Appendix

A. Algorithmic and implementation details

A.1. Constrained optimization

Corollary 1 requires an independence assumption between A_t and Φ_t, conditional on X_t. We can therefore cast the problem of learning Φ_t as a constrained optimization problem, where the loss $\mathcal{L}_{\text{hs}}$ measures how predictive of the return Φ_t is, and the constraint enforces a maximum (tolerance) value of β_{IM} for the independence maximization loss $\mathcal{L}_{\text{IM}}$ (exact independence is obtained by $\beta_{\text{IM}} = 0$, but this is hard to achieve exactly in practice).

The resulting optimization problem for finding an appropriate counterfactual baseline is given by:

$$\min_{\theta} \mathbb{E}\left[\mathcal{L}_{\text{hs}}\right] \quad \text{subject to: } \forall t \;\; \mathcal{L}_{\text{IM}}(X_t) \leq \beta_{\text{IM}} \tag{7}$$

The resulting hindsight baseline can then be used in the policy gradient estimate. There are two problems remaining to solve. First, the form of the (IM) loss used requires knowing the exact hindsight probability $\mathbb{P}(A_t|X_t, \Phi_t)$. As explained in the main text, we replace it by the classifier h, tracking the optimal classifier by stochastically minimizing the supervised loss (optimizing it only with respect to the parameters of the hindsight classifier). Second, we relax the constraint using a Lagrangian method (the Lagrangian parameter can either be set as a hyperparameter, or optimized using an algorithm like GECO (Rezende & Viola, 2018)).

A.2. Parameter updates

The corresponding parameter updates are as follows:

For each trajectory $(X_t, A_t, R_t)_{t\geq 0}$, compute the parameter updates :

- $\Delta\theta_{\text{fs}} = -\lambda_{\text{PG}} \sum_t \gamma^t \nabla_{\theta_{\text{fs}}} \log \pi(A_t|X_t)(G_t - V(X_t, \Phi_t)) + \lambda_{\text{H}} \sum_t \nabla_{\theta_{\text{fs}}} \mathcal{L}_{\text{H}}(t) + \lambda_{\text{hs}} \sum_t \nabla_{\theta_{\text{fs}}} \mathcal{L}_{\text{hs}}(t)$
  where $\mathcal{L}_{\text{H}}(t) = -\sum_a \pi(a|X_t) \log \pi(a|X_t)$ is an entropy bonus.
- $\Delta\theta_{\text{hs}} = \lambda_{\text{hs}}(t) \sum_t \nabla_{\theta_{\text{hs}}} \mathcal{L}_{\text{hs}} + \lambda_{\text{IM}} \sum_t \nabla_{\theta_{\text{hs}}} (\mathcal{L}_{\text{IM}}(t) - \beta \mathbb{H}[A_t|X_t])$
- $\Delta\omega = \sum_t \nabla_\omega \mathcal{L}_{\text{sup}}(t)$
- $\Delta\lambda_{\text{IM}} = -\lambda_{\text{IM}} \sum_t (\mathcal{L}_{\text{IM}}(t) - \beta \mathbb{H}[A_t|X_t])$ (when using GECO)

A.3. Design choices

Here we detail practical choices for two aspects of the general CCA algorithm. These concern a) the form of the hindsight function, b) the form of the independence maximization constraint.

Choice of the hindsight function φ

In principle, this function can take any form: in practice, we investigated two architectures. The first is a backward RNN, where $(\Phi_t, B_t) = \text{RNN}(X_t, B_{t+1})$, where B_t is the state of the backward RNN. Backward RNNs are justified in that they can extract information from arbitrary length sequences, and allow making the statistics Φ_t a function of the entire trajectory. They also have the inductive bias of focusing more on near-future observations. The second is a transformer (Vaswani et al., 2017; Parisotto et al., 2019). Alternative networks could be used, such as attention-based networks (Hung et al., 2019) or RIMs (Goyal et al., 2019).

Independence maximization constraint $\mathcal{L}_{\text{IM}}$

We investigated two IM losses. The first is the conditional mutual information $\mathbb{I}(A_t; \Phi_t|X_t) = \mathbb{E}_{\Phi_t|X_t}[\mathbb{H}[A_t|X_t] - \mathbb{H}[A_t|X_t, \Phi_t]]$, where $\mathbb{H}[A|B]$ denotes the conditional entropy $\mathbb{H}[A|B] = -\sum_a P(A = a|B) \log P(A = a|B)$. The expectation can be stochastically approximated by the trajectory sample value $\mathbb{H}(A_t|X_t) - \mathbb{H}(A_t|X_t, \Phi_t)$. The first term is simply the entropy of the policy $-\sum_a \pi(a|X_t) \log \pi(a|X_t)$. The second term is estimated using the h network. The

second we investigated is the Kullback-Leibler divergence, $\mathbb{KL}(\pi(A_t|X_t)||\mathbb{P}(A_t|X_t,\Phi_t)) = \sum_a \pi(a|X_t)\log\pi(a|X_t) - \sum_a \pi(a|X_t)\log\mathbb{P}(a|X_t,\Phi_t)$. Again, we approximate the second term using h. We did not see significant differences between the two, with the KL slightly outperforming the mutual information.

B. Additional Experimental Details

B.1. Bandits

B.1.1. Environment

Our bandit with feedback environment is defined by two positive integers (N, K), a noise level $\sigma_r > 0$ and three arbitrary matrices U, V, W, where $U, V \in \mathbb{R}^{K\times N}$ and $W \in \mathbb{R}^K$. For each replication of the experiment (i.e. each seed), these matrices are sampled from a standard Gaussian distribution and kept constant throughout all episodes. For each episode (of length 1, since this is a bandit problem tackled without meta-learning), we sample a context $-N \leq C \leq N$. Given C, an agent chooses an action $-N \leq A \leq N$. The agent then receives a reward $R = -(C-A)^2 + \epsilon_r$, where ϵ_r is sampled from $\mathcal{N}(0, \sigma_r)$. The agent additionally receives a K-dimensional feedback vector $F = U_C + V_A + W\epsilon_r$, where U_C (resp. V_A) denotes the C$^{\text{th}}$ (resp. A$^{\text{th}}$) column of U (resp. V).

The choices above were made without any particular intent: we would expect the intuitions to generalize for other noise distributions and feedback functions. In section B.1.3, we investigate a decentralized multiagent variant of this problem where the exogenous noise actually corresponds to other players' actions.

B.1.2. Architecture

For the bandit problems, the agent architecture is as follows:

- The hindsight feature Φ is computed by a backward RNN. We tried multiple cores for the RNN: GRU (Chung et al., 2015) with 32 hidden units, a recurrent adder ($h_t = h_{t-1} + \text{MLP}(x_t)$, where the MLP has two layers of 32 units), or an exponential averager ($h_t = \lambda h_{t-1} + (1-\lambda)\text{MLP}(x_t)$).
- The hindsight classifier h_ω is a simple MLP with two hidden layers with 32 units each.
- The policy and value functions are computed as the output of a simple linear layer with concatenated observation and feedback as input.
- All weights are jointly trained with Adam (Kingma & Ba, 2014).
- Hyperparameters are chosen as follows: learning rate $4e$-4, entropy loss $4e$-3, independence maximization tolerance $\beta_{\text{IM}} = 0.1$, $\lambda_{\text{sup}} = \lambda_{\text{hs}} = 1$, λ_{IM} is set through Lagrangian optimization (with GECO).

