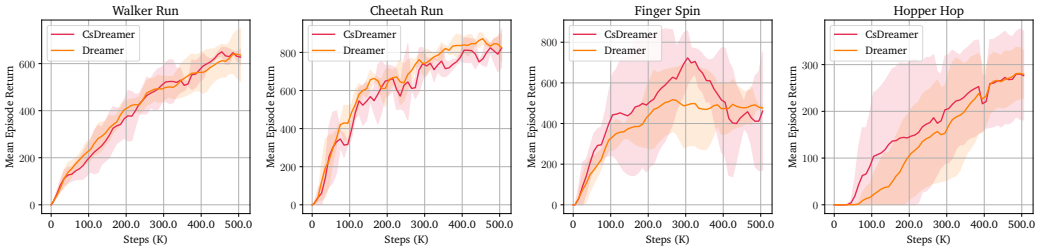


Figure 12: Performance on Standard DMC.

E.4 Performance on DMC + GB Using Train-Eval Split

In the experiments above on **DMC + GB**, we follow previous work by training and testing using the same dataset as background. Specifically, we utilize 16 car videos beginning with the letter 'A' from the Kinetics-400 dataset, mentioned in TIA [10]. We download the complete' driving car' class using the GitHub repository[4] to ascertain whether CsDreamer's robustness is a result of merely memorizing background information or if it genuinely possesses an intrinsic denoising capability. Due to the reasons mentioned in the repository, we succeed in obtaining 641 videos. Then we alphabetically split the dataset into a training dataset (512 videos) and an evaluation dataset (129 videos) using an $80:20$ ratio. We choose the best-performing baseline TIA in **DMC + GB** as the baseline, and the experimental results are presented in Figure 13. Compared to the results in **DMC + GB** above, the outcomes for *Walker Run*, *Cheetah Run*, and *Finger Spin* are slightly inferior, while the outcome for *Hopper Hop* is somewhat superior. The results depicted in the figure demonstrate that CsDreamer outperforms TIA across all tasks. This confirms that our approach effectively denoises rather than merely memorizing backgrounds.

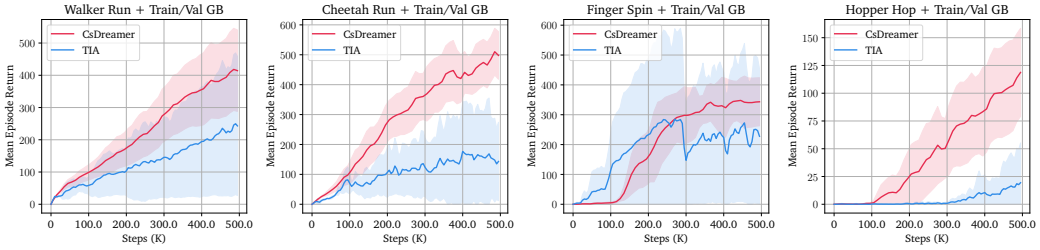


Figure 13: Performance on the train-eval dataset, with 641 driving car videos in total, 512 videos for the training dataset, and 129 videos for the evaluation dataset.

E.5 Additional Ablation Results

We conduct more module ablation studies on the DMC benchmark using colorful natural videos as the background. Similar to that in Section 5.3, we compare the performance of the baseline **Dreamer**, **CsDreamer w/o MI** (CsDreamer with CsRSSM framework without the conditional mutual information-based regularization), **CsDreamer w/o CD** (CsDreamer with CsRSSM framework without the conditional dependence) and **CsDreamer**. Shown in Figure 14, the results are consistent with those in Section 5.3. The comparison between **Dreamer** and **CsDreamer w/o MI** reveals that the CsRSSM can significantly boost performance in visual input with complex distractors by utilizing the conditional dependence. By introducing the decoupling regularization, **CsDreamer** consistently outperforms baselines on all tasks, achieving the highest performance. This ablation study also demonstrates that CsRSSM with the decoupling information regularization significantly enhances learning efficiency and task performance.

