A Graph Details

In the following, we provide more details on all the considered Directed Acyclic Graphs (DAGs).

A.1 Linear Structural Equation model

We consider the following Linear Structural Equation model (SEM),

$$\begin{aligned} S &= U_S \\ A &= \alpha S + U_A \\ R &= \beta_1 S + \beta_2 A + U_R, \end{aligned} \tag{1}$$

where U_S, U_A and U_R are the exogenous error terms. The coefficients α, β_1 and β_2 are the path coefficients or the structural parameters, and carry causal information. For example, β_2 stands for the change in R induced by raising A one unit, while keeping all other variables constant. In terms of do-calculus, β_2 can be interpreted as the slope $\beta_2 = \delta/\delta a \mathbb{E}[R \mid do(a), do(s)]$. The corresponding DAG is illustrated in Figure 2a.

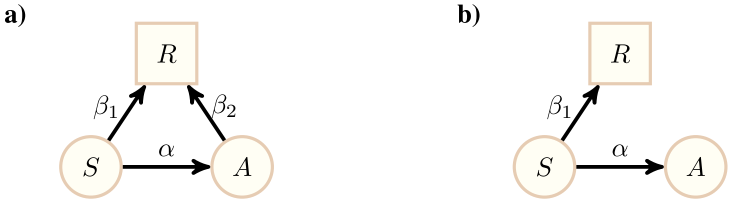

Figure 2: Example of a Directed Acyclic Graph (DAG) for a $T = 1$ setting where (a) variable R serves as a collider, and (b) R is not a collider. Variables conditioned on are depicted as rectangles.

A.2 Complete Graph Environment (CG1)

We consider a finite horizon MDP with a time horizon of $T = 7$. In this setting, all variables are endogenous, as represented by the corresponding DAG in Figure 3. The behavior policy takes as input the previous time-point action and state variables. The action space is binary, with $A = 1$ corresponding to “assign action" and $A = 0$ indicating “don’t assign action". The data-generating process (DGP) corresponding to the state variables is as follows,

$$\begin{aligned} S_1 &\sim Normal(0, \sigma) \\ S_t &\sim Normal(\mu_{a,t}, \sigma), \text{ for } 1 < t \leq 3 \\ S_t &\sim Normal(\mu_{b,t}, \sigma), \text{ for } t \leq T \end{aligned}$$

where $\mu_{a,t} = -0.7A_{t-1} + 0.4S_{t-1}$ and $\mu_{b,t} = 0.4A_{t-1} + 0.4S_{t-1}$. The reward at time t is equal to 1 if S_t exceeds the third quantile of the asymptotic distribution of S_t. Otherwise, it is 0. In the DAG shown in Figure 3, both states and actions are colliders. As rewards are descendants of colliders, conditioning on rewards will have the same effect as conditioning on the collider directly. By conditioning on the future, we introduce spurious associations between states and actions at earlier time points (which negatively affects the return) and later time points (which positively affects the return). As a result, the optimal policy learned by GCRL will be biased because it fails to correctly learn the optimal policy at the early time points.

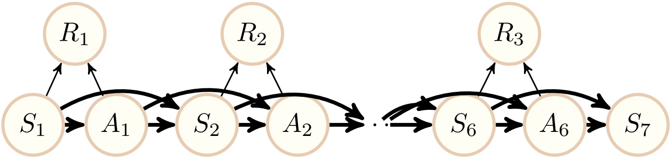

Figure 3: DAG corresponding to the Complete Graph environment (CG1) with a horizon of $T = 7$.

A.3 Incomplete Graph Environment (IG1)

We consider a finite horizon DGP with a time horizon of $T = 7$. In this setting, all variables are endogenous except for an *unknown*, exogenous variable $\epsilon \sim Normal(1, 0.2)$. The corresponding DAG is depicted in Figure 4. The behavior policy takes as input the previous time-point state and action variable. Similar to the previous scenario (CG1), the action space is binary. The data-generating process corresponding to the state variables is as follows,

$$S_1 \sim 0.8\epsilon$$
$$S_t \sim Normal(\mu_t, \sigma), \text{ for } 1 < t \leq T$$

where $\mu_t = -0.9A_{t-1} - 0.9S_{t-1} + 5\epsilon$. The reward at time t is equal to 1 if S_t exceeds the third quantile of the asymptotic distribution of S_t. Otherwise, it is 0. In the DAG depicted in Figure 4, states are colliders. It's important to note that ϵ positively influences the outcome, while actions have a negative impact on the trajectory's return. By conditioning on the future, we introduce an association between actions and ϵ, which results in a biased optimal policy learned by GCRL.