B.1.3. Additional results

Multi-agent Bandit Problem: In the multi-agent version, the environment is composed of M replicas of the bandit with feedback task. Each agent $i = 1, \ldots, M$ interacts with its own version of the environment, but feedbacks and rewards are coupled across agents. The multi-agent bandit is obtained by modifying the single agent version as follows:

- The contexts C^i are sampled i.i.d. from $\{-N, \ldots, N\}$. C and A now denote the concatenation of all agents' contexts and actions.
- The feedback tensor is (M, K) dimensional, and is computed as $W_c\mathbf{1}(C) + W_a\mathbf{1}(A) + \epsilon_f$; where the W are now three dimensional tensors. Effectively, the feedback for agent i depends on the context and actions of all other agents.
- The terminal joint reward is $\sum_i -(C^i - A^i)^2$ for all agents.

The multi-agent version does not require the exogenous noise ϵ_r, as other agents play the role of exogenous noise; it is a minimal implementation of the example found in section 2.3.

We report results from the multi-agent version of the environment in Fig. 6. As the number of interacting agents increases, the effective variance of the vanilla PG estimator increases as well, and the performance of each agent decreases. In contrast, CCA-PG agents learn faster and reach higher performance (though they never learn the optimal policy).

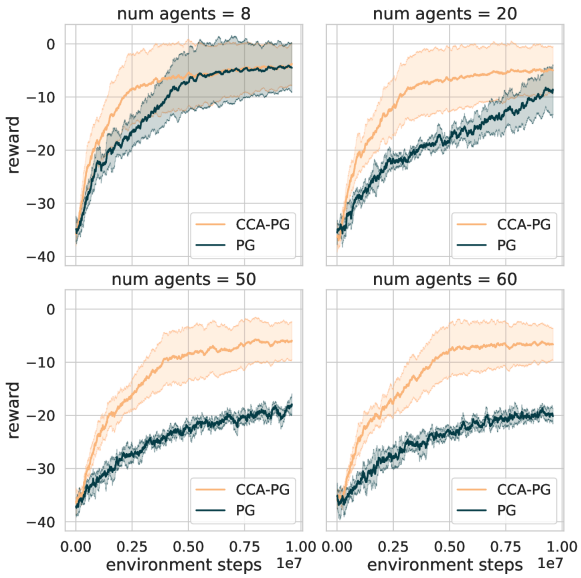

Figure 6: Multi-agent versions of the bandit problem. CCA-PG agents outperform vanilla PG ones.

B.2. Key to Door Tasks

B.2.1. Environment details

Observations returned by the key-to-door family of environments for each of the three phases can be visualized in Fig. 7.

Figure 7: Key-To-Door environments visual. The agent is represented by the beige pixel, key by brown, apples by green, and the final door by blue. The agent has a partial field of view, highlighted in white.

		Lucky (high apple reward)	Unlucky (low apple reward)
Hindsight Advantage Estimate	Skillful (Got key + Door)	1	1
	Unskillful (Did not get key or door)	0	0
Forward Advantage Estimate	Skillful (Got key + Door)	46	-44
	Unskillful (Did not get key or door)	45	-45

Table 1: The advantage estimate of the action of picking up a key in High-Variance-Key-To-Door, as computed by an agent that always picks up every apple, and never picks up the key or the door. We see that an advantage estimate learned using hindsight clearly differentiates between the skillful and unskillful actions; whereas for an advantage estimate learned without using hindsight, this difference is dominated by the extrinsic randomness.

To motivate our approach, Table 1 shows the advantage estimates for either picking up the key or not on High-Variance-Key-To-Door, for an agent that has a perfect apple-phase policy, but never picks up the key or door. Since there are 10 apples which can be worth 1 or 10, the return will be either 10 or 100. Thus the forward baseline in the key phase, i.e. before it has seen how much an apple is worth in the current episode, will be 55. As seen in Table 1, the difference in advantage estimates due to 'luck' is far larger than the difference in advantage estimates due to 'skill' when not using hindsight. This makes learning difficult and leads to the policy never learning to start picking up the key or opening the door. However, when we use a hindsight-conditioned baseline, we are able to learn a Φ (such as the value of a single apple in the current episode) that is completely independent from the actions taken by the agent, but which can provide a perfect hindsight-conditioned baseline of either 10 or 100.

B.2.2. ARCHITECTURE

The agent architecture is as follows:

- The observations are first fed to 2-layer CNN with $(16, 32)$ output channels, kernel shapes of $(3, 3)$ and strides of $(1, 1)$. The output of the CNN is flattened and fed to a linear layer of size 128.
- The agent state is computed by a forward LSTM with a state size of 128. The input to the LSTM is the output of the previous linear layer, concatenated with the reward at the previous timestep.
- The hindsight feature Φ is computed either by a backward LSTM (i.e CCA-PG RNN) with a state size of 128 or by an attention mechanism (Vaswani et al., 2017) (i.e CCA-PG Att) with value and key sizes of 64, 1 transformer block with 2 attention heads, a 1 hidden layer MLP of size 1024, an output size of 128 and a rate of dropout of 0.1. The input provided is the concatenation of the output of the forward LSTM and the reward at the previous timestep.
- The policy is computed as the output of a single-layer MLP with 64 units where the output of the forward LSTM is provided as input.
- The forward baseline is computed as the output of a 3-layer MLP of 128 units each where the output of the forward LSTM is provided as input.
- The hindsight baseline is computed as the sum of the forward baseline and a hindsight residual baseline; the hindsight residual baseline is the output of a 3-layer MLP of 128 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input. It is trained to learn the residual between the return and the forward baseline.
- For CCA, the hindsight classifier h_ω is computed as the concatenation of the output of an MLP, with four hidden layers with 256 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input, and the log of the policy outputs.
- For State HCA, the hindsight classifier h_ω is computed as the output of an MLP, with four hidden layers with 256 units each, where the concatenation of the outputs of the forward LSTM at two given time steps is provided as input.
- For Return HCA, the hindsight classifier h_ω is computed as the output of an MLP, with four hidden layers with 256 units each, where the concatenation of the output of the forward LSTM and the return is provided as input.
- All weights are jointly trained with RMSprop (Hinton et al., 2012) with epsilon $1e\text{-}4$, momentum 0 and decay 0.99.

For High-Variance-Key-To-Door, the optimal hyperparameters found for each algorithm can be found in Table 2.

For Key-To-Door, the optimal hyperparameters found for each algorithm can be found in Table 3.

The agents are trained on full-episode trajectories, using a discount factor of 0.99.

B.2.3. ADDITIONAL RESULTS

As shown in Fig. 8, in the case of vanilla policy gradient, the baseline loss increases at first. As the reward associated with apples varies from one episode to another, getting more apples also means increasing the forward baseline loss. On the

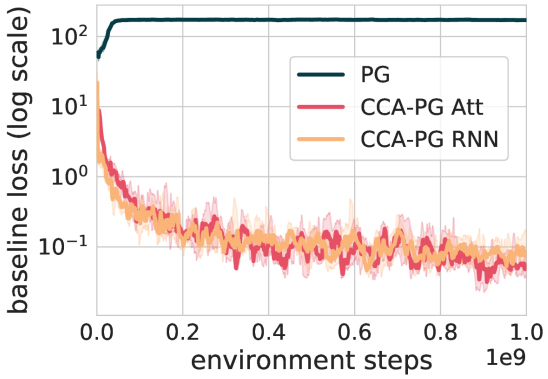

Figure 8: Baseline loss for vanilla PG versus hindsight baseline loss for CCA in **High-Variance-Key-To-Door**.

