Appendix for “Learning World Models with Identifiable Factorization"

Anonymous Author(s)
Affiliation
Address
email

Appendix organization:

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

S1 Theoretical Proofs

S1.1 Proof of Proposition 1

Proposition 1 shows that s_t^r, which has directed paths to $r_{t+\tau}$ (for $\tau \geq 0$), is minimally sufficient for policy learning that aims to maximize the future reward and can be characterized by conditional dependence with the cumulative reward variable R_t.

Proposition 1. *Under the assumption that the graphical representation, corresponding to the environment model, is Markov and faithful to the measured data, $s_t^r \subseteq \mathbf{s_t}$ is a minimal subset of state dimensions that are sufficient for policy learning, and $s_{i,t} \in s_t^r$ if and only if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$.*

We first give the definitions of the Markov condition and the faithfulness assumption, which will be used in the proof.

Definition 1 (Global Markov Condition [1, 2])**.** *The distribution p over a set of variables $\mathbf{V}$ satisfies the global Markov property on graph G if for any partition (A, B, C) such that if B d-separates A from C, then $p(A, C|B) = p(A|B)p(C|B)$.*

Definition 2 (Faithfulness Assumption [1, 2])**.** *There are no independencies between variables that are not entailed by the Markov Condition in the graph.*

Below, we give the proof of Proposition 1.

Proof. The proof contains the following three steps.

- In step 1, we show that a state dimension $s_{i,t}$ is in s_t^r, that is, it has a directed path to $r_{t+\tau}$, if and only if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$.
- In step 2, we show that for $s_{i,t}$ with $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$, if and only if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$.
- In step 3, we show that s_t^r are minimally sufficient for policy learning.

Step 1: We first show that if a state dimension $s_{i,t}$ is in s_t^r, then $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$.

We prove it by contradiction. Suppose that $s_{i,t}$ is independent of R_t given a_{t-1} and $\mathbf{s}_{t-1}$. Then according to the faithfulness assumption, we can see from the graph that $s_{i,t}$ does not have a directed path to $r_{t+\tau}$, which contradicts the assumption, because, otherwise, a_{t-1} and $\mathbf{s}_{t-1}$ cannot break the paths between $s_{i,t}$ and R_t which leads to the dependence.

We next show that if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$, then $s_{i,t} \in s_t^r$.

Similarly, by contradiction suppose that $s_{i,t}$ does not have a directed path to $r_{t+\tau}$. From the graph, it is easy to see that a_{t-1} and $\mathbf{s}_{t-1}$ must d-separate the path between $s_{i,t}$ and R_t. According to the Markov assumption, $s_{i,t}$ is independent of R_t given a_{t-1} and $\mathbf{s}_{t-1}$, which contradicts to the assumption. Since we have a contradiction, it must be that $s_{i,t}$ has a directed path to $r_{t+\tau}$, i.e. $s_{i,t} \in s_t^r$.

Step 2: In step 1, we have shown that $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$, if and only if it has a directed path to $r_{t+\tau}$. From the graph, it is easy to see that for those state dimensions which have a directed path to $r_{t+\tau}$, a_{t-1} and $\mathbf{s}_{t-1}$ cannot break the path between $s_{i,t}$ and R_t. Moreover, for those state dimensions which do not have a directed path to $r_{t+\tau}$, a_{t-1} and s_{t-1}^r are enough to break the path between $s_{i,t}$ and R_t.

Therefore, for $s_{i,t}$, $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, \mathbf{s}_{t-1}$, if and only if $s_{i,t} \not\perp\!\!\!\perp R_{t+1} | a_{t-1}, s_{t-1}^r$.

Step 3: In the previous steps, it has been shown that if a state dimension $s_{i,t}$ is in s_t^r, then $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$, and if a state dimension $s_{i,t}$ is not in s_t^r, then $s_{i,t} \perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$. This implies that s_t^r are minimally sufficient for policy learning to maximize the future reward. □

S1.2 Proof of Proposition 2

Moreover, the proposition below shows that s_t^a, which receives an edge from a_{t-1}, can be directly controlled by actions and can be characterized by conditional dependence with the action variable.

Proposition 2. *Under the assumption that the graphical representation, corresponding to the environment model, is Markov and faithful to the measured data, $s_t^a \subseteq \mathbf{s_t}$ is a minimal subset of state dimensions that are sufficient for direct control, and $s_{i,t} \in s_t^a$ if and only if $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$.*

Below, we give the proof of Proposition 2.

Proof. The proof contains the following two steps.

- In step 1, we show that a state dimension $s_{i,t}$ is in s_t^a, that is, it receives an edge from a_{t-1}, if and only if $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$.
- In step 2, we show that s_t^a contains a minimally sufficient subset of state dimensions that can be directly controlled by actions.

Step 1: We first show that if a state dimension $s_{i,t}$ is in s_t^a, then $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$.

We prove it by contradiction. Suppose that $s_{i,t}$ is independent of a_{t-1} given $\mathbf{s}_{t-1}$. Then according to the faithfulness assumption, we can see from the graph that $s_{i,t}$ does not receive an edge from a_{t-1}, which contradicts the assumption, because, otherwise, $\mathbf{s}_{t-1}$ cannot break the paths between $s_{i,t}$ and a_{t-1} which leads to the dependence.

We next show that if $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$, then $s_{i,t} \in s_t^a$.

Similarly, by contradiction suppose that $s_{i,t}$ does not receive an edge from a_{t-1}. From the graph, it is easy to see that $\mathbf{s}_{t-1}$ must break the path between $s_{i,t}$ and a_{t-1}. According to the Markov assumption, $s_{i,t}$ is independent of a_{t-1} given $\mathbf{s}_{t-1}$, which contradicts to the assumption. Since we have a contradiction, it must be that $s_{i,t}$ has an edge from a_{t-1}.

Step 2: In the previous steps, it has been shown that if a state dimension $s_{i,t}$ is in s_t^a, then $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$, and if a state dimension $s_{i,t}$ is not in s_t^a, then $s_{i,t} \perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$. This implies that s_t^a is minimally sufficient for one-step direct control. □

S1.3 Proof of Proposition 3

Furthermore, based on Proposition 1 and Proposition 2, we can further differentiate s_t^{ar}, $s_t^{\bar{a}r}$, $s_t^{a\bar{r}}$ from s_t^r and s_t^a, which is given in the following proposition.

Proposition 3. *Under the assumption that the graphical representation, corresponding to the environment model, is Markov and faithful to the measured data, we can build a connection between the graph structure and statistical independence of causal variables in the RL system, with (1) $s_{i,t} \in s_t^{ar}$ if and only if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$ and $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$, (2) $s_{i,t} \in s_t^{\bar{a}r}$ if and only if $s_{i,t} \not\perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$ and $s_{i,t} \perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$, (3) $s_{i,t} \in s_t^{a\bar{r}}$ if and only if $s_{i,t} \perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$ and $s_{i,t} \not\perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$, and (4) $s_{i,t} \in s_t^{\bar{a}\bar{r}}$ if and only if $s_{i,t} \perp\!\!\!\perp R_t | a_{t-1}, s_{t-1}^r$ and $s_{i,t} \perp\!\!\!\perp a_{t-1} | \mathbf{s_{t-1}}$.*

Proof. This proposition can be easily proved by levering the results from Propositions 1 and 2. □

S1.4 Proof of Theorem 1

According to the causal process in the RL system (as described in Eq.1 in [3]), we can build the following mapping from latent state variables $\mathbf{s}_t$ to observed variables o_t and future cumulative reward R_t:

$$[o_t, R_t] = f(s_t^r, s_t^{\bar{r}}, \eta_t), \tag{1}$$

where

$$\begin{aligned} o_t &= f_1(s_t^r, s_t^{\bar{r}}), \\ R_t &= f_2(s_t^r, \eta_t). \end{aligned} \tag{2}$$

Here, note that to recover s_t^r, it is essential to take into account all future rewards $r_{t:T}$, because any state dimension $s_{i,t} \in \mathbf{s}_t$ that has a directed path to the future reward $r_{t+\tau}$, for $\tau > 0$, is involved in s_t^r. Hence, we consider the mapping from s_t^r to the future cumulative reward R_t, and η_t represents residuals, except s_t^r, that have an effect to R_t.

The following theorem shows that the different types of states s_t^{ar}, $s_t^{\bar{a}r}$, $s_t^{a\bar{r}}$, and $s_t^{\bar{a}\bar{r}}$ are blockwise identifiable from observed image variable o_t, reward variable r_t, and action variable a_t, under reasonable and weak assumptions.

Theorem 1. *Suppose that the causal process in the RL system and the four categories of latent state variables can be described as that in Section 2 and illustrated in Figure 1(c). Under the following assumptions*

A1. The mapping f in Eq. 1 is smooth and invertible with smooth inverse.

A2. For all $i \in \{1, \ldots, d_o + d_R\}$ and $j \in \mathcal{F}_{i,:}$, there exist $\{\tilde{\mathbf{s}}_t^{(l)}\}_{l=1}^{|\mathcal{F}_{i,:}|}$, so that span$\{\mathbf{J}_f(\tilde{\mathbf{s}}_t^{(l)})_{i,:}\}_{l=1}^{|\mathcal{F}_{i,:}|} = \mathbb{R}^{d_{\tilde{s}}}_{\mathcal{F}_{i,:}}$, and there exists a matrix T with its support identical to that of $\mathbf{J}_{\hat{f}}^{-1}(\hat{\tilde{\mathbf{s}}}_t)\mathbf{J}_f(\tilde{\mathbf{s}}_t)$, so that $[\mathbf{J}_f(\tilde{\mathbf{s}}_t^{(l)})T]_{j,:} \in \mathbb{R}^{d_{\tilde{s}}}_{\hat{\mathcal{F}}_{i,:}}$.

Then, reward-relevant and controllable states s_t^{ar}, reward-relevant but not controllable states $s_t^{\bar{a}r}$, reward-irrelevant but controllable states $s_t^{a\bar{r}}$, and noise $s_t^{\bar{a}\bar{r}}$, are blockwise identifiable.