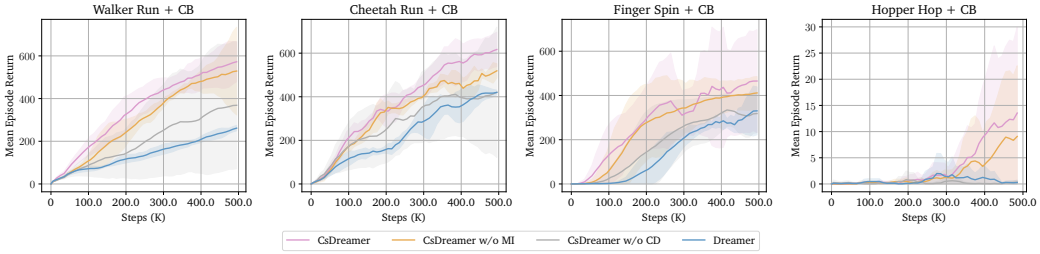


Figure 14: Ablation Performance on DMC using colorful background.

We also examine the role of regularization in the *Walker Run + GB* as in Section 5.3. As illustrated in Figure 15, the results echo those from Section 5.3: as the regularization parameter λ increases,

[4]We use the Github repository in https://github.com/Showmax/kinetics-downloader.

the upper bound of mutual information progressively decreases, while the mean return follows a non-monotonic trend—initially rising and then falling. All these experiments validate the analysis presented in Section 4.3.

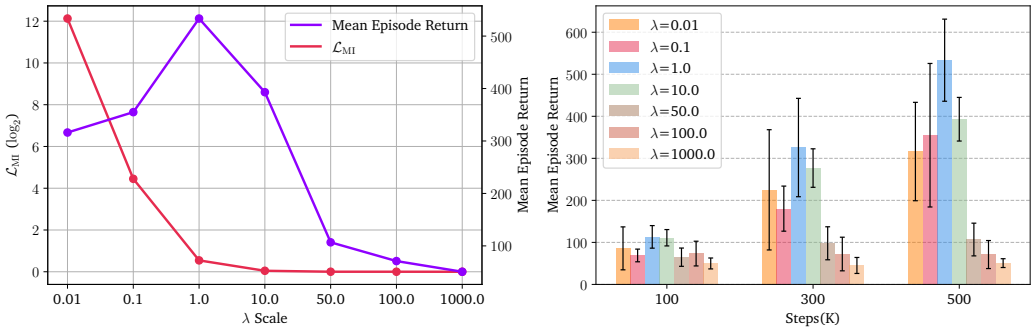


Figure 15: Ablation study across different λ on *Walker Run* using gray driving car as background.

E.6 Correlation analysis

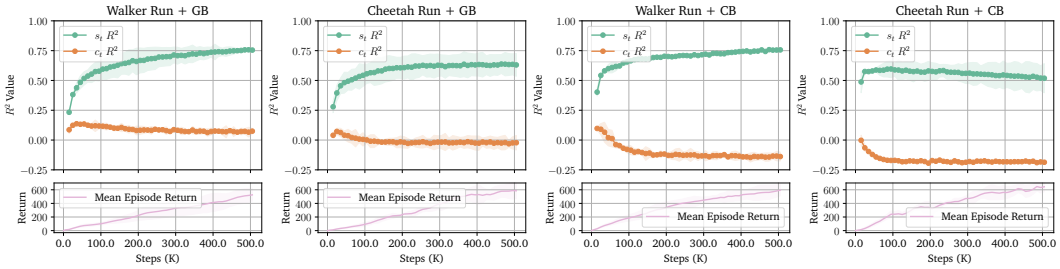


Figure 16: Coefficient of determination (R^2) on DMC tasks with background distractors.