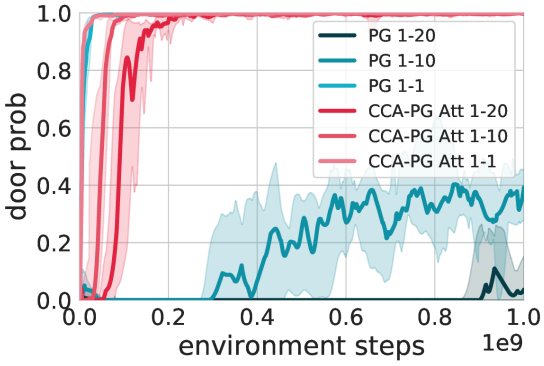

Figure 9: Impact of variance over credit assignment performances. Probability of picking up the key and opening the door as a function of the variance level induced by the apple reward discrepancy between episodes.

other hand, as CCA is able to take into account trajectory specific exogenous factors, the hindsight baseline loss can nicely decrease as learning takes place.

Fig. 9 shows the impact of the variance level induced by the apple reward discrepancy between episodes on the probability of picking up the key and opening the door. Thanks to the use of hindsight in its value function, CCA-PG is almost not impacted by this whereas vanilla PG sees its performances drop dramatically as variance increases.

Fig. 10 shows a qualitative analysis of the attention weights learned by CCA-PG Att on the High-Variance-Key-To-Door task. For this experiment, we used only a single attention head for easier interpretation of the hindsight function, and show both a heatmap of the attention weights over the entire episode, and a histogram of attention weights at the step where the agent picks up the key. As expected, the most attention is paid to timesteps just after the agent picks up an apple - since these are the points at which the apple reward is provided to the Φ computation. In particular, very little attention is paid to the timestep where the agent opens the door. These insights further show that the hindsight function learned is highly predictive of the episode return, while not having mutual information with the action taken by the agent, thus ensuring an unbiased policy gradient estimator.

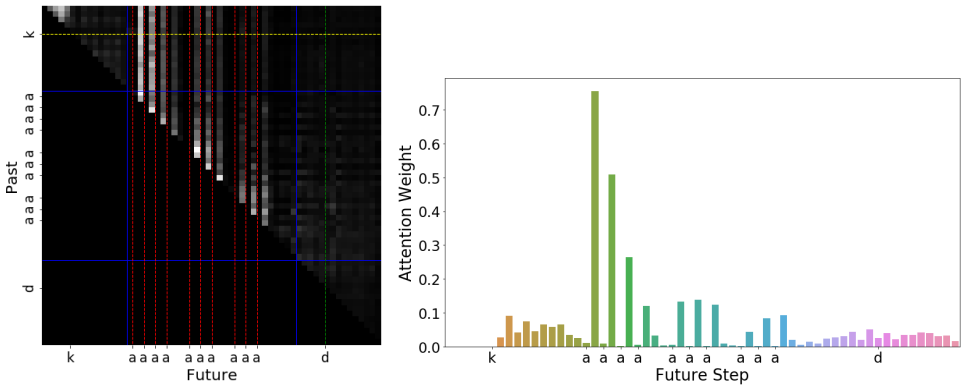

Figure 10: Visualization of attention weights on the High-Variance-Key-To-Door task. **Top:** a 2-dimensional heatmap showing how the hindsight function at each step attends to each step in the future. Red lines indicate the timesteps at which apples are picked up (marked as 'a'); green indicates the door (marked as 'd'); yellow indicates the key (marked as 'k'). **Bottom:** A bar plot of attention over future timesteps, computed at the step where the agent is just about to pick up the key.

	CCA Att	CCA RNN	PG	State HCA	Return HCA
Policy cost	1	1	1	1	1
Entropy cost	5e-3	5e-3	5e-3	5e-3	5e-3
Forward baseline cost	5e-2	5e-2	5e-2	5e-2	5e-2
Hindsight residual baseline cost	5e-2	5e-2	—	———	———
Hindsight classifier cost	1e-2	1e-2	—	1e-2	1e-2
Action independence cost	1e2	1e2	—	———	———
Learning rate	5e-4	5e-4	1e-4	5e-4	1e-3

Table 2: High-Variance-Key-To-Door hyperparameters

	CCA Att	CCA RNN	PG	State HCA	Return HCA
Policy cost	1	1	1	1	1
Entropy cost	5e-3	5e-3	5e-3	5e-3	5e-3
Forward baseline cost	5e-2	5e-2	5e-2	5e-2	5e-2
Hindsight residual baseline cost	5e-2	5e-2	—	———	———
Hindsight classifier cost	1e-2	1e-2	—	1e-2	1e-2
Action independence cost	1e2	1e2	—	———	———
Learning rate	5e-4	5e-4	5e-4	5e-4	5e-4

Table 3: Key-To-Door hyperparameters

B.3. Task Interleaving

B.3.1. Environment details

For each task, a random, but fixed through training, set of 5 out of 10 colored squares are leading to a positive reward. Furthermore, a small reward of 0.5 is provided to the agent when it picks up any colored square. As mentioned previously, each episode are 140 steps long and it takes at least 9 steps for the agent to reach one colored square from its initial position.

The 6 tasks we consider (numbered $\#1$ to $\#6$) are respectively associated with a reward of 80, 4, 100, 6, 2 and 10. Tasks $\#2$, $\#4$, $\#5$ and $\#6$ are referred to as 'hard' while tasks $\#1$ and $\#3$ as 'easy' because of their large associated rewards. The settings 2, 4 and 6-task are respectively considering tasks 1-2, 1-4 and 1-6.

B.3.2. Architecture

We use the same architecture setup as reported in Appendix B.2.2. The agents are also trained on full-episode trajectories, using a discount factor of 0.99.

For Task Interleaving, the optimal hyperparameters found for each algorithm can be found in Table 4.

B.3.3. Additional results

Fig. 11 shows that CCA is able to solve all 6 tasks quickly despite the variance induced by the exogenous factors. Vanilla PG on the other hand despite solving the 'easy' tasks 1 and 3 for which the agent receives big rewards, it is incapable of reliably solve the 4 remaining tasks for which the associated reward is smaller. This helps unpacking Fig. 5.

B.3.4. Ablation Study

Fig.12 shows the impact of the number of back-propagation through time steps performed into the backward RNN of the hindsight function while performing full rollouts. This shows that learning in 'hard' tasks, i.e. where hindsight is crucial for performances, is not much impacted by the number of back-propagation steps performed into the backward RNN. This is great news as this indicates that learning in challenging credit assignment tasks still works when the hindsight function sees the whole future but can only backprop through a limited window.

Fig.13 shows how performances of CCA-RNN are impacted by the unroll length. As expected, the less it is able to look into the future, the harder it becomes to solve hard credit assignment tasks as it is limited in its capacity to take into account

exogenous effects.

The two previous results are promising since CCA seems to only require to have access to as many steps into the future as possible while not needing to do back-propagation through the full sequence.

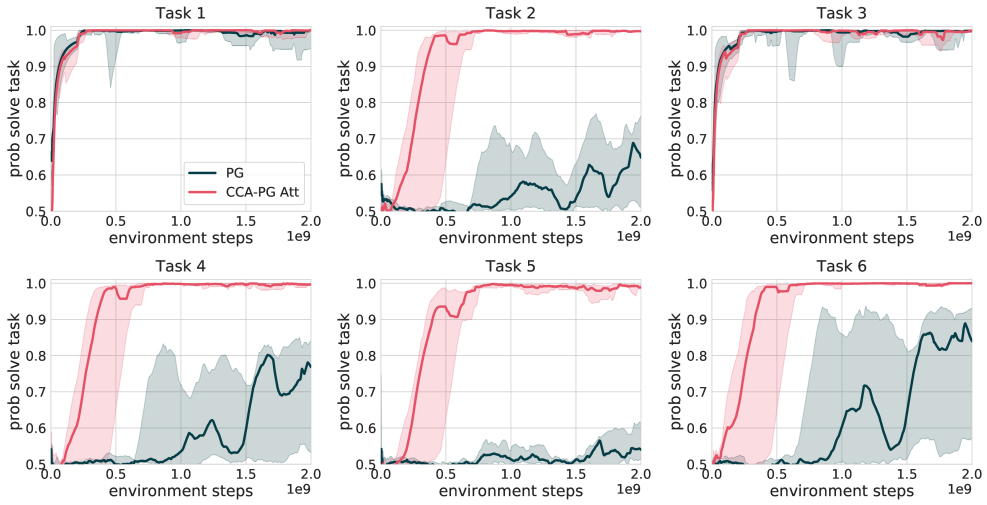

Figure 11: Probability of solving each task in the 6-task setup for **Task Interleaving**.