In the theorem presented above, Assumption $A1$ only assumes the invertibility of function f, while functions f_1 and f_2 are considered general and not necessarily invertible. Since the function f is the mapping from all (latent) variables, including noise factors, that influence the observed variables, the invertibility assumption holds reasonably. However, note that it is not reasonable to assume the invertibility of the function f_2 since usually, the reward function is not invertible. Assumption $A2$, which is also given in [4, 5], aims to establish a more generic condition that rules out certain sets of parameters to prevent ill-posed conditions. Specifically, it ensures that the Jacobian is not partially constant. This condition is typically satisfied asymptotically, and it is necessary to avoid undesirable situations where the problem becomes ill-posed.

Proof. The proof consists of four steps.

1. In step 1, we show that $s_t^a = s_t^{ar} \cup s_t^{a\bar{r}}$ is blockwise identifiable, by using the characterization that the action variable a_t only directly influences s_t^{ar} and $s_t^{a\bar{r}}$.
2. In step 2, we show that $s_t^r = s_t^{ar} \cup s_t^{\bar{a}r}$ is blockwise identifiable, by using the characterization that the future cumulative reward R_t is only influenced by s_t^{ar} and $s_t^{\bar{a}r}$.
3. In step 3, we show that s_t^{ar} is blockwise identifiable, by using the identifiability of $s_t^{ar} \cup s_t^{a\bar{r}}$ and $s_t^{ar} \cup s_t^{\bar{a}r}$.
4. In step 4, we further show the blockwise identifiability of $s_t^{\bar{a}r}$, $s_t^{a\bar{r}}$, and $s_t^{\bar{a}\bar{r}}$.

Step 1: prove the block identifiability of s_t^a.

For simplicity of notation, below, we omit the subscript t.

Let $h := f^{-1} \circ \hat{f}$. We have

$$\hat{\mathbf{s}} = h(\mathbf{s}), \tag{3}$$

where $h = f^{-1} \circ \hat{f}$ is the transformation between the true latent variable and the estimated one, and $\hat{f} : \mathcal{S} \to \mathcal{X}$ denotes the estimated invertible generating function. Note that as both f^{-1} and $\hat{f}$ are smooth and invertible, h and h^{-1} is smooth and invertible.

Since $h(\cdot)$ is smooth over $\mathcal{S}$, its Jocobian can be written as follows:

$$J_{h^{-1}} = \begin{bmatrix} A := \frac{\partial s^{\bar{a}}}{\partial \hat{s}^{\bar{a}}} & B := \frac{\partial s^{\bar{a}}}{\partial \hat{s}^{a}} \\ C := \frac{\partial s^{a}}{\partial \hat{s}^{\bar{a}}} & D := \frac{\partial s^{a}}{\partial \hat{s}^{a}} \end{bmatrix} \tag{4}$$

The invertibility of h^{-1} implies that $J_{h^{-1}}$ is full rank. Since s^a has changing distributions over the action variable a while $s^{\bar{a}}$ has invariant distributions over different values of a, we can derive that $C = 0$. Furthermore, because $J_{h^{-1}}$ is full rank and C is a zero matrix, D must be of full rank, which implies $h_a'^{-1}$ is invertible, where $h_a'^{-1}$ denotes the first derivative of h_a^{-1}. Therefore, s^a is blockwise identifiable up to invertible transformations.

Step 2: prove the blockwise identifiability of s_t^r.

Recall that we have the following mapping:

$$[o_t, R_t] = f(s_t^r, s_t^{\bar{r}}, \eta_t),$$

where

$$\begin{aligned} o_t &= f_1(s_t^r, s_t^{\bar{r}}), \\ R_t &= f_2(s_t^r, \eta_t). \end{aligned}$$

Note that here to recover s_t^r, we need to take into account all future rewards, because s_t^r contains all those state dimensions that have a directed path to future rewards $r_{t+1:T}$. η_t represents all other factors, except s_t^r, that influence R_t at time instance t. Further note here we assume the invertibility of f, while f_1 and f_2 are general functions not necessarily invertible.

We denote by $\tilde{\mathbf{s}} = (s^r, s^{\bar{r}}, \eta)$. We further denote by the dimension of s^r by d_{s^r}, the dimension of $s^{\bar{r}}$ by $d_{s^{\bar{r}}}$, the dimension of $\tilde{\mathbf{s}}$ by $d_{\tilde{s}}$, the dimension of o by d_o, and the dimension of R_t by d_R.

We denote by $\mathcal{F}$ the support of $\mathbf{J}_f(\mathbf{s})$, by $\hat{\mathcal{F}}$ the support of $\mathbf{J}_{\hat{f}}(\hat{\mathbf{s}})$, and by $\mathcal{T}$ the support of $\mathbf{T}(\mathbf{s})$. We also denote T as a matrix with the same support as $\mathcal{T}$. The proof technique is similar to that in [4, 5].

Since $h := \hat{f}^{-1} \circ f$, we have $\hat{f} = f \circ h^{-1}(\tilde{\mathbf{s}})$. By applying the chain rule repeatedly, we have

$$\mathbf{J}_{\hat{f}}(\hat{\tilde{\mathbf{s}}}) = \mathbf{J}_f(\tilde{\mathbf{s}}) \cdot \mathbf{J}_{h^{-1}}(h(\tilde{\mathbf{s}})). \tag{5}$$

With Assumption A2, for any $i \in \{1, \ldots, d_o + d_R\}$, there exists $\{\tilde{\mathbf{s}}^{(l)}\}_{l=1}^{|\mathcal{F}_{i,:}|}$, s.t. $\operatorname{span}(\{\mathbf{J}_f(\tilde{\mathbf{s}}^{(l)})_{i,:}\}_{l=1}^{|\mathcal{F}_{i,:}|}) = \mathrm{R}^{d_{\tilde{s}}}_{\mathcal{F}_{i,:}}$.

Since $\{\mathbf{J}_f(\tilde{\mathbf{s}}^{(l)})_{i,:}\}_{l=1}^{|\mathcal{F}_{i,:}|}$ forms a basis of $\mathrm{R}^{d_{\tilde{s}}}_{\mathcal{F}_{i,:}}$, for any $j_0 \in \mathcal{F}_{i,:}$, we can write canonical basis vector $e_{j_0} \in \mathrm{R}^{d_{\tilde{s}}}_{\mathcal{F}_{i,:}}$ as:

$$e_{j_0} = \sum_{l \in \mathcal{F}_{i,:}} \alpha_l \cdot \mathbf{J}_g(\tilde{\mathbf{s}}^{(l)})_{i,:}, \tag{6}$$

where $\alpha_l \in \mathrm{R}$ is a coefficient.

Then, following Assumption A2, there exists a deterministic matrix T such that

$$T_{j_0,:} = e_{j_0}^\top T = \sum_{l \in \mathcal{F}_{i,:}} \alpha_l \cdot \mathbf{J}_g(\tilde{\mathbf{s}}^{(l)})_{i,:} T \in \mathrm{R}^{d_{\tilde{s}}}_{\hat{\mathcal{F}}_{i,:}}, \tag{7}$$

where $\in$ is due to that each element in the summation belongs to $\mathrm{R}^{d_{\tilde{s}}}_{\hat{\mathcal{F}}_{i,:}}$.

Therefore,

$$\forall\, j \in \mathcal{F}_{i,:}, T_{j,:} \in \mathrm{R}^{d_{\tilde{s}}}_{\hat{\mathcal{F}}_{i,:}}.$$

Equivalently, we have:

$$\forall (i, j) \in \mathcal{F}, \quad \{i\} \times T_{j,:} \subset \hat{\mathcal{F}}. \tag{8}$$

We would like to show that $\hat{s}^r$ does not depend on $s^{\bar{r}}$ and η, that is, $T_{i,j} = 0$ for $i \in \{1, \ldots, d_{s^r}\}$ and $j \in \{d_{s^r} + 1, \ldots, d_{\tilde{s}}\}$.

We prove it by contradiction. Suppose that $\hat{s}^r$ had dependence on $s^{\bar{r}}$, that is, $\exists (j_{s^r}, j_{s^{\bar{r}}}) \in \mathcal{T}$ with $j_{s^r} \in \{1, \ldots, d_{s^r}\}$ and $j_{s^{\bar{r}}} \in \{d_{s^r} + 1, \ldots, d_{s^r} + d_{s^{\bar{r}}}\}$.

Hence, there must exist $i_r \in \{d_o + 1, \ldots, d_o + d_R\}$, such that, $(i_r, j_{s^{\bar{r}}}) \in \mathcal{F}$.

It follows from Equation 8 that:

$$\{i_r\} \times \mathcal{T}_{j_{s^{\bar{r}}},:} \in \hat{\mathcal{F}} \implies (i_r, j_{s^{\bar{r}}}) \in \hat{\mathcal{F}}. \tag{9}$$

However, due to the structure of $\hat{f}_2$, $[\mathbf{J}_{\hat{f}_2}]_{i_r, j_{\hat{s}^{\bar{r}}}} = 0$, which results in a contradiction. Therefore, such $(i_r, j_{s^{\bar{r}}})$ does not exist and $\hat{s}^r$ does not depend on $s^{\bar{r}}$. The same reasoning implies that $\hat{s}^r$ does not dependent on η. Thus, $\hat{s}_r$ does not depend on $(s^{\bar{r}}, \eta)$. In conclusion, $\hat{s}^r$ does not contain extra information beyond s^r.

Similarly, we can show that $(\hat{s}^{\bar{r}}, \hat{\eta})$ does not contain information of s_r.

Therefore, there is a one-to-one mapping between s^r and $\hat{s}^r$.

Step 3: prove the blockwise identifiability of s_t^{ar}.

In Step 1 and Step 2, we have shown that both s^a and s^r are blockwise identifiable. That is,

$$\begin{aligned} \hat{s}^r &= h_r(s^r), \\ \hat{s}^a &= h_a(s^a), \end{aligned} \tag{10}$$

where h_a and h_r are invertible functions.

According to the invariance relation of s^{ar}, We have the following relations:

$$\hat{s}^{ar} = h_r(s^r)_{1:d_{s^{ar}}} = h_a(s^a)_{1:d_{s^{ar}}}. \tag{11}$$

It remains to show that both $\tilde{h}_r := h_r(\cdot)_{1:d_{s^{ar}}}$ and $\tilde{h}_a := h_a(\cdot)_{1:d_{s^{ar}}}$ do not depend on $s^{\bar{a}r}$ and $s^{a\bar{r}}$ in their arguments.