In this section, we assess the coefficient of determination (R^2) within our framework. We find that the distribution of the test data significantly influences the R^2 values. Specifically, evaluating R^2 using the data sampled from the current policy may result in artificially high values. To obtain a more reliable assessment, we use the sampled data from the final phase of a prior training session as our dataset because, at this stage, the policy has sufficiently learned to perform the task effectively, leading to the observation continuously undergoing noticeable changes. In detail, we utilize the last 20 sampled episodes from a previous training process, with each episode comprising 1,000 timesteps (derived from 500 data points with an action repeat factor of 2). Each data point has one observation and one corresponding agent state. We utilize the observation in these episodes to yield the features for variables s_t and c_t. We use the feature of s_t and the feature of c_t as the independent variables separately to predict the agent's real state. To mitigate potential interference from newly introduced training parameters, we utilize a parameter-free k-nearest neighbors (KNN) regressor for prediction. The results are shown in Figure 16. The R^2 value for s_t rapidly increases in the early stages and stabilizes, demonstrating a strong and consistent correlation. In contrast, the R^2 value for c_t decreases and remains low throughout the experiments. The results highlight that s_t exhibits a much stronger correlation to the real agent state than c_t, suggesting that s_t can effectively capture the task-relevant information.

E.7 Additional Results Using Transformer-Based Framework

To clarify whether the proposed collider-structure approach is restricted to RSSM-based designs or if it is compatible with transformer-style sequence encoders, we extended our methodology to TransDreamer [51]. TransDreamer utilizes a Transformer State-Space Model (TSSM) to capture long-term dependencies, replacing the RSSM of Dreamer. We integrate our approach into this baseline, resulting in the Collider-structure Transformer State-Space Model (CsTSSM) and the corresponding CsTransDreamer. A primary adaptation challenge stems from TransDreamer's architecture, which

is optimized for parallel training. To achieve this, TSSM employs a *Myopic Representation Model*, where the posterior inference $q(z_t|o_t)$ is independent of the deterministic history state h_t. CsTSSM adapts the collider framework within this constraint.

The key modifications in CsTSSM are as follows:

- **Dual Transformer Dynamics**, where we replace the single dynamic model with two independent Transformer networks. One models the task-relevant dynamics, generating h_t^s from past relevant states and actions; the other models the task-irrelevant dynamics, generating h_t^c solely from past task-irrelevant states.
- **Parallel Collider Inference**, where we implement the collider-structure inference $p_\phi\left(c_t, s_t \mid o_t\right) = p_\phi\left(c_t \mid o_t\right) p_\phi\left(s_t \mid c_t, o_t\right)$. Crucially, because this inference path does not depend on the history states, the entire sequence can be encoded in parallel across the time dimension before the Transformer dynamics are computed, preserving the efficiency of the architecture.
- **Decoupling Regularization**, where we adapt the decoupling regularization loss to this architecture. We introduce the variational estimator $q_\xi\left(s_t \mid o_t\right)$ to approximate the task-relevant distribution without conditioning on c_t, thereby balancing the conditional dependence within the parallel framework.

Since TransDreamer open-sources its code for the Atari benchmark, we conduct experiments on the first three environments of the Atari 100K benchmark (shown in Figure 17). The results demonstrate that the benefits of leveraging conditional dependence via the collider structure are architecture-agnostic. The proposed methodology is not limited to RSSM-based designs and can effectively enhance transformer-based world models in environments with complex noise interference.

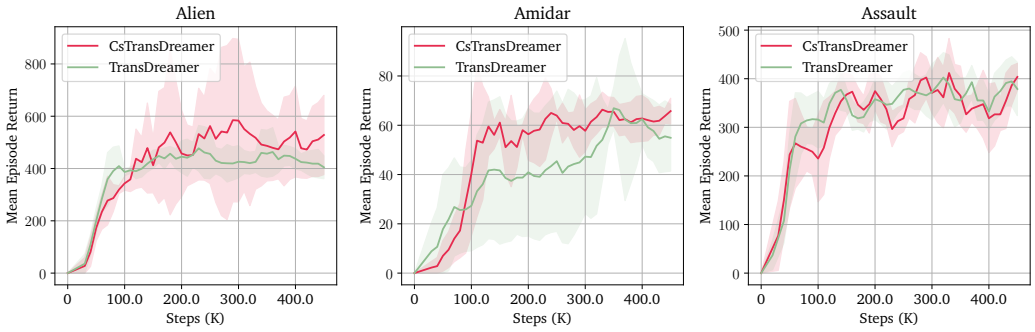


Figure 17: Transformer-based results on three Atari environments.