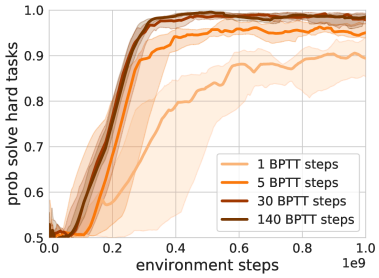

Figure 12: Impact of the number of back-propagation through time steps performed into the hindsight function for CCA-RNN. Probability of solving the 'hard' tasks in the 6-task setup of **Task Interleaving**.

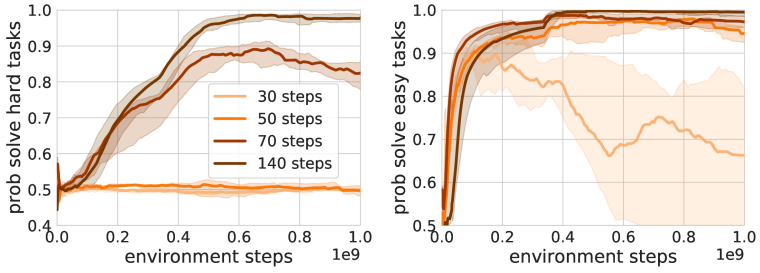

Figure 13: Impact of the unroll length for CCA-RNN. Probability of solving the 'hard' and 'easy' tasks in the 6-task setup of **Task Interleaving**.

	CCA Att	CCA RNN	PG
Policy cost	1	1	1
Entropy cost	5e-2	5e-2	5e-2
Forward baseline cost	1e-2	5e-3	5e-2
Hindsight residual baseline cost	1e-2	5e-3	—
Hindsight classifier cost	1e-2	1e-2	—
Action independence cost	1e1	1e1	—
Learning rate	5e-4	5e-4	1e-3

Table 4: Task Interleaving hyperparameters

C. Relation between HCA, CCA, and FC estimators

The FC estimators generalize both the HCA and CCA estimators. From FC, we can derive CCA by assuming that Φ_t and A_t are conditionally independent (see next section). We can also derive state and return HCA from FC.

For return HCA, we obtain both an all-action and baseline version of return HCA by choosing $\Phi_t = G_t$. For state HCA, we first need to decompose the return into a sum of rewards, and apply the policy gradient estimator to each reward separately. For a pair (X_t, R_{t+k}), and assuming that R_{t+k} is a function of X_{t+k} for simplicity, we choose $\Phi_t = X_{t+k}$. We then sum the different FC estimators for different values of k and obtain both an all-action and single-action version of state HCA.

Note however that HCA and CCA *cannot* be derived from one another. Both estimators leverage different approaches for unbiasedness, one (HCA) leveraging importance sampling, and the other (CCA) eschewing importance sampling in favor of constraint satisfaction (in the context of inference, this is similar to the difference between obtaining samples of the posterior by importance sampling versus directly parametrizing the posterior distribution).

D. Proofs

D.1. Policy gradients

Proof of equation 1. By linearity of expectation, the expected return can be written as $\mathbb{E}[G] = \sum_t \gamma^t \mathbb{E}[R_t]$. Writing the expectation as an integral over trajectories, we have:

$$\mathbb{E}[R_t] = \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\prod_{s \leq t} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) R(x_t, a_t)$$

Taking the gradient with respect to θ:

$$\nabla_\theta \mathbb{E}[R_t] \quad = \quad \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\sum_{s' \leq t} \nabla_\theta \pi_\theta(a_{s'}|x_{s'}) P(x_{s'+1}|x_{s'}, a_{s'}) \left(\prod_{s \leq t, s \neq s'} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) \right) R(x_t, a_t)$$

We then rewrite $\nabla_\theta \pi_\theta(a_{s'}|x_{s'}) = \nabla_\theta \log \pi_\theta(a_{s'}|x_{s'}) \pi_\theta(a_{s'}|x_{s'})$, and obtain

$$\begin{aligned}
\nabla_\theta \mathbb{E}[R_t] =& \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\sum_{s' \leq t} \nabla_\theta \pi_\theta(a_{s'}|x_{s'}) \left(\prod_{s \leq t, s} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) \right) R(x_t, a_t) \\
=& \mathbb{E}\left[\sum_{s' \leq t} \nabla_\theta \log \pi_\theta(A_{s'}|X_{s'}) R_t \right]
\end{aligned}$$

Summing over t, we obtain

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t \sum_{s' \leq t} \nabla_\theta \log \pi_\theta(A_{s'}|X_{s'}) R_t \right]$$

which can be rewritten (with a change of variables):

$$\begin{aligned}
\nabla_\theta \mathbb{E}[G] =& \mathbb{E}\left[\sum_{t \geq 0} \nabla_\theta \log \pi_\theta(A_t|X_t) \sum_{t' \geq t} \gamma^{t'} R_{t'} \right] \\
=& \mathbb{E}\left[\sum_{t \geq 0} \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \sum_{t' \geq t} \gamma^{t'-t} R_{t'} \right] \\
=& \mathbb{E}\left[\sum_{t \geq 0} \gamma^t S_t G_t \right]
\end{aligned}$$

To complete the proof, we need to show that $\mathbb{E}[S_t V(X_t)] = 0$. By iterated expectation, $\mathbb{E}[S_t V(X_t)] = \mathbb{E}[\mathbb{E}[S_t V(X_t)|X_t]] = \mathbb{E}[V(X_t)\mathbb{E}[S_t|X_t]]$, and we have $\mathbb{E}[S_t|X_t] = \sum_a \nabla_\theta \pi(a|X_t) = \nabla_\theta(\sum_a \pi(a|X_t)) = \nabla_\theta 1 = 0$. □

Proof of equation 2. We start from the single action policy gradient $\nabla_\theta \mathbb{E}[G] = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t S_t G_t\right]$ and analyse the term for time t, $\mathbb{E}[S_t G_t]$.

$$\begin{aligned}
\mathbb{E}[S_t G_t] =& \mathbb{E}[\mathbb{E}[S_t G_t | X_t, A_t]] \\
=& \mathbb{E}[S_t \mathbb{E}[G_t | X_t, A_t]] \\
=& \mathbb{E}[S_t Q(X_t, A_t)] \\
=& \mathbb{E}\left[\mathbb{E}[S_t Q(X_t, A_t)|X_t]\right] \\
=& \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) Q(X_t, a)\right]
\end{aligned}$$

The first and fourth inequality come from different applications of iterated expectations, the second from the fact S_t is a constant conditional on X_t, A_t, and the third from the definition of $Q(X_t, A_t)$. □