We will prove this by contradiction. Without loss of generality, we suppose $\exists l \in \{1, \cdots, d_{s^{\bar{a}r}}\}$, $s^{r*} \in \mathcal{S}^r$, s.t., $\frac{\partial \tilde{h}_r}{\partial s_l^{\bar{a}r}}(s^{r*}) \neq 0$. As h is smooth, it has continuous partial derivatives. Thus, $\frac{\partial \tilde{h}_r}{\partial s_l^{\bar{a}r}} \neq 0$ holds true in a neighbourhood of s^{r*}, i.e.,

$$\exists \eta > 0, s.t., s_l^{\bar{a}r} \to \tilde{h}_r(s^{ar*}, (s_{-l}^{\bar{a}r*}, s_l^{\bar{a}r*})) \text{ is strictly monotonic on } (s_l^{\bar{a}r*} - \eta, s_l^{\bar{a}r*} + \eta), \tag{12}$$

where $s_{-l}^{\bar{a}r}$ denotes variable $s^{\bar{a}r}$ excluding the dimension l.

We further define an auxiliary function $\psi : \mathcal{S}^{ar} \times \mathcal{S}^{\bar{a}r} \times \mathcal{S}^{a\bar{r}} \to \mathtt{R}_{\geq 0}$ as follows:

$$\psi(s^{ar}, s^{\bar{a}r}, s^{a\bar{r}}) := |\tilde{h}_r(s^r) - \tilde{h}_a(s^a)|. \tag{13}$$

To obtain the contradiction to the invariance, it remains to show that $\psi > 0$ with a probability greater than zero w.r.t. the true generating process.

There are two situations at $(s^{ar*}, s^{\bar{a}r*}, s^{a\bar{r}*})$ where $s^{\bar{a}r*}$ is an arbitrary point in $\mathcal{S}^{\bar{a}r}$:

- situation 1: $\psi(s^{ar*}, s^{\bar{a}r*}, s^{a\bar{r}*}) > 0$;
- situation 2: $\psi(s^{ar*}, s^{\bar{a}r*}, s^{a\bar{r}*}) = 0$.

In situation 1, we have identified a specific point $\psi(s^{ar*}, s^{\bar{a}r*}, s^{a\bar{r}*})$ that makes $\psi > 0$.

In situation 2, Eq. 12 implies that $\forall s_l^{\bar{a}r} \in (s_l^{ar*}, s_l^{ar*} + \eta)$

$$\psi(s^{ar*}, (s_{-l}^{\bar{a}r*}, s_l^{\bar{a}r}), s^{a\bar{r}*}) > 0.$$

Thus, in both situations, we can locate a point $(s^{ar*}, s^{\bar{a}r*'}, s^{a\bar{r}*})$ such that $\psi(s^{ar*}, s^{\bar{a}r*'}, s^{a\bar{r}*}) > 0$, where $s^{\bar{a}r*'} = s^{\bar{a}r*}$ in case 1 and $s_l^{\bar{a}r*'} \in (s_l^{ar*}, s_l^{ar*} + \eta)$, $s_{-l}^{ar*'} = s_{-l}^{ar*}$ in situation 2.

Since ψ is a composition of continuous functions, it is continuous. As pre-image of open sets are always open for continuous functions, the open set $\mathtt{R}_{>0}$ has an open set $\mathcal{U} \in \mathcal{S}^{ar} \times \mathcal{S}^{\bar{a}r} \times \mathcal{S}^{a\bar{r}}$ as its preimage. Due to $(s^{ar*}, s^{\bar{a}r*'}, s^{a\bar{r}*}) \in \mathcal{U}$, $\mathcal{U}$ is nonempty. As $\mathcal{U}$ is nonempty and open, $\mathcal{U}$ has a Lebesgue measure of greater than zero.

As we assume that $p_{s^{ar}, s^{\bar{a}r}, s^{a\bar{r}}}$ is fully supported over the entire domain $\mathcal{S}^{ar} \times \mathcal{S}^{\bar{a}r} \times \mathcal{S}^{a\bar{r}}$, we can deduce that $\mathtt{P}_p[\mathcal{U}] > 0$. That is, $\psi > 0$ with a probability greater than zero, which contradicts the invariance condition, Therefore, it has been shown that $\hat{h}_r(s^r)$ does not depend on $s^{\bar{a}r}$. This proof technique is related to that in [6].

Similarly, we can show that $\hat{h}_a(s^a)$ does not depend on $s^{a\bar{r}}$.

Finally, the smoothness and invertibility of $\hat{h}_r$ and $\hat{h}_a$ follow from the smoothness and invertibility of h_r and h_a over the entire domain.

Therefore, $h_r(h_a)$ is a smooth invertible mapping between s^{ar} and $\hat{s}^{ar}$. That is, s^{ar} is blockwise invertible.

Step 4: prove the blockwise identifiability of $s_t^{\bar{a}r}$, $s_t^{a\bar{r}}$, and $s_t^{\bar{a}\bar{r}}$.

We can use the same technique in Step 3 to show the identifiability of $s^{\bar{a}r}$ and $s^{a\bar{r}}$. Specifically, since s_r and s^{ar} are identifiable, we can show that $s^{\bar{a}r}$ is identifiable. Similarly, since s^a and s^{ar} are identifiable, we can show that $s^{a\bar{r}}$ is identifiable. Furthermore, since s^{ar}, $s^{\bar{a}r}$, and $s^{a\bar{r}}$ are identifiable, we can show that $s^{\bar{a}\bar{r}}$ is identifiable □

S2 Derivation of the objective function

We start by defining the components of the world mode as follows:

$$\begin{cases} \textit{Observation Model:} & p_\theta\left(o_t \mid \mathbf{s_t}\right) \\ \textit{Reward Model:} & p_\theta\left(r_t \mid s_t^r\right) \\ \textit{Transition Model:} & p_\gamma\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}\right) \\ \textit{Representation Model:} & q_\phi\left(\mathbf{s_t} \mid o_t, \mathbf{s_{t-1}}, a_{t-1}\right) \end{cases} \tag{14}$$

The latent dynamics can be disentangled into four catogories:

$$\begin{array}{ll} \textit{Disentangled Transition Model:} & \textit{Disentangled Representation Model:} \\ \begin{cases} p_{\gamma_1}\left(s_t^{ar} \mid s_{t-1}^r, a_{t-1}\right) \\ p_{\gamma_2}\left(s_t^{\bar{a}r} \mid s_{t-1}^r\right) \\ p_{\gamma_3}\left(s_t^{a\bar{r}} \mid \mathbf{s_{t-1}}, a_{t-1}\right) \\ p_{\gamma_4}\left(s_t^{\bar{a}\bar{r}} \mid \mathbf{s_{t-1}}\right) \end{cases} & \begin{cases} q_{\phi_1}\left(s_t^{ar} \mid o_t, s_{t-1}^r, a_{t-1}\right) \\ q_{\phi_2}\left(s_t^{\bar{a}r} \mid o_t, s_{t-1}^r\right) \\ q_{\phi_3}\left(s_t^{a\bar{r}} \mid o_t, \mathbf{s_{t-1}}, a_{t-1}\right) \\ q_{\phi_4}\left(s_t^{\bar{a}\bar{r}} \mid o_t, \mathbf{s_{t-1}}\right) \end{cases} \end{array} \tag{15}$$

We define the information bottleneck objective for latent dynamics models [7, 8]

$$\max I\left(\mathbf{s_{1:T}}; (o_{1:T}, r_{1:T}) \mid a_{1:T}\right) - \beta \cdot I\left(\mathbf{s_{1:T}}, i_{1:T} \mid a_{1:T}\right), \tag{16}$$

where β is scalar and i_t are dataset indices that determine the observations $p(o_t|i_t) = \delta(o_t - \bar{o}_t)$ as in [9].

Maximizing the objective leads to model states that can predict the sequence of observations and rewards while limiting the amount of information extracted at each time step. We derive the lower bound of the first term in Equation 16:

$$\begin{aligned} & I\left(\mathbf{s_{1:T}}; (o_{1:T}, r_{1:T}) \mid a_{1:T}\right) \\ =& \mathrm{E}_{q(o_{1:T}, r_{1:T}, \mathbf{s_{1:T}}, a_{1:T})}\left(\sum_t \ln p\left(o_{1:T}, r_{1:T} \mid \mathbf{s_{1:T}}, a_{1:T}\right) - \ln p\left(o_{1:T}, r_{1:T} \mid a_{1:T}\right)\}\right) \\ \pm& \mathrm{E}\left(\sum_t \ln p\left(o_{1:T}, r_{1:T} \mid \mathbf{s_{1:T}}, a_{1:T}\right)\right) \\ \geq& \mathrm{E}\left(\sum_t \ln p\left(o_{1:T}, r_{1:T} \mid \mathbf{s_{1:T}}, a_{1:T}\right)\right) - \mathrm{KL}\left(p\left(o_{1:T}, r_{1:T} \mid \mathbf{s_{1:T}}, a_{1:T}\right) \| \prod_t p_\theta\left(o_t \mid s_t\right) p_\theta\left(r_t \mid s_t^r\right)\right) \\ =& \mathrm{E}\left(\sum_t \ln p_\theta\left(o_t \mid \mathbf{s_t}\right) + \ln p_\theta\left(r_t \mid s_t^r\right)\right). \end{aligned} \tag{17}$$

Thus, we obtain the objective function:

$$\mathcal{J}_{\mathrm{O}}^t = \ln p_\theta\left(o_t \mid \mathbf{s_t}\right) \quad \mathcal{J}_{\mathrm{R}}^t = \ln p_\theta\left(r_t \mid s_t^r\right) \tag{18}$$