E.8 Additional Reconstruction Visualization

We extend the reconstruction visualization to additional environments following the configurations presented in Section 5.4, yielding consistent results (Figure 18-25). Furthermore, in the complex autonomous driving scenario within CARLA (Figure 18), we find that the reconstruction of task-irrelevant variables excluded nearby blue and red vehicles (as shown in the third row). In contrast, the reconstruction of task-relevant variables retains these vehicles, as their proximity influences the decision-making of the autonomous vehicle. These experiments the interpret semantic information extracted by the latent variables and qualitatively demonstrate that our method can effectively extract task-relevant information from noisy observations in complex environments. We also conduct the reconstruction visualization for the train-eval-split experiments in Appendix E.4. In each figure (Figure 26-29), the fourth and fifth rows exhibit a heavily blurred background while maintaining clear task-relevant details. These visualizations demonstrate that CsDreamer effectively learns to extract task-relevant information from noisy observations rather than memorizing the background information, thereby accounting for its superior performance.

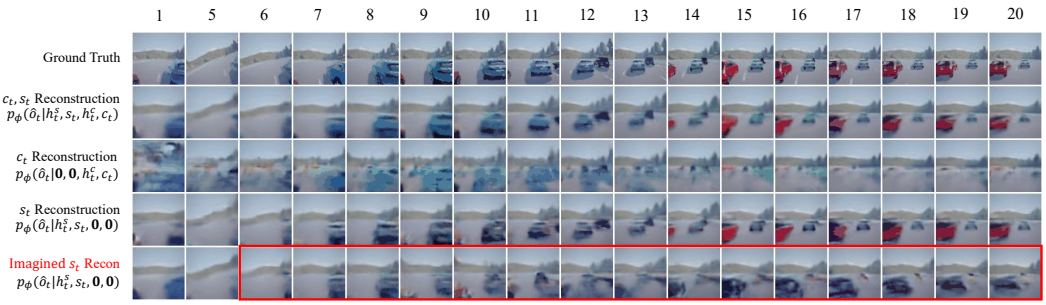


Figure 18: Reconstruction visualization of CARLA

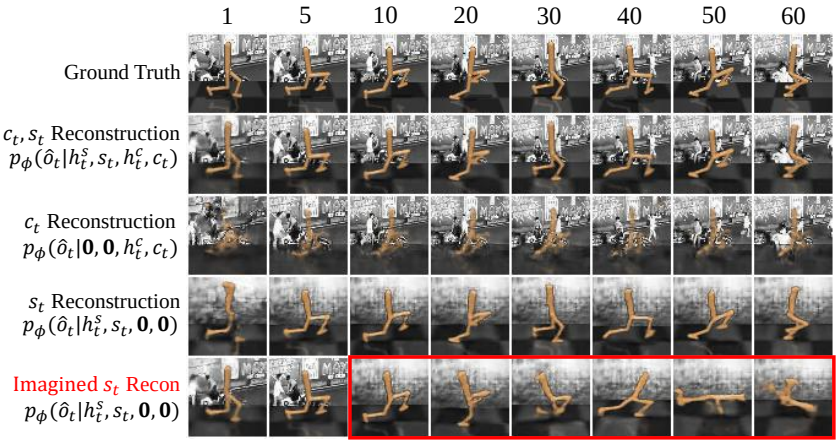


Figure 19: Reconstruction visualization of *Walker Run* using gray videos as background.