D.2. Proof of FC-PG theorem

Proof of theorem 1 (single action). We need to show that $\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\right] = 0$, so that $\frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)$ is a valid baseline. As previously, we proceed with the law of iterated expectations, by conditioning successively on X_t then Φ_t

$$\begin{aligned}
\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\right] =& \mathbb{E}\left[\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\middle| X_t, \Phi_t\right]\right] \\
=& \mathbb{E}\left[V(X_t, \Phi_t)\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)}\middle| X_t, \Phi_t\right]\right]
\end{aligned}$$

Then we note that

$$\begin{aligned}
\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)}\middle| X_t, \Phi_t\right] &= \sum_a \mathbb{P}(a|X_t,\Phi_t)\nabla \log \pi(a|X_t) \frac{\pi(a|X_t)}{\mathbb{P}(a|X_t,\Phi_t)} \\
&= \sum_a \nabla \pi(a|X_t) = 0.
\end{aligned}$$

□

Proof of theorem 1 (all-action). We start from the definition of the Q function:

$$\begin{aligned}
Q(X_t, a) &= \mathbb{E}\left[G_t | X_t, A_t = a\right] \\
&= \mathbb{E}_{\Phi_t}\left[\mathbb{E}\left[G_t | X_t, \Phi_t, A_t = a\right] | X_t, A_t = a\right] \\
&= \int_\phi \mathbb{P}(\Phi = \varphi | X_t, A_t = a) Q(X_t, \Phi_t = \varphi, a)
\end{aligned}$$

We also have

$$\mathbb{P}(\Phi = \varphi | X_t, A_t) = \frac{\mathbb{P}(\Phi = \varphi | X_t)\mathbb{P}(A_t = a | X_t, \Phi_t = \phi)}{\mathbb{P}(A_t = a | X_t)},$$

which combined with the above, results in:

$$\begin{aligned}
Q(X_t, a) =& \int_\phi \mathbb{P}(\Phi = \varphi | X_t) \frac{\mathbb{P}(A_t = a | X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a) \\
=& \mathbb{E}\left[\frac{\mathbb{P}(A_t = a | X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]
\end{aligned}$$

For the compatibility with policy gradient, we start from:

$$\mathbb{E}[S_t G_t] = \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) Q(X_t, a)\right]$$

We replace $Q(X_t, a)$ by the expression above and obtain

$$\begin{aligned}
\mathbb{E}[S_t G_t] =& \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) \mathbb{E}\left[\frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) \frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[\sum_a \nabla_\theta \log \pi_\theta(a|X_t) \mathbb{P}(A_t = a|X_t, \Phi_t = \phi) Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\sum_a \nabla_\theta \log \pi_\theta(a|X_t) \mathbb{P}(A_t = a|X_t, \Phi_t = \phi) Q(X_t, \Phi_t, a)\right]
\end{aligned}$$

Note that in the case of a large number of actions, the above can be estimated by

$$\frac{\nabla_\theta \log \pi_\theta(A'_t|X_t) \mathbb{P}(A'_t|X_t, \Phi_t = \phi)}{\pi_\theta(A'_t|X_t)} Q(X_t, \Phi_t, A'_t),$$

where A'_t is an independent sample from $\pi(.|X_t)$; note in particular that A'_t shall NOT be the action A_t that gave rise to Φ_t, which would result in a biased estimator.

D.3. Proof of CCA-PG theorem

Assume that Φ_t and A_t are conditionally independent on X_t. Then, $\frac{\mathbb{P}(A_t=a|X_t,\Phi_t=\phi)}{\mathbb{P}(A_t=a|X_t)} = 1$. In particular, it is true when evaluating at the random value A_t. From this simple observation, both CCA-PG theorems follow from the FC-PG theorems.

To prove the lower variance of the hindsight advantage estimate, note that

$$\begin{aligned}
\mathbb{V}[G_t - V(X_t, \Phi)] =& \mathbb{E}[(G_t - V(X_t, \Phi_t))^2] \\
=& \mathbb{E}[G_t^2] - \mathbb{E}[V(X_t, \Phi_t)^2] \\
\mathbb{V}[G_t - V(X_t)] =& \mathbb{E}[(G_t - V(X_t))^2] \\
=& \mathbb{E}[G_t^2] - \mathbb{E}[V(X_t)^2]
\end{aligned}$$

To prove the first statement, we have $(G_t - V(X_t, \Phi_t))^2 = G_t^2 + V(X_t, \Phi_t)^2 - 2G_t V(X_t, \Phi_t)$, and apply the law of iterated expectations to the last term:

$$\begin{aligned}
\mathbb{E}[G_t V(X_t, \Phi_t)] =& \mathbb{E}[\mathbb{E}[G_t V(X_t, \Phi_t)|X_t, \Phi_t]] \\
=& \mathbb{E}[V(X_t, \Phi_t)\mathbb{E}[G_t|X_t, \Phi_t]] \\
=& \mathbb{E}[V(X_t, \Phi_t)^2]
\end{aligned}$$

The proof for the second statement is identical. Finally, we note that by Jensen's inequality, we have $\mathbb{E}[V(X_t, \Phi_t)^2] \leq \mathbb{E}[V(X_t)^2]$, from which we conclude that $\mathbb{V}[G_t - V(X_t, \Phi_t)] \leq \mathbb{V}[G_t - V(X_t)]$.

□

E. Variance analysis

E.1. Relation between variance of advantage and variance of policy gradient

Consider an advantage estimate Y_t, i.e. a variable such that $\mathbb{E}[Y_t|X_t = x, A_t = a] = Q(x, a) - V(x)$. Possible choices for Y_t include the CCA estimate $G_t - V(X_t, \Phi_t)$ as well as the actual advantage $\mathcal{A}(x, a) = Q(x, a) - V(x)$. Note that

$\nabla_\theta \mathbb{E}[G_t] = \mathbb{E}[\sum_t \gamma^t S_t Y_t]$. We aim to analyze the variance of a single term $S_t Y_t$ (understanding the variance of the sum is more involved). More precisely, we compare the variance $\mathbb{V}[S_t Y_t | X_t]$ of the policy gradient term $S_t Y_t$ given X_t when using Y_t to that of $S_t \mathcal{A}_t$.

We use the conditional variance formula:

$$\mathbb{V}[S_t Y_t | X_t] = \mathbb{E}[\mathbb{V}[S_t Y_t | X_t, A_t] | X_t] + \mathbb{V}[\mathbb{E}[S_t Y_t | X_t, A_t] | X_t],$$

where S_t is constant given X_t, A_t. Therefore the first term becomes $\mathbb{E}[S_t^2 \mathbb{V}[Y_t | X_t, A_t] | X_t]$, and the second one $\mathbb{V}[S_t \mathbb{E}[Y_t | X_t, A_t] | X_t] = \mathbb{V}[S_t \mathcal{A}_t | X_t]$; this term does not depend on the actual advantage estimate used - it is equal to the variance of the policy gradient estimate when using the exact advantage $\mathcal{A}_t$. The additional variance incurred by using an unbiased advantage estimate Y_t instead of the exact advantage $\mathcal{A}_t$ is therefore:

$$\mathbb{V}[S_t Y_t | X_t] - \mathbb{V}[S_t \mathcal{A}_t | X_t] = \mathbb{E}[S_t^2 \mathbb{V}[Y_t | X_t, A_t] | X_t].$$