For the second term in Equation 16, we use the non-negativity of the KL divergence to obtain an upper bound,

$$
\begin{aligned}
& I\left(\mathbf{s_{1:T}}; i_{1:T} \mid a_{1:T}\right) \\
=& \mathrm{E}_{q(o_{1:T}, r_{1:T}, \mathbf{s_{1:T}}, a_{1:T}, i_{1:T})}\left(\sum_t \ln q\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}, i_t\right) - \ln p\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}\right)\right) \\
=& \mathrm{E}\left(\sum_t \ln q_\phi\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}, o_t\right) - \ln p\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}\right)\right) \\
\leq& \mathrm{E}\left(\sum_t \ln q_\phi\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}, o_t\right) - \ln p_\gamma\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}\right)\right) \\
=& \mathrm{E}\left(\sum_t \mathrm{KL}\left(q_\phi\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}, o_t\right) \| p_\gamma\left(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}\right)\right)\right).
\end{aligned}
\tag{19}
$$

According to equation 15, we have $p_\gamma = p_{\gamma_1} \cdot p_{\gamma_2} \cdot p_{\gamma_3} \cdot p_{\gamma_4}$ and $q_\phi = q_{\phi_1} \cdot q_{\phi_2} \cdot q_{\phi_3} \cdot q_{\phi_4}$.

$$
\begin{aligned}
\mathrm{KL}(q_\phi \| p_\gamma) &= \mathrm{KL}(q_{\phi_1} \cdot q_{\phi_2} \cdot q_{\phi_3} \cdot q_{\phi_4} \| p_{\gamma_1} \cdot p_{\gamma_2} \cdot p_{\gamma_3} \cdot p_{\gamma_4}) \\
&= E_{q_\phi}\left(\ln \frac{q_{\phi_1}}{p_{\gamma_1}} + ln \frac{q_{\phi_2}}{p_{\gamma_2}} + ln \frac{q_{\phi_3}}{p_{\gamma_3}} + ln \frac{q_{\phi_4}}{p_{\gamma_4}}\right) \\
&= \mathrm{KL}(q_{\phi_1} \| p_{\gamma_1}) + \mathrm{KL}(q_{\phi_2} \| p_{\gamma_2}) + \mathrm{KL}(q_{\phi_3} \| p_{\gamma_3}) + \mathrm{KL}(q_{\phi_4} \| p_{\gamma_4})
\end{aligned}
\tag{20}
$$

We introduce additional hyperparameters to regulate the amount of information contained within each category of variables:

$$
\mathcal{J}_{\mathrm{D}}^t = -\beta_1 \mathrm{KL}\left(q_{\phi_1} \| p_{\gamma_1}\right) - \beta_2 \mathrm{KL}\left(q_{\phi_2} \| p_{\gamma_2}\right) - \beta_3 \mathrm{KL}\left(q_{\phi_3} \| p_{\gamma_3}\right) - \beta_4 \mathrm{KL}\left(q_{\phi_4} \| p_{\gamma_4}\right). \tag{21}
$$

Additionally, we introduce two supplementary objectives to explicitly capture the distinctive characteristics of the four distinct representation categories. Specifically, we characterize the reward-relevant representations by measuring the dependence between s_t^r and R_t, given a_{t-1} and s_{t-1}^r, that is $I(s_t^r, R_t | a_{t-1}, s_{t-1}^r)$ (see Figure 7(a). Note that if there exists an directed edge from s_t^r to a_t in the graphical model, a_t should also be conditioned). To ensure that s_t^r are minimally sufficient for policy training, we maximize $I(s_t^r, R_t | a_{t-1}, s_{t-1}^r)$ while minimizing $I(s_t^{\bar{r}}, R_t | a_{t-1}, s_{t-1}^r)$ to discourage the inclusion of redundant information in $s^{\bar{r}}t$ concerning the rewards:

$$
I(s_t^r; R_t \,|\, a_{t-1}, s_{t-1}^r) - I(s_t^{\bar{r}}; R_t \,|\, a_{t-1}, s_{t-1}^r). \tag{22}
$$

The conditional mutual information can be expressed as the disparity between two mutual information values.

$$
\begin{aligned}
I(s_t^r; R_t \,|\, a_{t-1}, s_{t-1}^r) &= I(R_t; s_t^r, a_{t-1}, s_{t-1}^r) - I(R_t; a_{t-1}, s_{t-1}^r), \\
I(s_t^{\bar{r}}; R_t \,|\, a_{t-1}, s_{t-1}^r) &= I(R_t; s_t^{\bar{r}}, a_{t-1}, s_{t-1}^r) - I(R_t; a_{t-1}, s_{t-1}^r).
\end{aligned}
\tag{23}
$$

Combining the above two equations, we eliminated the identical terms, ultimately yielding the following formula

$$
I(R_t; s_t^r, a_{t-1}, s_{t-1}^r) - I(R_t; s_t^{\bar{r}}, a_{t-1}, s_{t-1}^r). \tag{24}
$$

We use the Donsker-Varadhan representation to express mutual information as a supremum over functions,

$$
\begin{aligned}
I(X; Y) &= D_{KL}(p(x, y) \| p(x)p(y)) \\
&= \sup_{T \in \mathcal{T}} \mathbb{E}_{p(x,y)}[T(x, y)] - \log \mathbb{E}_{p(x)p(y)}[e^{T(x,y)}].
\end{aligned}
\tag{25}
$$

We employ mutual information neural estimation [10] to approximate the mutual information value. We represent the function T using a neural network that accepts variables (x, y) as inputs and is parameterized by α. The neural network is optimized through stochastic gradient ascent to find the supremum. Substituting x and y with variables defined in Equation 24, our objective is reformulated as follows:

$$
\mathcal{J}_{\mathrm{RS}}^t = \lambda_1 \cdot \{I_{\alpha_1}(R_t; s_t^r, a_{t-1}, \mathbf{sg}(s_{t-1}^r)) - I_{\alpha_2}(R_t; s_t^{\bar{r}}, a_{t-1}, \mathbf{sg}(s_{t-1}^r))\}. \tag{26}
$$

To incorporate the conditions from the original objective, we apply the stop_gradient operation to the variable s^r_{t-1}. Similarly, to ensure that the representations s^a_t are directly controllable by actions, while $s^{\bar{a}}_t$ are not, we maximize the following objective:

$$I(s^a_t; a_{t-1} \mid \mathbf{s_{t-1}}) - I(s^{\bar{a}}_t, a_{t-1} | \mathbf{s_{t-1}}), \tag{27}$$

By splitting the conditional mutual information and eliminating identical terms, we obtain the following objective function:

$$\mathcal{J}^t_{\mathrm{AS}} = \lambda_2 \cdot \{I_{\alpha_3}(a_{t-1}; s^a_t, \mathbf{sg}(\mathbf{s_{t-1}})) - I_{\alpha_4}(a_{t-1}; s^{\bar{a}}_t, \mathbf{sg}(\mathbf{s_{t-1}}))\}. \tag{28}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ can be obtained by maximizing Equation 25. Intuitively, these two objective functions ensure that s^r_t is predictive of the reward, while $s^{\bar{r}}_t$ is not; similarly, s^a_t can be predicted by the action, whereas $s^{\bar{a}}_t$ cannot.

Combine the equation 18, equation 21, equation 26 and equation 28, the total objective function is:

$$\begin{aligned}
\mathcal{J}_{\mathrm{TOTAL}} &= \max_{\phi,\theta,\gamma,\alpha_1,\alpha_3} \min_{\alpha_2\alpha_4} \mathrm{E}_{q_\phi}\left(\sum_t \left(\mathcal{J}^t_{\mathrm{O}} + \mathcal{J}^t_{\mathrm{R}} + \mathcal{J}^t_{\mathrm{D}} + \mathcal{J}^t_{\mathrm{RS}} + \mathcal{J}^t_{\mathrm{AS}}\right)\right) + \text{ const} \\
&= \max_{\phi,\theta,\gamma,\alpha_1,\alpha_3} \min_{\alpha_2,\alpha_4} \mathbb{E}_{q_\phi}\{\log p_\theta(o_t \mid \mathbf{s_t}) + \log p_\theta(r_t \mid s^r_t) \\
&\quad - \sum_{i=1}^{4} \beta_i \cdot \mathrm{KL}(q_{\phi_i} \| p_{\gamma_i}) + \lambda_1 \cdot (I_{\alpha_1} - I_{\alpha_2}) + \lambda_2 \cdot (I_{\alpha_3} - I_{\alpha_4})\} + \text{ const}\,.
\end{aligned} \tag{29}$$

The expectation is computed over the dataset and the representation model. Throughout the model learning process, the objectives for estimating mutual information and learning the world model are alternately optimized.

S2.1 Discussions

In this subsection, we examine the mutual information constraints in equation 26 and equation 28 and their relationship with other objectives. Our findings reveal that while other objectives partially fulfill the desired functionality of the mutual information constraints, incorporating both mutual information objectives is essential for certain environments.

The objective functions can be summarized as follows::

$$\begin{aligned}
\mathcal{J}^t_{\mathrm{O}} = \ln p_\theta(o_t \mid \mathbf{s_t}), \quad \mathcal{J}^t_{\mathrm{R}} = \ln p_\theta(r_t \mid s^r_t), \quad \mathcal{J}^t_{\mathrm{D}} = -\mathrm{KL}\left(q_\phi \| p_\gamma\right), \\
\mathcal{J}^t_{\mathrm{RS}} = \lambda_1 \cdot \{I_{\alpha_1}(R_t; s^r_t, a_{t-1}, \mathbf{sg}(s^r_{t-1})) - I_{\alpha_2}(R_t; s^{\bar{r}}_t, a_{t-1}, \mathbf{sg}(s^r_{t-1}))\}, \\
\mathcal{J}^t_{\mathrm{AS}} = \lambda_2 \cdot \{I_{\alpha_3}(a_{t-1}; s^a_t, \mathbf{sg}(\mathbf{s_{t-1}})) - I_{\alpha_4}(a_{t-1}; s^{\bar{a}}_t, \mathbf{sg}(\mathbf{s_{t-1}}))\},
\end{aligned} \tag{30}$$

and the KL divergence term can be further decomposed into 4 components:

$$\begin{aligned}
\mathcal{J}^t_{D_1} &= -\beta_1 \cdot \mathrm{KL}(q_{\phi_1}\left(s^{ar}_t \mid o_t, s^r_{t-1}, a_{t-1}\right) \| p_{\gamma_1}\left(s^{ar}_t \mid s^r_{t-1}, a_{t-1}\right)) \\
\mathcal{J}^t_{D_2} &= -\beta_2 \cdot \mathrm{KL}(q_{\phi_2}\left(s^{\bar{a}r}_t \mid o_t, s^r_{t-1}\right) \| p_{\gamma_2}\left(s^{\bar{a}r}_t \mid s^r_{t-1}\right)) \\
\mathcal{J}^t_{D_3} &= -\beta_3 \cdot \mathrm{KL}(q_{\phi_3}\left(s^{a\bar{r}}_t \mid o_t, \mathbf{s_{t-1}}, a_{t-1}\right) \| p_{\gamma_3}\left(s^{a\bar{r}}_t \mid \mathbf{s_{t-1}}, a_{t-1}\right)) \\
\mathcal{J}^t_{D_4} &= -\beta_4 \cdot \mathrm{KL}(q_{\phi_4}\left(s^{\bar{a}\bar{r}}_t \mid o_t, \mathbf{s_{t-1}}\right) \| p_{\gamma_4}\left(s^{\bar{a}\bar{r}}_t \mid \mathbf{s_{t-1}}\right)).
\end{aligned} \tag{31}$$