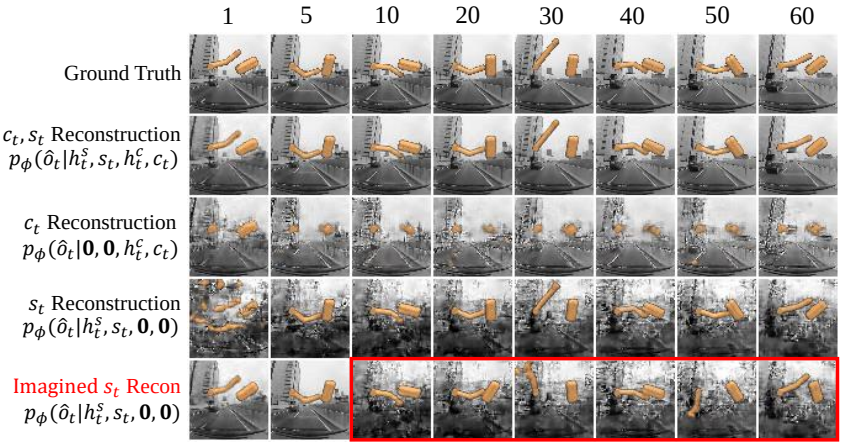


Figure 20: Reconstruction visualization of *Finger Spin* using gray videos as background.

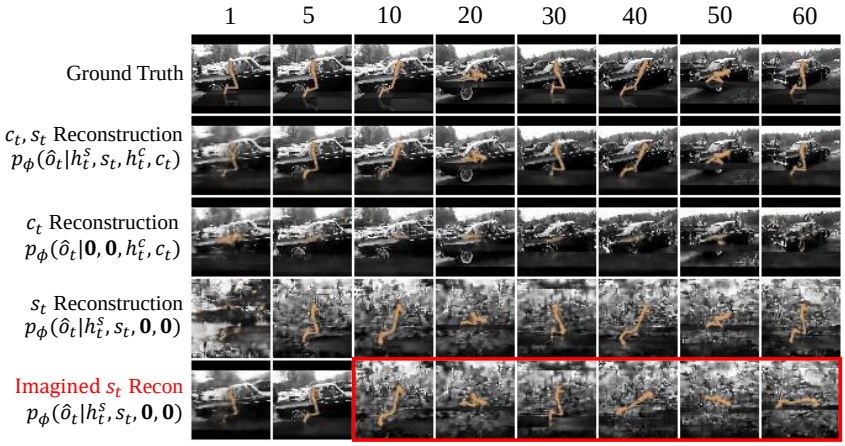


Figure 21: Reconstruction visualization of *Hopper Hop* using gray videos as background.

Figure 22: Reconstruction visualization of *Cheetah Run* using colorful videos as background.

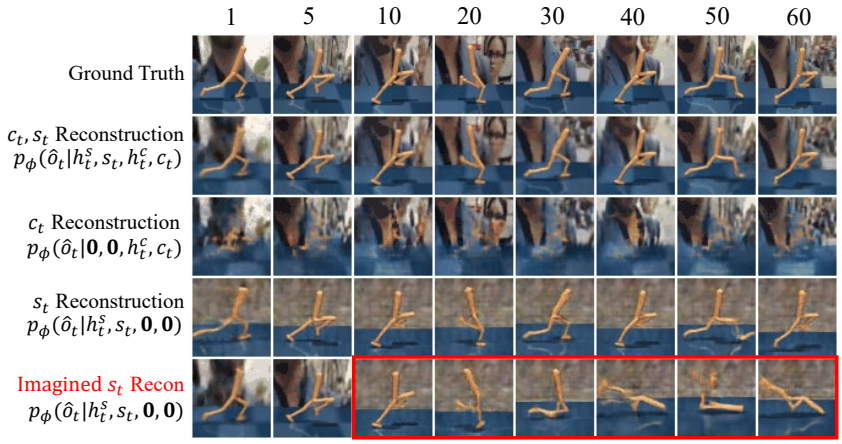


Figure 23: Reconstruction visualization of *Walker Run* using colorful videos as background.