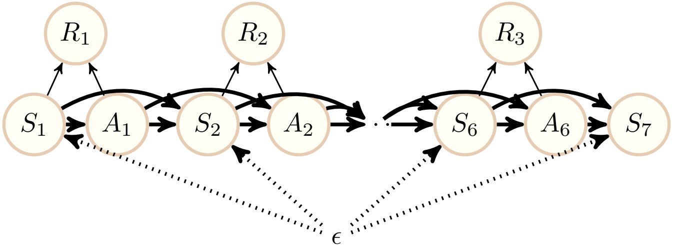

Figure 4: DAG corresponding to the Incomplete Graph environment (IG1) with $T = 7$. Dotted arrows represent paths from unknown, exogenous (unobserved) variable to endogenous (observed) state variables.

B Motivation for Examining Causality of GCRL

What if the collected trajectories are not all from an expert, so now we have data with $R = 1$ and $R = 0$? Let's say we aim to learn $P(a \mid s, r)$ as in GCRL, where $R = r$. Note that $P(a \mid s, r) = P(a, s \mid r)/P(r \mid s)$, thus we use the same approach as in the previous example, just with $R = r$. For both, we have unblocked extraneous information between S and A due to conditioning on the collider R, which falls under the rubric of spurious association [12]. Conditioning on R during data selection process, or while trying to learn population conditional probabilities, can induce spurious association between it's parents, S and A. For instance, while correlation between A and S was 0.26 when we condition on $R = 1$, it is 0.3 conditional on $R = 0$. In language of Pearl's do-calculus, there is no $do()$ operator on the state variable [12]. In comparison, in Figure 2b, R is no longer a collider. Now we can recover $P(A = 1 \mid S = 0)$ from $P(s, a \mid R = 1)$, and correlation between A and S conditional on $R = 1$ or $R = 0$ remains 0.23.

B.1 Recoverability in the $T = 1$ Setting

The following definition and notation follows from Bareinboim and Pearl [1]. We refer to DAG in Figure 2 with $T = 1$ and (S, A, R) structure, denoted as the G_r graph.

Definition 1 (r-Recoverability). *In the context of a causal graph denoted as G_r representing the selection mechanism, we define that the distribution $Q = P(a \mid s)$ is considered r-recoverable from selection-biased data within G_r if the assumptions inherent in the causal model allow Q to be expressed in terms of the distribution under selection bias, denoted as $P(a, s \mid R = 1)$.*

Lemma 1. *$P(a \mid s)$ is not r-recoverable from a DAG in Figure 2.*

Proof. The proof follows immediately from the subgraph Figure 1d considered in [2], where $G_r \setminus \{S \to R\}$. As the extra edge can be inactive in a compatible parametrization [12], lack of r-recoverability in $G_r \setminus \{S \to R\}$ proves $P(a \mid s)$ is not r-recoverable in G_r. □

C Experiment Results

C.1 Is stochasticity driving performance?

Recent studies indicate that GCRL algorithms struggle in stochastic environments [7, 4]. In our analysis, we evaluate the performance of GCRL and FQI across different levels of variability, represented by the parameter σ, in the DGPs of CG1 and IG1. The results, based on various training dataset sizes, are presented in Figure 5, revealing that FQI consistently outperforms GCRL under all levels of stochasticity.