Specifically, maximizing I_{α_1} in $\mathcal{J}^t_{\mathrm{RS}}$ enhances the predictability of R_t based on the current state s^r_{t-1} conditioning on (s^r_{t-1}, a_{t-1}). However, notice that this objective can be partially accomplished by optimizing $\mathcal{J}^t_{\mathrm{R}}$. When learning the world model, both the transition function and the reward function are trained: the reward function predicts the current reward r_t using s^r_t, while the transition model predicts the next state. These combined predictions contribute to the overall prediction of R_t.

Minimizing I_{α_2} in $\mathcal{J}^t_{\mathrm{RS}}$ eliminates extraneous reward-related information present in $s^{\bar{r}}_t$. According to our formulation, $s^{\bar{r}}_t$ can still be predictive of R_t as long as it does not introduce additional predictability

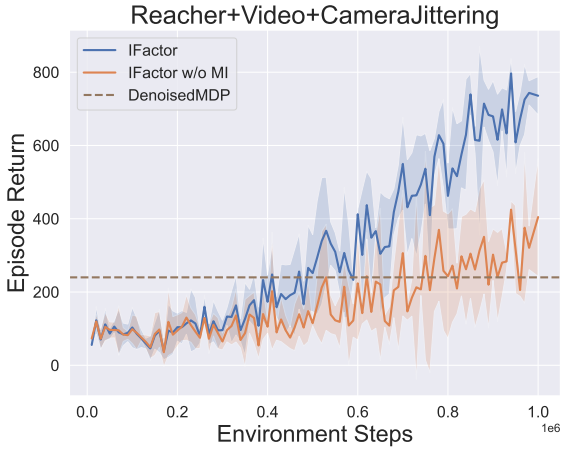


Figure 1: Ablation of the mutual information constraints in the Reacher environment with video background and jittering camera. The dashed brown line illustrates the policy performance of Denoised MDP after 1 million environment steps.

beyond what is already captured by (s^r_{t-1}, a_{t-1}). This is because we only assume that $s^{\bar{r}}_t$ is **conditionally** independent from R_t when conditioning on s^r_{t-1}. If we don't condition on s^r_{t-1}, it introduces s^r_{t-1} as the confounding factor between $s^{\bar{r}}_t$ and R_t, establishing association between $s^{\bar{r}}_t$ and R_t (refer to Figure 7(a)). Note that the KL divergence constraints govern the information amount within each state category. By amplifying the weight of the KL constraints on $s^{\bar{r}}_t$, the value of I_{α_2} can indirectly be diminished.

By maximizing I_{α_3} and minimizing I_{α_4} in $\mathcal{J}^t_{\text{AS}}$, we ensure that s^a_t can be predicted based on $(a_{t-1}, \mathbf{s_{t-1}})$ while $s^{\bar{a}}_t$ cannot. The KL constraints on s^{ar}_t and $s^{a\bar{r}}_t$ incorporate the action a_{t-1} into the prior and posterior, implicitly requiring that s^a_t should be predictable given a_{t-1}. Conversely, the KL constraints on $s^{\bar{a}r}_t$ and $s^{\bar{a}\bar{r}}_t$ do not include the action a_{t-1} in the prior and posterior, implicitly requiring that s^a_t should not be predictable based on a_{t-1}. However, relying solely on indirect constraints can sometimes be ineffective, as it may lead to entangled representations that negatively impact policy performance (see Figure 5).

Ablation of the mutual information constraints. The inclusion of both $\mathcal{J}^t_{\text{RS}}$ and $\mathcal{J}^t_{\text{AS}}$ is essential in certain environments to promote disentanglement and enhance policy performance, despite sharing some common objectives. We have observed improved training stability in the variant of Robodesk environment (see Figure 4) and significant performance gains in the Reacher environment with video background and camera jittering (see Figure 1). When two mutual information objectives are removed, we notice that entangled representations emerged in these environments, as depicted in Figure 5. We assign values of 0.1 to λ_1 and λ_2 in the environment of modified Cartpole, variant of Robodesk and Reacher with video background and jittering camera. Empirically, a value of 0.1 has been found to be preferable for both λ_1 and λ_2. Using a higher value for regularization might negatively impact the learning of representation and transition model. In other DMC environments, the ELBO loss alone has proven effective due to the inherent structure of our disentangled latent dynamics. The choice of hyperparameters $(\beta_1, \beta_2, \beta_3, \beta_4)$ depends on the specific goals of representation learning and the extent of noise interference in the task. If the objective is to accurately recover the true latent variables for understanding the environment, it is often effective to assign equal weights to the four KL divergence terms (for experiments on synthetic data and modified cartpole). When the aim is to enhance policy training stability by mitigating noise, it is recommended to set the values of β_1 and β_2 higher than β_3 and β_4 (for experiments on variants of Robodesk and DMC). Moreover, in environments with higher levels of noise, it is advisable to increase the discrepancy in values between the hyperparameters.

S3 Environment descriptions

S3.1 Synthetic data

For the sake of simplicity, we consider one lag for the latent processes in Section 4. Our identifiability proof can actually be applied for arbitrary lags directly because the identifiability does not rely on the number of previous states. We extend the latent dynamics of the synthetic environment to incorporate a general time-delayed causal effect with $\tau \geq 1$ in the synthetic environment. When $\tau = 1$, it reduces to a common MDP. The ground-truth generative model of the environment is as follows::

$$\begin{cases} \textit{Observation Model:} & p_\theta\left(o_t \mid \mathbf{s_t}\right) \\ \textit{Reward Model:} & p_\theta\left(r_t \mid s_t^r\right) \\ \textit{Transition Model:} & p_\gamma\left(\mathbf{s_t} \mid \mathbf{s_{t-\tau:t-1}}, a_{t-\tau:t-1}\right) \end{cases} \quad \textit{Transition}: \begin{cases} p_{\gamma_1}\left(s_t^{ar} \mid s_{t-\tau:t-1}^r, a_{t-\tau:t-1}\right) \\ p_{\gamma_2}\left(s_t^{\bar{a}r} \mid s_{t-\tau:t-1}^r\right) \\ p_{\gamma_3}\left(s_t^{a\bar{r}} \mid \mathbf{s_{t-\tau:t-1}}, a_{t-\tau:t-1}\right) \\ p_{\gamma_4}\left(s_t^{\bar{a}\bar{r}} \mid \mathbf{s_{t-\tau:t-1}}\right) \end{cases} \tag{32}$$

Data Generation We generate synthetic datasets with 100, 000 data points according to the generating process in Equation 32, which satisfies the identifiability conditions stated in Theorem 1. The latent variables $\mathbf{s_t}$ have 8 dimensions, where $s_t^{ar} = s_t^{\bar{a}r} = s_t^{a\bar{r}} = s_t^{\bar{a}\bar{r}} = 2$. At each timestep, a one-hot action of dimension 5, denoted as a_t, is taken. The lag number of the process is set to $\tau = 2$. The observation model $p_\theta\left(o_t, \mid \mathbf{s_t}\right)$ is implemented using a random three-layer MLPs with LeakyReLU units. The reward model $p_\theta\left(r_t, \mid s_t^r\right)$ is represented by a random one-layer MLP. It's worth noting that the reward model is not invertible due to the scalar nature of r_t. Four distinct transition functions, namely p_{γ_1}, p_{γ_2}, p_{γ_3}, and p_{γ_4}, are employed and modeled using random one-layer MLP with LeakyReLU units. The process noise is sampled from an i.i.d. Gaussian distribution with a standard deviation of $\sigma = 0.1$. To simulate nonstationary noise for various latent variables in RL, the process noise terms are coupled with the historical information by multiplying them with the average value of all the time-lagged latent variables, as suggested in [11].

S3.2 Modified Cartpole

We have modified the original Cartpole environment by introducing two distractors. The first distractor is an uncontrollable Cartpole located in the upper portion of the image, which does not affect the rewards. The second distractor is a controllable green light positioned below the reward-relevant Cartpole in the lower part of the image, but it is not associated with any rewards. The task-irrelevant cartpole undergoes random actions at each time step and stops moving when its angle exceeds 45 degrees or goes beyond the screen boundaries. The action space consists of three independent degrees of freedom: direction (left or right), force magnitude (10N or 20N), and green light intensity (lighter or darker). This results in an 8-dimensional one-hot vector. The objective of this variant is to maintain balance for the reward-relevant cartpole by applying suitable forces.

S3.3 Variant of Robodesk

The RoboDesk environment with noise distractors [12] is a control task designed to simulate realistic sources of noise, such as flickering lights and shaky cameras. Within the environment, there is a large TV that displays natural RGB videos. On the desk, there is a green button that controls both the hue of the TV and a light. The agent's objective is to manipulate this button in order to change the TV's

Figure 2: Visualization of the environments used in our experiments.