We see that the (conditional) advantage variance $\mathbb{V}[Y_t | X_t, A_t]$ (as well as the variance of the score function, and their correlation) drives the variance of the policy gradient estimator. We can further find a loose upper bound purely in terms of the unconditional variance of the advantage. First, suppose that the actions are discrete, and that the action distribution is parametrized by a softmax over logits $l_1, \ldots, l_k$, where k is the number of actions. Note that the score is $S_t = \frac{\partial \log \pi(A_t)}{\partial \theta} = \sum_{a'} \frac{\partial \log \pi(A_t)}{\partial l_{a'}} \frac{\partial l_{a'}}{\partial \theta}$, so

$$\begin{aligned}
|S_t| =& |\sum_{a'} \frac{\partial \log \pi(A_t)}{\partial l_{a'}} \frac{\partial l_{a'}}{\partial \theta}| \\
\leq& \sum_{a'} |\frac{\partial \log \pi(A_t)}{\partial l_{a'}}| |\frac{\partial l_{a'}}{\partial \theta}| \\
\leq& \sum_{a'} |\frac{\partial l_{a'}}{\partial \theta}| \leq ||J||_1
\end{aligned}$$

where J is the jacobian of the function mapping parameters θ to logits. The second inequality is due to $|\frac{\partial \log \pi(a)}{\partial l_{a'}}| = |\delta_{a,a'} - \pi(a')| \leq 1$. It follows that:

$$\begin{aligned}
\mathbb{V}[S_t Y_t | X_t] - \mathbb{V}[S_t \mathcal{A}_t | X_t] \leq& ||J||_1^2 \, \mathbb{E}[\mathbb{V}[Y_t | X_t, \mathcal{A}_t] | X_t] \\
\leq& ||J||_1^2 \, (\mathbb{V}[Y_t | X_t] - \mathbb{V}[\mathcal{A}_t | X_t]))
\end{aligned}$$

again using the law of conditional variance $\mathbb{V}[Y_t | X_t] = \mathbb{V}[\mathbb{E}[Y_t | X_t, \mathcal{A}_t] | X_t] + \mathbb{E}[\mathbb{V}[Y_t | X_t, \mathcal{A}_t] | X_t]$. We thus see that the excess variance incurred by using Y_t in the policy gradient estimate can be upper bounded by a constant times the excess variance of the advantage estimate.

E.2. Variance analysis in the bandit problem

Here we provide a back-of-the-envelope variance analysis of the bandit problem. For simplicity (but reasoning can easily be extended), we assume no context and only two actions $\{0, 1\}$, and three vectors $W, V_0, V_1 \in \mathbb{R}^K$ (randomly sampled from a Gaussian and kept constant across all episodes). ϵ_r and ϵ_f are the reward and observation noise respectively, with standard deviations $\sigma_r \gg \sigma_f$. The feedback vector for action a is $W\epsilon_r + V_a + \epsilon_f$.

A forward (in this case, constant) baseline for this problem will have square advantage roughly scale as a σ_r^2.

Let's consider linear hindsight baseline $\alpha^T F$, which is equal to $\epsilon_r(\alpha^T W) + \alpha^T V_a + \alpha^T \epsilon_f$. The expected square advantage $\mathbb{E}[(G - \alpha^T F)^2]$ is therefore

$$\begin{aligned}
\mathbb{E}[(G - \alpha^T F)^2] =& \pi_0 \mathbb{E}[\left(\epsilon_r(\alpha^T W - 1) + \alpha^T V_0 + \alpha^T \epsilon_f\right)^2] + \pi_1 \mathbb{E}[\left(\epsilon_r(\alpha^T W - 1) + (\alpha^T V_1 - 1) + \alpha^T \epsilon_f\right)^2] \\
=& (\alpha^T W - 1)^2 \sigma_r^2 + (\alpha^T \alpha)\sigma_f^2 + \pi_0 (\alpha^T V_0)^2 + \pi_1 (\alpha^T V_1 - 1)^2
\end{aligned}$$

The vectors W, V_0 and V_1 are independent with probability one (in fact they are nearly orthogonal), one can find a hindsight baseline such that $\alpha^T W - 1 = \alpha^T V_0 = \alpha^T V_1 - 1 = 0$, which leaves an expected squared advantage of $\sigma_f^2 \alpha^T \alpha$ which

is small (for random vectors the matrices will be well-conditioned, the resulting α will have small norm); however that advantage leads to a biased update since the advantage is independent of the action. However, choosing $\alpha^T W = 1$ but $\alpha^T V_0 = \alpha^T V_1 = 0$ leads to a hindsight baseline which is equal to $\epsilon_r + \alpha^T \epsilon_f$, independent from the action; the effect of the noise ϵ_r will be removed entirely from the squared advantage, leading to an unbiased gradient estimator with a considerably lower variance (of order σ_f^2).

F. RL algorithms, common randomness, structural causal models

In this section, we provide an alternative view and intuition behind the CCA-PG algorithm by investigating credit assignment through the lens of causality theory, in particular *structural causal models* (SCMs) (Pearl, 2009a). These ideas are very related to the use of common random numbers (CRN), a standard technique in optimization with simulators (Glasserman & Yao, 1992).

F.1. Structural causal model of the MDP

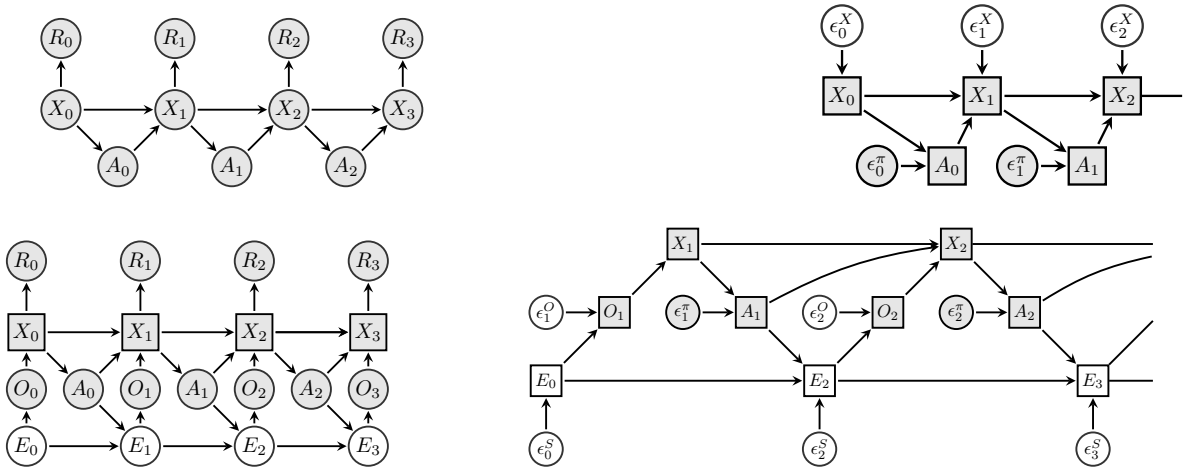

Figure 14: Graphical models and corresponding SCMs for RL problems. Top: MDP, bottom: POMDP; left: graphical model, right: structural causal model. Squares represent deterministic nodes, while circles represent stochastic nodes. Observed nodes are shaded in gray.

Structural causal models (SCM) (Pearl, 2009a) are, informally, models where all randomness is exogenous, and where all variables of interest are modeled as deterministic functions of other variables and of the exogenous randomness. They are of particular interest in causal inference as they enable reasoning about interventions, i.e. how would the *distribution* of a variable change under external influence (such as forcing a variable to take a given value, or changing the process that defines a variable), and about counterfactual interventions, i.e. how would a particular observed outcome (sample) of a variable have changed under external influence. Formally, a SCM is a collection of model variables $\{V \in \boldsymbol{V}\}$, exogenous random variables $\{\mathcal{E} \in \boldsymbol{\mathcal{E}}\}$, and distributions $\{p_{\mathcal{E}}(\varepsilon), \mathcal{E} \in \boldsymbol{\mathcal{E}}\}$, one per exogenous variable, and where the exogenous random variables are all assumed to be independent. Each variable V is defined by a function $V = f_V(\text{pa}(V), \boldsymbol{\mathcal{E}})$, where $\text{pa}(V)$ is a subset of $\boldsymbol{V}$ called the parents of V. The model can be represented by a directed graph in which every node has an incoming edge from each of its parents. For the SCM to be valid, the induced graph has to be a directed acyclic graph (DAG), i.e. there exists a topological ordering of the variables such that for any variable V_i, $\text{pa}(V_i) \subset \{V_1, \ldots, V_{i-1}\}$; in the following we will assume such an ordering. This provides a simple sampling mechanism for the model, where the exogenous random variables are first sampled according to their distribution, and each node is then computed in topological order. Note that any probabilistic model can be represented as a SCM by virtue of reparametrization (Kingma & Ba, 2014; Buesing et al., 2019). However, such a representation is not unique, i.e. different SCMs can induce the same distribution.