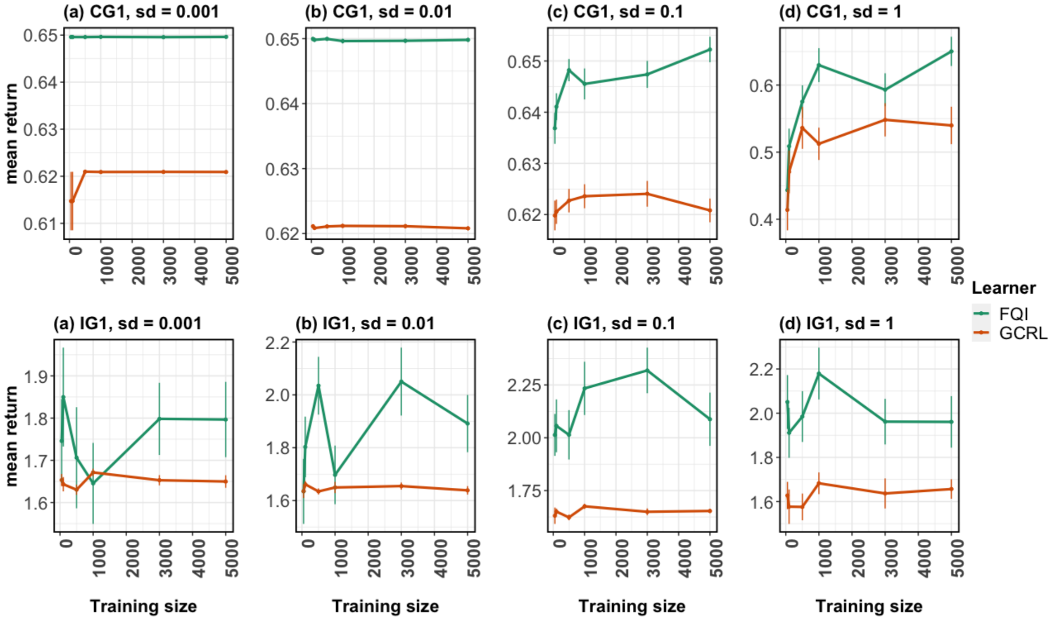

Figure 5: Mean return for CG1 and IG1 Data Generating Process (DGP) at $t = 7$ and its corresponding standard error, calculated over 100 Monte Carlo (MC) iterations. In the upper panels (a)-(d), we illustrate the CG1 DGP, while in the lower panels (a)-(d), we depict the IG1 DGP under different levels of $\sigma = 0.001, 0.01, 0.1, 1$, indicating increasing stochasticity in the process. The training dataset sizes considered are $50, 100, 500, 1000, 3000, 5000$, and a validation size of 20 is used for all cases.

C.2 Do we need different policy estimators?

Practical recommendations suggest that simple implementations can achieve competitive performance, if not better, compared to more complex architectures and value-based RL methods [7]. Other point to importance of complex neural network architectures as, even if the behavior policy is simple, conditional policy learned by GCRL might not be [11, 5]. In our analysis, we explore various choices for policy estimation, including: (1) simple main terms generalized linear model (glm), (2) Super Learner (SL), an ensemble learner based on cross-validation, (3) high-capacity feed-forward fully-connected neural network and (4) high-capacity neural network with regularization. The SL is a convex combination of predictions made by glm, generalized additive model, shallow neural network, regularized gradient boosting and random forest [14, 6, 16, 15]. Figure 6 presents the results for different policy estimators at $\sigma = 0.1$. It demonstrates that FQI consistently outperforms GCRL across all considered policy estimators and sample sizes.