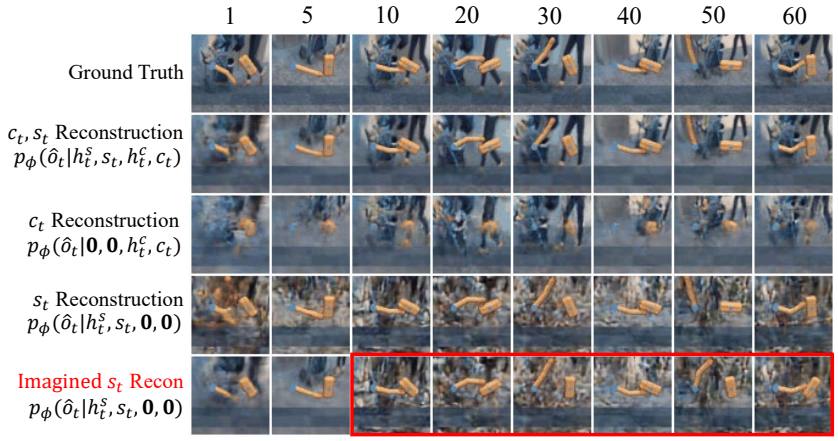


Figure 24: Reconstruction visualization of *Finger Spin* using colorful videos as background.

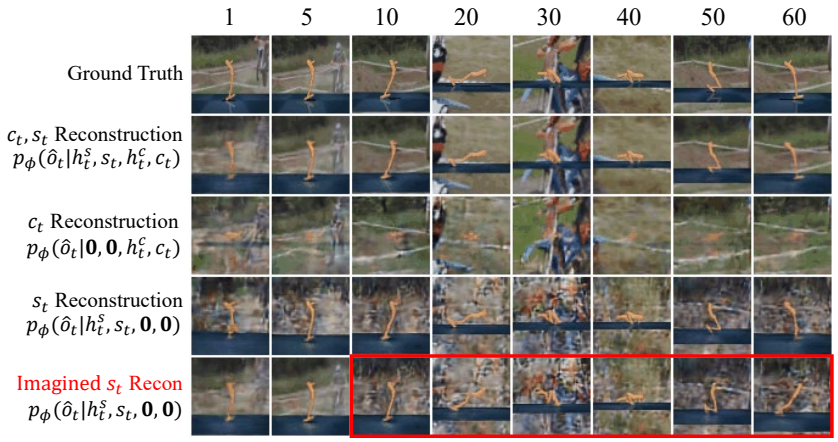


Figure 25: Reconstruction visualization of *Hopper Hop* using colorful videos as background.

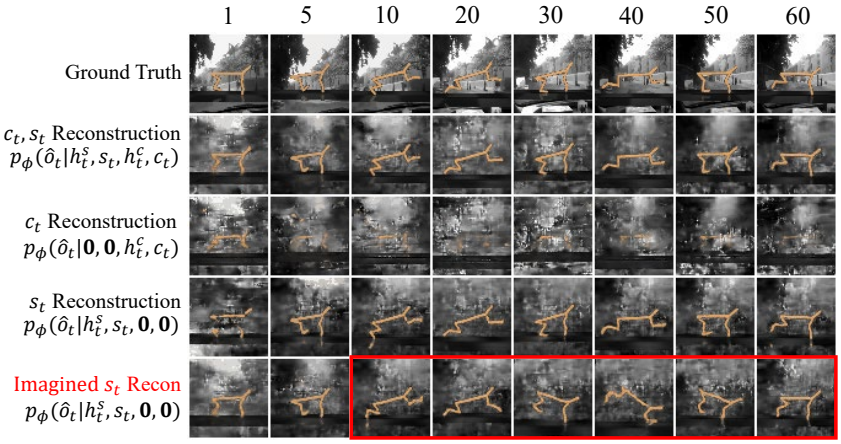


Figure 26: Reconstruction visualization of *Cheetah Run* using train-eval gray videos as background.