In the following we give an SCM representation of a MDP (see Fig.14 for the causal graphical model and corresponding SCM for MDPs and POMDPs). The transition from X_t to X_{t+1} under A_t is given by the transition function f^X: $X_{t+1} = f^X(X_t, A_t, \mathcal{E}_t^X)$ with exogenous variable / random number $\mathcal{E}_t^X$. The policy function f^π maps a random number $\mathcal{E}_t^\pi$, policy parameters θ, and current state X_t to the action $A_t = f^\pi(X_t, \mathcal{E}_t^\pi, \theta)$. Together, f^π and $\mathcal{E}_t^\pi$ induce the policy, a distribution $\pi_\theta(A_t|X_t)$ over actions. Without loss of generality we assume that the reward is a deterministic function

of the state and action: $R_t = f^R(X_t, A_t)$. $\mathcal{E}^X$ and $\mathcal{E}^\pi$ are random variables with a fixed distribution; all changes to the policy are absorbed by changes to the deterministic function f^π. Denoting $\mathcal{E}_t = (\mathcal{E}_t^X, \mathcal{E}_t^\pi)$, note the next reward and state (X_{t+1}, R_t) are deterministic functions of X_t and $\mathcal{E}_t$, since we have $X_{t+1} = f^X(X_t, f^\pi(X_t, \mathcal{E}_t^\pi, \theta), \mathcal{E}_t^X)$ and similarly $R_t = R(X_t, f^\pi(X_t, \mathcal{E}_t^\pi, \theta)$. Let $X_{t^+} = (X_{t'})_{t'>t}$ and similarly, $\mathcal{E}_{t^+} = (\mathcal{E}_t^X, \mathcal{E}_{t'})_{t'>t}$ Through the composition of the functions f^X, f^π and R, the return G_t (under policy π) is a deterministic function f^G of X_t, A_t and $\mathcal{E}_{t^+}$.

F.2. Proof of theorem 2

For notation purposes, in the rest of this section, we will focus on credit assignment for action A_t (since policy gradient terms are additive with respect to time), and will denote $X = X_t$, $A = A_t$, $\varepsilon = \mathcal{E}_{t^+}$, and $\tau = (X_s, R_r, A_s)_{s \geq t}$. Furthermore, we will denote $\Phi = \Phi_t$.

From the arguments in the section above, one can write $\tau = f^\tau(X, A, \varepsilon)$, $G = f^G(\tau)$, and $\Phi = f^\phi(\tau)$. We may integrate out τ, in which case the graph only contains X, A, ε and G. In that graph, by the faithfulness assumption, there can be no causal path from A to Φ, as this would violate the conditional independence assumption. It follows that there are functions g_G and g_Φ such that $G = g_G(X, A, \varepsilon)$ and $\Phi = g_\Phi(X, \varepsilon)$.

The resulting structural causal models can be seen in Fig. 15.

The conditional expectation $Q(x, a, \phi)$ is given by $Q(x, a, \phi) = \int_\varepsilon p(\varepsilon|x, \phi, a) G(x, a, \varepsilon)$ The counterfactual return for action a, having observed ϕ is given by $\mathbb{E}[G(\tau')|\tau' \sim P(\tau'|X = x, \text{observe}(\Phi = \phi))]$ is equal to $\int_\varepsilon p(\varepsilon|x, \phi) G(x, a, \phi)$.

Finally, note that from d-separation $p(\varepsilon|x, \phi) = p(\varepsilon|x, a, \phi)$, and the result follows.

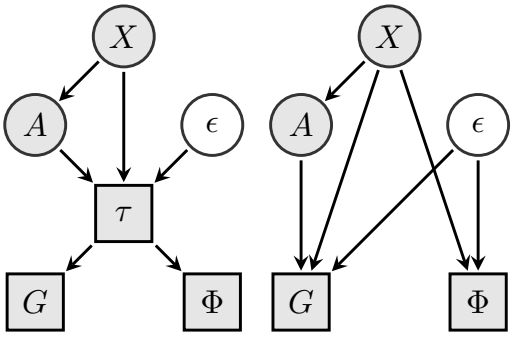

Figure 15: SCMs for the reduced action selection problem; left: including the trajectory; right: trajectory is integrated out. There is no arrow from A to Φ on the right since the graph is assumed to be faithful and A and Φ are conditionally independent given X.

G. Individual Treatment Effects, (Conditional) Average Treatment Effects, Counterfactuals and Counterfactual identifiability

In this section, we will further link the ideas developed in this report to causality theory. In particular we will connect them to two notions of causality theory known as individual treatment effect (ITE) and average treatment effect (ATE). In the previous section, we extensively leveraged the framework of structural causal models. It is however known that distinct SCMs may correspond to the same distribution; learning a model from data, we may learn a model with correct distribution but with with incorrect structural parametrization and counterfactuals. We may therefore wonder whether counterfactual-based approaches may be flawed when using such a model. We investigate this question, and analyze our algorithm in very simple settings for which closed-form computations can be worked out.

G.1. Individual and Average Treatment Effects

Consider a simple medical example which we model with an SCM as illustrated in Fig. 16. We assume population of patients, each with a full medical state denoted S, which summarizes all factors, known or unknown, which affect a patient's future health such as genotype, phenotype etc. While S is never known perfectly, some of the patient's medical history H may be known, including current symptoms. On the basis of H, a treatment decision T is taken; as is often done, for

simplicity we consider T to be a binary variable taking values in {1='treatment', 0='no treatment'}. Finally, health state S and treatment T result in a observed medical outcome O, a binary variable taking values in {1='cured', 0='not cured'}. For a given value $S = s$ and $T = t$, the outcome is a function (also denoted O for simplicity) $O(s, t)$. Additional medical information F may be observed, e.g. further symptoms or information obtained after the treatment, from tests such as X-rays, blood tests, or autopsy.

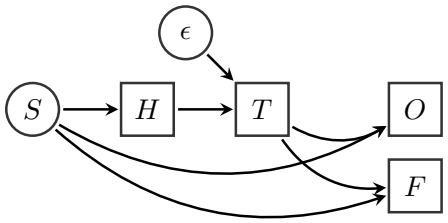

Figure 16: The medical treatment example as a structured causal model.

In this simple setting, we can charactertize the effectiveness of the treatment for an individual a patient with profile S by the Individual Treatment Effect (ITE) which is defined as the difference between the outcome under treatment and no treatment.