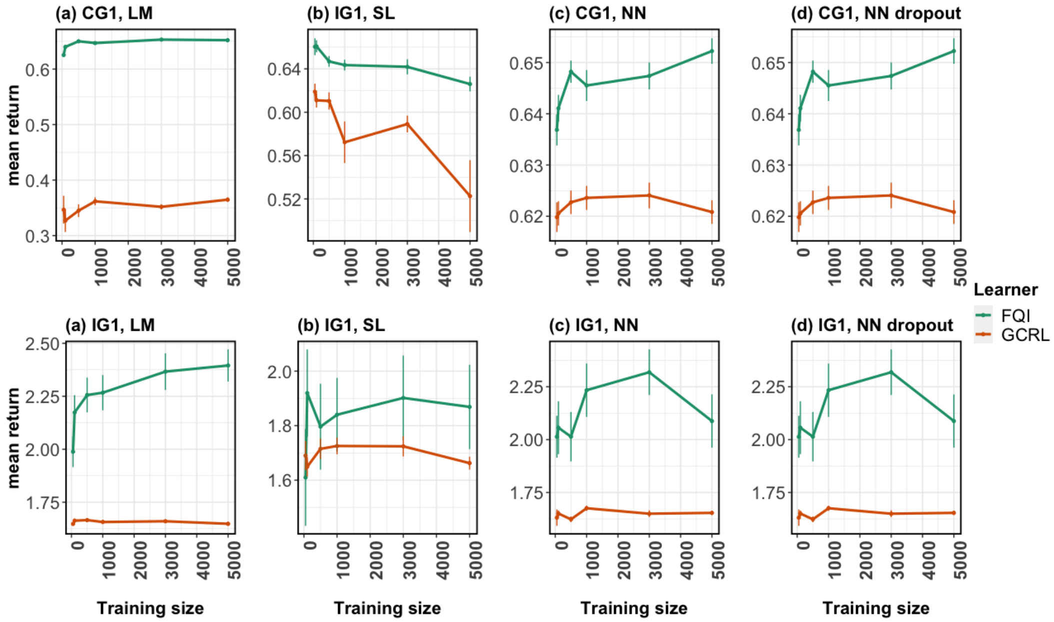

Figure 6: Mean return for CG1 and IG1 Data Generating Process (DGP) at $\sigma = 0.1$ and $t = 7$ with its corresponding standard error, calculated over 100 Monte Carlo (MC) iterations. In the upper panels (a)-(d), we illustrate the CG1 DGP, while in the lower panels (a)-(d), we depict the IG1 DGP with policy estimated using different estimators: linear models (LM), ensemble learner (SL), Neural Network (NN) and Neural Network with dropout. The training dataset sizes considered are $50, 100, 500, 1000, 3000, 5000$, and a validation size of 20 is used for all cases.

D Experiment Details

In Table 1, we provide details about the neural network architecture used in all experiments, unless a different algorithm is explicitly mentioned. In the last row of Table 1 we specifically note that we investigated regularization through dropout as a separate estimator in our exploration of various conditional policy estimators. The table also enumerates all the algorithms included in the ensemble learner's library, known as the Super Learner (SL) [14]. The Super Learner library comprised the following algorithms: (1) generalized linear model (glm), (2) single layer neural network, (3) generalized additive model, (4) random forest and (5) regularized gradient boosting. [6, 16, 15]. We considered different configurations of random forests and gradient boosting based on their hyperparameters, such as the number of trees, maximum depth, and eta. The Super Learner employed 10-fold cross-validation.

In Table 2, we provide a comprehensive list of simulation parameters. Each experiment was conducted independently 100 times, corresponding to 100 Monte Carlo (MC) simulations or iterations (independent experiments). During each iteration, we trained both a GCRL and FQI algorithm using training sets of various sizes, where ($n = 50, 100, 500, 1000, 3000, 5000$). For every experiment, we used a validation set consisting of 20 trajectories. The final reported return is the average over 100 Monte Carlo simulations. To achieve the desired return values for GCRL, we set the target return to be 0.7 for CG1 and 2.4 for IG1. These target values were determined based on the asymptotic distribution consistent with the dynamics of the CG1 and IG1 DGP at the end of each trajectory. Specifically, they correspond to the upper tails (3rd quantile) of the asymptotic distribution observed in CG1 and IG1 DGPs and are supported by the training data used in each experiment.

Table 1: Neural network architecture, ensemble learner specification and design parameters used for considered experiments.

Hyperparameter	**Value**	**Environment**
Hidden Layers	2	All
Layer Width	1024	All
Nonlinearity	ReLU	All
Learning rate	1e-3	All
Epochs	20	All
Dropout	0	All
	0.1	All
Ensemble Learner	glm	All
	gam	All
	neural network	All
	random forest	All
	xgboost	All
CV	10	All

Table 2: Simulation parameters used for considered experiments.

Hyperparameter	**Value**	**Environment**
Number of MC iterations	100	All
Training size	50	All
	100	All
	500	All
	1000	All
	3000	All
	5000	All
Validation size	20	All
Goal max	0.7	CG1
Goal max	2.4	IG1