		Ctrl + Rew	**$\overline{\text{Ctrl}}$ + Rew**	**Ctrl + $\overline{\text{Rew}}$**	**$\overline{\text{Ctrl}}$ + $\overline{\text{Rew}}$**
Modified Cartpole		Agent	Agent	Green Light	Distractor cartpole
Robodesk		Agent, Button, Light on desk	TV content, Button sensor noise	Blocks on desk, Handle on desk, Other movable objects **(Green hue of TV)**	Jittering and flickering environment lighting, Jittering camera
DMC	**Noiseless**	Agent	**(Agent)**	—	—
	Video Background	Agent	**(Agent)**	—	Background
	Video Background + Noisy Sensor	Agent	**(Agent)** Background	—	—
	Video Background + Camera Jittering	Agent	**(Agent)**	—	Background Jittering camera

Table 1: Categorization of various types of information in the environments we evaluated. We use the red color to emphasize the categorization difference between IFactor and Denoised MDP. Unlike Denoised MDP that assumes independent latent processes, IFactor allows for causally-related processes. Therefore, in this paper, the term "controllable" refers specifically to one-step controllability, while "reward-relevant" is characterized by the conditional dependence between s_t^* and the cumulative reward variable R_t when conditioning on (s_{t-1}, a_{t-1}). Following this categorization, certain agent information can be classified as (one-step) uncontrollable (including indirectly controllable and uncontrollable factors) but reward-relevant factors, such as some position information determined by the position and velocity in the previous time-step rather than the action. On the other hand, the green hue of TV in Robodesk is classified as controllable but reward-irrelevant factors, as they are independent of the reward given the state of the robot arm and green button, aligning with the definition of $s_t^{a\bar{r}}$.

hue to green. The agent's reward is determined based on the greenness of the TV image. In this environment, all four types of information are present (see Table 1).

S3.4 Variants of DeepMind Control Suite

Four variants [12] are introduced for each DMC task:

- **Noiseless**: Original environment without distractors.
- **Video Background**: Replacing noiseless background with natural videos [13] (**$\overline{\text{Ctrl}}$ + $\overline{\text{Rew}}$**).
- **Video Background + Sensor Noise**: Imperfect sensors sensitive to intensity of a background patch (**$\overline{\text{Ctrl}}$ + Rew**).
- **Video Background + Camera Jittering**: Shifting the observation by a smooth random walk (**$\overline{\text{Ctrl}}$ + $\overline{\text{Rew}}$**).

The video background in the environment incorporates grayscale videos from Kinetics-400, where pixels with high blue channel values are replaced. Camera jittering is introduced through a smooth random walk shift using Gaussian-perturbing acceleration, velocity decay, and pulling force. Sensor noise is added by perturbing a specific sensor based on the intensity of a patch in the background video. The perturbation involves adding the average patch value minus 0.5. Different sensors are perturbed for different environments. These sensor values undergo non-linear transformations, primarily piecewise linear, to compute rewards. While the additive reward noise model may not capture sensor behavior perfectly, it is generally sufficient as long as the values remain within moderate ranges and stay within one linear region. (Note: the variants of Robodesk and DMC are not the contributions of this paper. We kindly refer readers to the paper of Denoised MDP [12] for a more detailed introduction.)

S4 Experimental Details

Computing Hardware We used a machine with the following CPU specifications: Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz; 32 CPUs, eight physical cores per CPU, a total of 256 logical CPU units. The machine has two GeForce RTX 2080 Ti GPUs with 11GB GPU memory.

Reproducibility We've included the code for the framework and all experiments in the supplement. We plan to release our code under the MIT License after the paper review period.

S4.1 Synthetic Dataset

Hyperparameter Selection and Network Structure We adopt a similar experimental setup to TDRL [11], while extending it by decomposing the dynamics into four causally related latent processes proposed in this paper (refer to Equation 32). For all experiments, we assign $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.003$ as the weights for the KL divergence terms. In this particular experiment, we set λ_1 and λ_2 to 0 because the utilization of the ELBO loss alone has effectively maximized J^t_{RS} and J^t_{AS}, as illustrated in Figure 3. Here, J^t_{RS} represents $I_{\alpha_1} - I_{\alpha_2}$, and J^t_{AS} represents $I_{\alpha_3} - I_{\alpha_4}$. The network structure employed in this experiment is presented in Table 2.

Training Details The models are implemented in PyTorch 1.13.1. The VAE network is trained using AdamW optimizer for 100 epochs. A learning rate of 0.001 and a mini-batch size of 64 are used. We have used three random seeds in each experiment and reported the mean performance with standard deviation averaged across random seeds.

Table 2: Architecture details. BS: batch size, T: length of time series, o_dim: observation dimension, s_dim: latent dimension, s_t^{ar}_dim: latent dimension for s_t^{ar}, $s_t^{\bar{a}r}$_dim: latent dimension for $s_t^{\bar{a}r}$, $s_t^{a\bar{r}}$_dim: latent dimension for $s_t^{a\bar{r}}$, $s_t^{\bar{a}\bar{r}}$_dim: latent dimension for s_t^{ar} (s_dim = s_t^{ar}_dim + $s_t^{\bar{a}r}$_dim + $s_t^{a\bar{r}}$_dim + $s_t^{\bar{a}\bar{r}}$_dim), LeakyReLU: Leaky Rectified Linear Unit.

Configuration	**Description**	**Output**
1. MLP-Obs-Encoder	Observation Encoder for Synthetic Data	
Input: $o_{1:T}$	Observed time series	BS × T × o_dim
Dense	128 neurons, LeakyReLU	BS × T × 128
Dense	128 neurons, LeakyReLU	BS × T × 128
Dense	128 neurons, LeakyReLU	BS × T × 128
Dense	Temporal embeddings	BS × T × s_dim
2. MLP-Obs-Decoder	Observation Decoder for Synthetic Data	
Input: $\hat{s}_{1:T}$	Sampled latent variables	BS × T × s_dim
Dense	128 neurons, LeakyReLU	BS × T × 128
Dense	128 neurons, LeakyReLU	BS × T × 128
Dense	o_dim neurons, reconstructed $\hat{\mathbf{o}}_{1:T}$	BS × T × o_dim
3. MLP-Reward-Decoder	Reward Decoder for Synthetic Data	
Input: $\hat{s}_{1:T}$	Sampled latent variables	BS × T × s_dim
Dense	1 neurons, LeakyReLU	BS × T × 1
4. Disentangled Prior for s_t^{ar}	Nonlinear Transition Prior Network	
Input	Sampled latents and actions $s^r_{1:T}, a_{1:T}$	BS × T ×(s^r_t_dim + a_dim)
Dense	s_t^{ar}_dim neurons, prior output	BS × T × s_t^{ar}_dim
5. Disentangled Prior for $s_t^{\bar{a}r}$	Nonlinear Transition Prior Network	
Input	Sampled latent variable sequence $s^r_{1:T}$	BS × T × s^r_t_dim
Dense	$s_t^{\bar{a}r}$_dim neurons, prior output	BS × T × $s_t^{\bar{a}r}$_dim
6. Disentangled Prior for $s_t^{a\bar{r}}$	Nonlinear Transition Prior Network	
Input	Sampled latents and actions $\mathbf{s_{1:T}}, a_{1:T}$	BS × T × (s_dim + a_dim)
Dense	$s_t^{a\bar{r}}$_dim neurons, prior output	BS × T × $s_t^{a\bar{r}}$_dim
7. Disentangled Prior for $s_t^{\bar{a}\bar{r}}$	Nonlinear Transition Prior Network	
Input	Sampled latent variable sequence $\mathbf{s_{1:T}}$	BS × T × s_dim
Dense	$s_t^{\bar{a}\bar{r}}$_dim neurons, prior output	BS × T × $s_t^{\bar{a}\bar{r}}$_dim

S4.1.1 Extra Results.

During the training process, we record the estimation value of four mutual information (MI) terms. The corresponding results are presented in Figure 3. Despite not being explicitly incorporated into the objective function, the terms $I_{\alpha_1} - I_{\alpha_2}$ and $I_{\alpha_3} - I_{\alpha_4}$ exhibit significant maximization. Furthermore, the estimation values of I_{α_2} and I_{α_2} are found to be close to 0. These findings indicate that the state variable $s_t^{\bar{r}}$ contains little information about the reward, and the predictability of $s_t^{\bar{a}}$ by the action is also low.

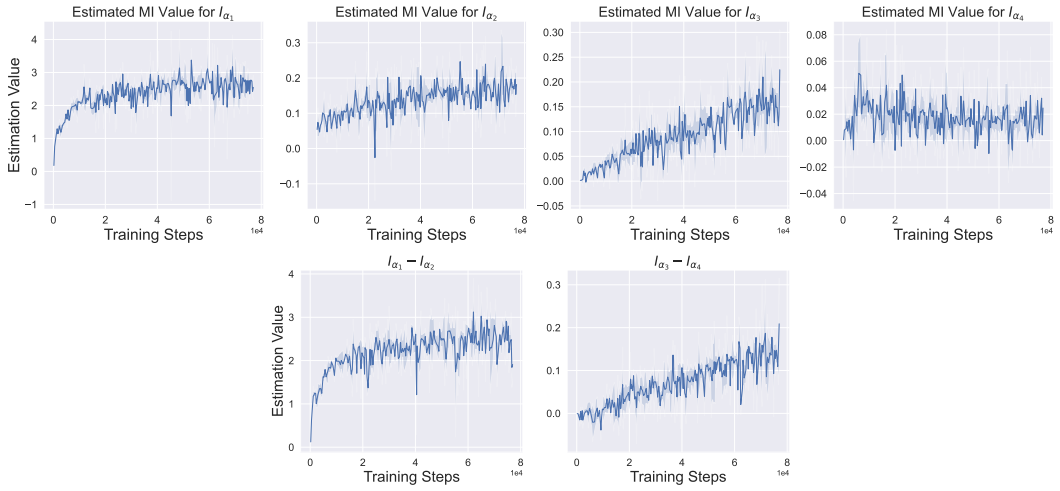


Figure 3: Estimation of the value of four mutual information terms and their differences in experiments on synthetic data.

.

S4.2 Modified Cartpole

In the modified Cartpole environment, we configure the values as follows: $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.1$ and $\lambda_1 = \lambda_2 = 0.1$. Recurrent State Space Model (RSSM) uses a deterministic part and a stochastic part to represent latent variables. The deterministic state size for four dynamics are set to be (15, 15, 15, 15), and the stochastic state size are set to be (2, 2, 1, 4). The architecture of the encoder and decoder for observation is shown in Table 3 and Table 4 (64×64 resolution). Reward model uses 3-layer MLPs with hidden size to be 100 and four mutual information neural estimators are 4-layer MLPs with hidden size to be 128.