Definition 1 (Individual Treatment Effect)**.**

$$\begin{aligned} \textit{ITE(s)} =& \mathbb{E}[O|S = s, \mathrm{do}(T = 1)] - \mathbb{E}[O|S = s, \mathrm{do}(T = 0)] \\ =& O(s, T = 1) - O(s, T = 0) \end{aligned} \tag{8}$$

The conditional average treatment effect is the difference in outcome between the choice of $T = 1$ and $T = 0$ when averaging over all patients with the same set of symptoms $H = h$

Definition 2 (Conditional Average Treatment Effect)**.**

$$\begin{aligned} \textit{ATE}(h) =& \mathbb{E}[O|H = h, \mathrm{do}(T = 1)] - \mathbb{E}[O|H = h, \mathrm{do}(T = 0)] \\ =& \int_s p(S = s|H = h)(O(s, T = 1) - O(s, T = 0)) \end{aligned} \tag{9}$$

Since the exogenous noise (here, S) is generally not known, the ITE is typically an unknowable quantity. For a particular patient (with hidden state S), we will only observe the outcome under $T = 0$ or $T = 1$, depending on which treatment option was chosen; the counterfactual outcome will typically be unknown. Nevertheless, for a given SCM, it can be counterfactually estimated from the outcome and feedback.

Definition 3 (Counterfactually Estimated Individual Treatment Effect)**.**

$$\textit{CF-ITE}[H = h, F = f, T = 1] = \delta(o = 1) - \int_{s'} P(S = s'|H = h, F = f, T = 1)O(s', T = 0) \tag{10}$$

$$\textit{CF-ITE}[H = h, F = f, T = 0] = \int_{s'} P(S = s'|H = h, F = f, T = 1)O(s', T = 0) - \delta(o = 1) \tag{11}$$

In general the counterfactually estimated ITE will not be exactly the ITE, since there may be remaining uncertainty on s. However, the following statements relate CF-ITE, ITE and ATE:

- If S is identifiable from O and F with probability one, then the counterfactually-estimated ITE is equal to the ITE.
- The average (over S, conditional on H) of the ITE is equal to the ATE.
- The average (over S and F, conditional on H) of CF-ITE is equal to the ATE.

Assimilating O to a reward, the above illustrates that the ATE (equation 9) essentially corresponds to a difference of Q functions, the ITE (equation 8) to a difference of returns under common randomness, and the counterfactual ITE to CCA-like

advantage estimates. In contrast, the advantage estimate $G_t - V(H_t)$ is a difference between a return (a sample-level quantity) and a value function (a population-level quantity, which averages over all individuals with the same medical history H); this discrepancy explains why the return-based advantage estimate can have very high variance.

As mentioned previously, for a given joint distribution over observations, rewards and actions, there may exist distinct SCMs that capture that distribution. Those SCMs will all have the same ATE, which measures the effectiveness of a policy on average. But they will generally have different ITE and counterfactual ITE, which, when using model-based counterfactual policy gradient estimators, will lead to different estimators. Choosing the 'wrong' SCM will lead to the wrong counterfactual, and so we may wonder if this is a cause for concern for our methods.

We argue that in terms of learning optimal behaviors (in expectation), estimating inaccurate counterfactual is not a cause for concern. Since all estimators have the same expectation, they would all lead to the correct estimates for the effect of switching a policy for another, and therefore, will all lead to the optimal policy given the information available to the agent. In fact, one could go further and argue that for the purpose of finding good policies in expectations, we should only care about the counterfactual for a precise patient inasmuch as it enables us to quickly and correctly taking better actions for future patients for whom the information available to make the decision (H) is very similar. This would encourage us to choose the SCM for which the CF-ITE has minimal variance, regardless of the value of the true counterfactual. In the next section, we elaborate on an example to highlight the difference in variance between different SCMs with the same distribution and optimal policy.

G.2. Betting against a fair coin

We begin from a simple example, borrowed from (Pearl, 2009b), to show that two SCMs that induce the same interventional and observational distributions can imply different counterfactual distributions. The example consists of a game to guess the outcome of a fair coin toss. The action A and state S both take their values in $\{h, t\}$. Under model $\mathbf{I}$, the outcome O is 1 if $A = S$ and 0 otherwise. Under model $\mathbf{II}$, the guess is ignored, and the outcome is simply $O = 1$ if $S = h$. For both models, the average treatment effect $E[O|A = h] - E[O|A = t]$ is 0 implying that in both models, one cannot do better than random guessing. Under model $\mathbf{I}$, the counterfactual for having observed outcome $O = 1$ and changing the action, is always $O = 0$, and vice-versa (intuitively, changing the guess changes the outcome). Therefore, the ITE is ± 1. Under model $\mathbf{II}$, all counterfactual outcomes are equal to the observed outcomes, since the action has in fact no effect on the outcome. The ITE is always 0.

In the next section, we will next adapt the medical example into a problem in which the choice of action does affect the outcome. Using the CF-ITE as an estimator for the ATE, we will find how the choice of the SCM affects the variance of that estimator (and therefore how the choice of the SCM should affect the speed at which we can learn which is the optimal treatment decision).

G.3. Medical example

Take the simplified medical example from Fig.16, where a population of patients with the same symptoms come to the doctor, and the doctor has a potential treatment T to administer. The state S represents the genetic profile of the patient, which can be one of three $\{\text{GENE}_\text{A}, \text{GENE}_\text{B}, \text{GENE}_\text{C}\}$ (each with probability $1/3$). We assume that genetic testing is not available and that we do not know the value of S for each patient. The doctor has to make a decision whether to administer drugs to this population or not, based on repeated experiments; in other words, they have to find out whether the average treatment effect is positive or not. We consider the two following models:

- In model $\mathbf{I}$, patients of type GENE_A always recover, patients of type GENE_C never do, and patients of type GENE_B recover if they get the treatment, and not otherwise; in particular, in this model, administering the drug never hurts.
- In model $\mathbf{II}$, patients of type GENE_A and GENE_B recover when given the drug, but not patients of type GENE_C; the situation is reversed (GENE_A and GENE_B patients do not recover, GENE_C do) when not taking the drug.

In both models - the true value of giving the drug is $2/3$, and not giving the drug $1/3$, which leads to an ATE of $1/3$. For each model, we will evaluate the variance of the CF-ITE, under one of the four possible treatment-outcome pair. The results are summarized in table 5. Under model $\mathbf{A}$, the variance of the CF-ITE estimate (which is the variance of the advantage estimate used in CCA-PG gradient) is $1/6$, while it is 1 under model $\mathbf{B}$, which would imply $\mathbf{A}$ is a better model to leverage counterfactuals into policy decisions.

Treatment	Outcome	Type	CF-Prob.		CF-O		ITE		CF-V		CF-ITE		Var
Drug	Cured	GENE_A	1/2	1/2	1	0	0	+1					
		GENE_B	1/2	1/2	0	0	+1	+1	1/2	0	1/2	1	
		GENE_C	0	0									1/6
	Not cured	GENE_A	0	0									1
		GENE_B	0	0					0	1	0	-1	
		GENE_C	1	1	0	1	0	-1					
No Drug	Cured	GENE_A	1	0	1	1	0	0					
		GENE_B	0	0					1	0	0	1	
		GENE_C	0	1	0	0	1	1					1/6
	Not cured	GENE_A	0	1/2	1	1	-1	-1					1
		GENE_B	1/2	1/2	1	1	-1	-1	1/2	1	-1/2	-1	
		GENE_C	1/2	0	0	0	0	0					

Table 5: CCA-PG variance estimates in the medical example. CF-Probs. Red value are estimates for model **I**, blue ones are for model **II**. CF-Prob denotes posterior probabilities of the genetic state S given the treatment T and outcome O. CF-O is the counterfactual outcome. The ITE is the individual treatment effect (difference between outcome and counterfactual outcome). CF-V is the counterfactual value function, computed as the average of CF-O under the posterior probabilities for S. CF-ITE is the counterfactual advantage estimate (difference between O and CF-V). Var is the variance of CF-ITE under the prior probabilities for the outcome.