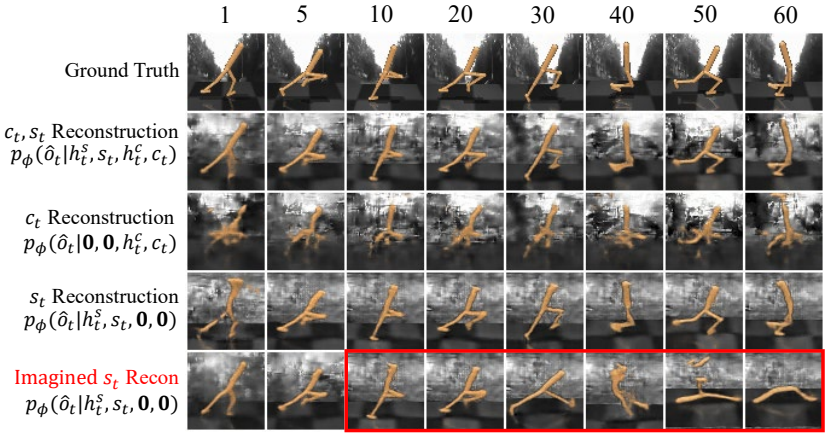


Figure 27: Reconstruction visualization of *Walker Run* using train-eval gray videos as background.

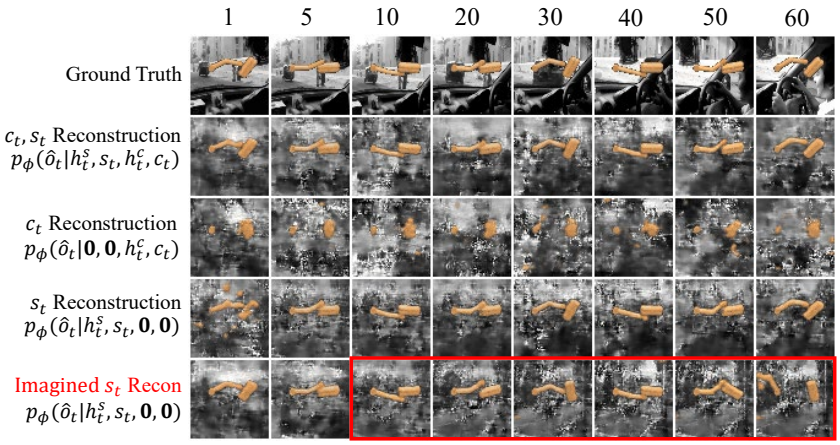


Figure 28: Reconstruction visualization of *Finger Spin* using train-eval gray videos as background.

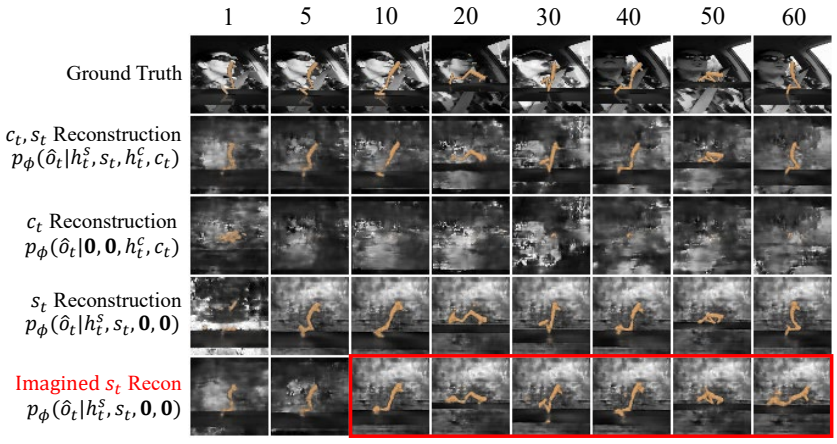


Figure 29: Reconstruction visualization of *Hopper Hop* using train-eval gray videos as background.