S4.3 Variant of Robodesk

In the variant of Robodesk, we conduct experiments with the following hyperparameter settings: $\beta_1 = \beta_2 = 2, \beta_3 = \beta_4 = 0.25$, and $\lambda_1 = \lambda_2 = 0.1$. For the four dynamics, we set the deterministic state sizes to (120, 40, 40, 40), and the stochastic state sizes to (30, 10, 10, 10). Denoised MDP utilizes two latent processes with deterministic state sizes [120, 120] and stochastic state sizes [20, 10]. For the mutual information neural estimators, we employ 4-layer MLPs with a hidden size of 128. To ensure a fair comparison, we align the remaining hyperparameters and network structure with those in the Denoised MDP. We reproduce the results of the Denoised MDP using their released code, maintaining consistency with their paper by employing the default hyperparameters. In order to evaluate the impact of the Mutual Information (MI) constraints, we conduct an ablation study. The results are shown is Figure 4. The constraints $\mathcal{J}_{RS}^t$ and $\mathcal{J}_{AS}^t$ are observed to stabilize the training process of IFactor. The results of IFactor are areaveraged over 5 runs, while the results of Denoised MDP and IFactor without MI are averaged over three runs.

Policy learning based on the learned representations by IFactor We retrain policies using the Soft Actor-Critic algorithm [14] with various combinations of the four learned latent categories as input. We wrap the original environment with visual output using our representation model to obtain compact features. In this process, both deterministic states and stochastic states are utilized

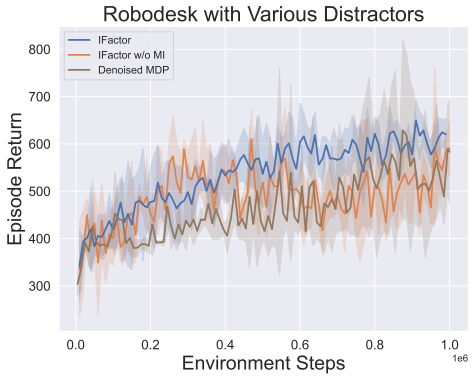


Figure 4: Comparison between IFactor and Denoised MDP in the variant of Robodesk environment.

to form the feature. For instance, when referring to s_t^r, we use both the deterministic states and stochastic states of s_t^r. The implementation of SAC algorithm is based on Stableb-Baselines3[15], with a learning rate of 0.0002. Both the policy network and Q network consist of 4-layer MLPs with a hidden size of 256. We use the default hyperparameter settings in Stable-Baselines3 for other parameters.

S4.4 Variants of Deep Mind Control Suite

In the noiseless DMC environments, we set $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 1$. For the DMC environments with video background, we set $\beta_1 = \beta_2 = 1$ and $\beta_3 = \beta_4 = 0.25$. In the DMC environments with video background and noisy sensor, we set $\beta_1 = \beta_2 = 2$ and $\beta_3 = \beta_4 = 0.25$. Lastly, for the DMC environments with video background and jittering camera, we set $\beta_1 = \beta_2 = 1$ and $\beta_3 = \beta_4 = 0.25$. Regarding the Reacher environment with video background and jittering camera, we set $\lambda_1 = \lambda_2 = 0.1$ for our experiments. For the other environments, we set $\lambda_1 = \lambda_2 = 0$. The deterministic state sizes for the four dynamics are set to (120, 120, 60, 60), while the stochastic state sizes are set to (20, 20, 10, 10). The four mutual information neural estimators utilize a 4-layer MLPs with a hidden size of 128. We align the other hyperparameters and network structure with those used in the Denoised MDP for a fair comparison.

Operator	Input Shape	Kernel Size	Stride	Padding
Input	$[3, 96, 96]$	—	—	—
Conv. + ReLU	$[32, 47, 47]$	4	2	0
Conv. + ReLU	$[64, 22, 22]$	4	2	0
Conv. + ReLU	$[128, 10, 10]$	4	2	0
Conv. + ReLU	$[256, 4, 4]$	4	2	0
Conv. + ReLU *	$[256, 2, 2]$	3	1	0
Reshape + FC	$[1024]$	—	—	—

Table 3: The encoder architecture designed for observation resolution of (96×96). Its output is then fed into other networks for posterior inference. The default activation function used in the network is RELU. For observations with a resolution of (64×64), the last convolutional layer(*) is removed.

S4.4.1 Extra Results.

Figure 1 demonstrates the notable improvement in policy performance in the Reacher environment with video background and jittering camera due to the inclusion of the constraints J_{RS}^t and J_{AS}^t. To

Operator	Input Shape	Kernel Size	Stride	Padding
Input	[input_size]	—	—	—
FC + ReLU + Reshape	[1024, 1, 1]	—	—	—
Conv. Transpose + ReLU *	[128, 3, 3]	3	1	0
Conv. Transpose + ReLU	[128, 9, 9]	5	2	0
Conv. Transpose + ReLU	[64, 21, 21]	5	2	0
Conv. Transpose + ReLU	[32, 46, 46]	6	2	0
Conv. Transpose + ReLU	[3, 96, 96]	6	2	0

Table 4: The decoder architecture designed for (96×96)-resolution observation. For (64×64)-resolution observation, the first transpose convolutional layer(*) is removed.

	Environment Steps	Action Repeat	Train Every	Collection Intervals	Batch Size	Sequence Length	Horizon
Modified Cartpole	200,000	1	5	5	20	30	8
Robodesk	1,000,000	2	1000	100	50	50	15
DMC	1,000,000	2	1000	100	25	50	12

Table 5: Some hyperparameters of our method in the environment of Modified Cartpole, Robodesk and DMC. *Environment Steps* represents the number of interactions between the agent and the environment. *Action Repeat* determines how many times an agent repeats an action in a step. *Train Every* specifies the environment step between adjacent training iterations. *Collection Intervals* defines the number of times the model is trained in each training iteration (including world models, policy networks and value networks). *Batch Size* refers to the number of trajectories in each mini-batch. *Sequence Length* denotes the length of the chuck used in training the world models. *Horizon* determines the length of dreaming when training the policy using the world model. Hyperparameters are aligned with those used in the Denoised MDP for fair comparison.

further investigate how they affects the model learning, we record the estimation values of four Mutual Information terms throughout the training process, as depicted in Figure 5. The results indicate that both $I_{\alpha_1} - I_{\alpha_2}$ and $I_{\alpha_3} - I_{\alpha_4}$ are maximized for both IFactor and IFactor without MI. However, IFactor exhibits a significantly higher rate of maximizing $I_{\alpha_3} - I_{\alpha_4}$ compared to IFactor without MI. This increased maximization leads to greater predictability of s_t^a by the action, ultimately contributing to the observed performance gain.

S4.5 Visualization for DMC

In this experiment, we investigate five types of representations, which can be derived from the combination of four original disentangled representation categories. Specifically, s_t^a is the controllable and reward relevant representation. $s_t^r = (s_t^{ar}, s_t^{\bar{a}r})$ is the reward-relevant representation. $s_t^{\bar{a}\bar{r}}$ is the controllable but reward-irrelevant representation. $s_t^{\bar{a}\bar{r}}$ is the uncontrollable and reward-irrelevant representation (noise). $s_t^{\bar{r}} = (s_t^{a\bar{r}}, s_t^{\bar{a}\bar{r}})$ is the reward-irrelevant representation. Only representations of s_t^r are used for policy optimization. We retrain 5 extra observation decoders to reconstruct the original image, which can precisely characterize what kind of information each type of representation contains, surpassing the limitations of the original decoder that is used in latent traversal. The visualization results are shown in Figure 6. It can be observed that s_t^{ar} captures the movement of the agent partially but not well enough; s_t^r captures the movement of the agent precisely but $s_t^{\bar{r}}$ fails (Reacher and Cheetah) or captures extra information of the background (Walker). This finding suggests that s_t^r contains sufficient information within the original noisy observation for effective control, while effectively excluding other sources of noise.

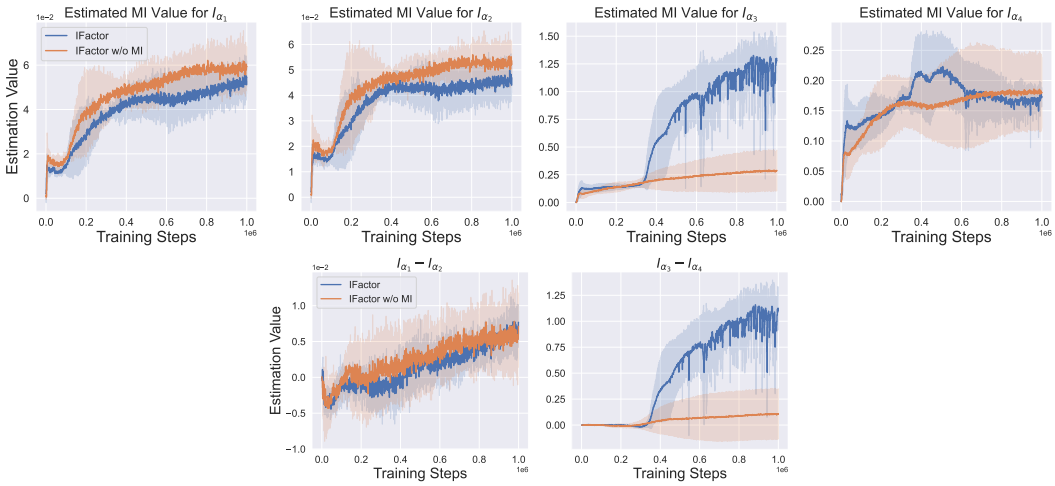


Figure 5: Estimation of the value of four mutual information terms and their differences in the Reacher Easy environment with video background and jittering camera.

.

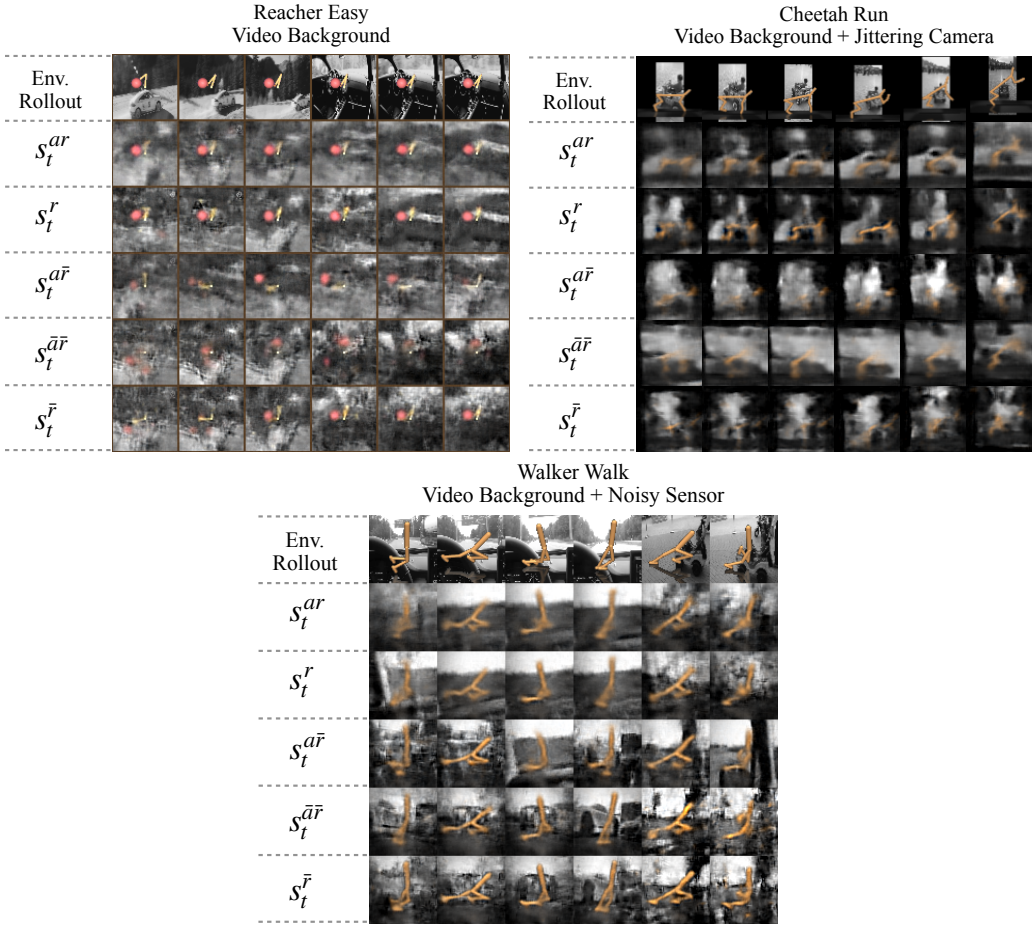
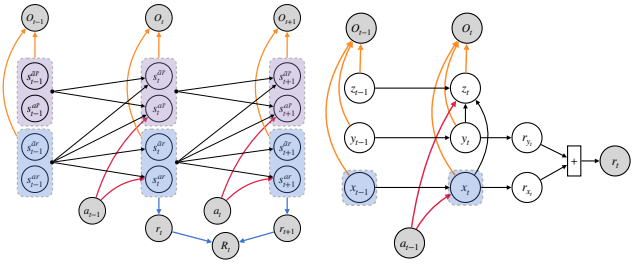


Figure 6: Visualization of the DMC variants and the factorization learned by IFactor.

S5 Comparison between IFactor and Denoised MDP

While both IFactor and Denoised MDP share the common aspect of factorizing latent variables based on controllability and reward relevance, it is crucial to recognize the numerous fundamental distinctions between them.



(a) IFactor (ours). (b) Denoised MDP.

Figure 7: Graphical illustration of our world model and Denoised MDP.

First and foremost, Denoised MDP adopts a rather stringent assumption by solely considering three types of latent variables and assuming independent latent processes for x_t and y_t. However, this strict assumption may not hold in many scenarios where uncontrollable yet reward-relevant factors exhibit dependencies on controllable and reward-relevant factors. Take, for instance, the case of car driving: the agent lacks control over surrounding vehicles, yet their behaviors are indirectly influenced by the agent's actions. In contrast, our approach encompasses four types of causally related latent variables while only assuming conditional independence when conditioning on the state in the previous time step. This assumption holds true naturally within the MDP framework.

Secondly, Denoised MDP is limited to factoring out additive rewards solely for x_t, disregarding the possibility of non-additive effects in many uncontrollable yet reward-relevant factors. In contrast, our method embraces the inclusion of non-additive effects of $s_t^{\bar{a}r}$ on the reward, which is more general.

Thirdly, Denoised MDP uses only controllable and reward-relevant latent variables for policy optimization, which we show in the theoretical analysis that it is generally insufficient. In contrast, our method utilize both controllable and uncontrollable reward-relevant factors for policy training.

Finally, Denoised MDP makes the assumption of an intermediate causal effect from x_t to z_t and from y_t to z_t, which is inherently unidentifiable without further intervention. It is worth noting that imposing interventions on the latent states is unrealistic in most control tasks, as agents can only choose actions at specific states and cannot directly intervene on the state itself. In contrast, our method assumes that there exists no intermediate causal effect for latent variables. In conjunction with several weak assumptions, we provide a proof of block-wise identifiability for our four categories of latent variables. This property serves two important purposes: (1) it ensures the removal of reward-irrelevant factors and the utilization of minimal and sufficient reward-relevant variables for policy optimization, and (2) it provides a potential means for humans to comprehend the learned representations within the reinforcement learning (RL) framework. Through latent traversal of the four types of latent variables, humans can gain insights into the specific kind of information that each category of representation contains within the image.

From the perspective of model structure, it is worth highlighting that the architecture of both the transition model (prior) and the representation model (posterior) in IFactor differs from that of Denoised MDP. The structure of prior and posterior of IFactor is shown as follows:

$$\begin{array}{ll} \text{Prior:} & \text{Posterior:} \\ \begin{cases} p_{\gamma_1}\left(s_t^{ar} \mid s_{t-1}^{r}, a_{t-1}\right) \\ p_{\gamma_2}\left(s_t^{\bar{a}r} \mid s_{t-1}^{r}\right) \\ p_{\gamma_3}\left(s_t^{a\bar{r}} \mid \mathbf{s_{t-1}}, a_{t-1}\right) \\ p_{\gamma_4}\left(s_t^{\bar{a}\bar{r}} \mid \mathbf{s_{t-1}}\right) \end{cases} & \begin{cases} q_{\phi_1}\left(s_t^{ar} \mid o_t, s_{t-1}^{r}, a_{t-1}\right) \\ q_{\phi_2}\left(s_t^{\bar{a}r} \mid o_t, s_{t-1}^{r}\right) \\ q_{\phi_3}\left(s_t^{a\bar{r}} \mid o_t, \mathbf{s_{t-1}}, a_{t-1}\right) \\ q_{\phi_4}\left(s_t^{\bar{a}\bar{r}} \mid o_t, \mathbf{s_{t-1}}\right) \end{cases} \end{array} \tag{33}$$

While Denoised MDP has the following prior and posterior:

$$\begin{array}{ll} \text{Prior:} & \text{Posterior:} \\ \begin{cases} p_{\gamma_1}\left(x_t \mid x_{t-1}, a_{t-1}\right) \\ p_{\gamma_2}\left(y_t \mid y_{t-1}\right) \\ p_{\gamma_3}\left(z_t \mid x_t, y_t, z_{t-1}\right) \end{cases} & \begin{cases} p_{\phi_1}\left(x_t \mid x_{t-1}, y_{t-1}, z_{t-1}, o_t, a_{t-1}\right) \\ p_{\phi_2}\left(y_t \mid x_{t-1}, y_{t-1}, z_{t-1}, o_t, a_{t-1}\right) \\ p_{\phi_3}\left(z_t \mid x_t, y_t, o_t, a_{t-1}\right) \end{cases} \end{array} \tag{34}$$

A notable distinction can be observed between Denoised MDP and IFactor in terms of the assumptions made for the prior and posterior structures. Denoised MDP assumes independent priors for x_t and y_t, whereas IFactor only incorporates conditional independence, utilizing s^r_{t-1} as input for the transition of both s^{ar}_t and $s^{\bar{a}r}_t$. Moreover, the posterior of y_t receives a_{t-1} as input, potentially implying controllability. Similarly, the posterior of x_t incorporates z_{t-1} as input, which may introduce noise from z_{t-1} into x_t. These implementation details can deviate from the original concept. In contrast, our implementation ensures consistency between the prior and posterior, facilitating a clean disentanglement in our factored model.

From the perspective of the objective function, IFactor incorporates two supplementary mutual information constraints, namely $\mathcal{J}^t_{\mathrm{RS}}$ and $\mathcal{J}^t_{\mathrm{AS}}$, to promote disentanglement and improve policy performance.

References

[1] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Spring-Verlag Lectures in Statistics, 1993.

[2] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

[3] 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 *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 9260–9279, Baltimore, Maryland, 2022.

[4] Sébastien Lachapelle, Pau Rodríguez, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Proceedings of the 1st Conference on Causal Learning and Reasoning*, volume 177, pages 428–484, Eureka, CA, 2022.

[5] Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. In *Advances in Neural Information Processing Systems 35*, New Orleans, Louisiana, 2022.

[6] Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*, pages 16451–16467, Virtual Event, 2021.

[7] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[8] Naftali Tishby, Fernando C. N. Pereira, and William Bialek. The information bottleneck method. *CoRR*, physics/0004057, 2000.

[9] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[10] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 530–539, Stockholm, Sweden, 2018.

[11] Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. In *Advances in Neural Information Processing Systems 35*, New Orleans, Louisiana, 2022.

[12] Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. In *Proceedings of the 39th International Conference on Machine Learning*, pages 22591–22612, Baltimore, Maryland, 2022.

[13] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *Proceedings of the 9th International Conference on Learning Representations*, Virtual Event, Austria, 2021.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International conference on machine learning*, pages 1861–1870, Stockholm, Sweden, 2018.

[15] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22:268:1–268:8, 2021.