Content:

- Section A contains the proofs that were omitted from the main text.
- Section B documents all experiments used in the paper.
- Section C provides additional results as robustness checks.

A. Proofs of Lemmas and Propositions

A.1. Proof of Proposition 1

Recall from the proof sketch that,

$$p(u|x,a) = p_u(u)\frac{\pi_b(a|x,u)}{\pi_b(a|x)}.$$

Proof.

$$\begin{aligned}
\mathbb{E}[f(x,a,x')|x,a] &= \sum_{u\in\mathcal{U}} p_u(u) \sum_{x'\in\mathcal{X}} p(x'|x,u,a)f(x,a,x') \\
&= \sum_{u\in\mathcal{U}} p(u|x,a)\frac{\pi_b(a|x)}{\pi_b(a|x,u)} \sum_{x'\in\mathcal{X}} p(x'|x,u,a)f(x,a,x') \\
&= \sum_{u\in\mathcal{U}} \sum_{x'\in\mathcal{X}} p(u,x'|x,a)\frac{\pi_b(a|x)}{\pi_b(a|x,u)} f(x,a,x') \\
&= \mathbb{E}_{\mathcal{D}_{\pi_b}}\left[\frac{\pi_b(a|x)}{\pi_b(a|x,u)} f(x,a,x') \middle| x,a\right].
\end{aligned}$$

□

A.2. Proof of Lemma 2

Proof. The first claim follows by definition. For the second claim, consider the nominal transition probabilities, $\hat{P}(x'|x,a)$:

$$\begin{aligned}
\hat{P}(x'|x,a) &= \sum_{u\in\mathcal{U}} p(u|x,a)P(x'|x,u,a) \\
&= \sum_{u\in\mathcal{U}} p_u(u)\frac{\pi_b(a|x,u)}{\pi_b(a|x)} P(x'|x,u,a) \\
&= \frac{1}{\pi_b(a|x)} \sum_{u\in\mathcal{U}} p_u(u)\pi_b(a|x,u)P(x'|x,u,a).
\end{aligned}$$

□

A.3. Proof of Lemma 3

Proof. Recall our original variable $\mathcal{B}$ contains the possible values of $\pi_b(a|x,u)$. In particular, for all $\pi_b(a|x,u) \in \mathcal{B}$:

$$\alpha(x,a) \le \frac{\pi_b(a|x)}{\pi_b(a|x,u)} \le \beta(x,a),$$

Our new parameters are:

$$\begin{aligned}
g(x,a,x') &= \sum_{u\in\mathcal{U}} \left(\frac{p(u|x,a)P(x'|x,u,a)}{\hat{P}(x'|x,a)}\right) \frac{1}{\pi_b(a|x,u)} \\
&= \sum_{u\in\mathcal{U}} \left(\frac{p(u|x,a)P(x'|x,u,a)}{\sum_{u'\in\mathcal{U}} p(u'|x,a)P(x'|x,u',a)}\right) \frac{1}{\pi_b(a|x,u)}.
\end{aligned}$$

Note that: the terms for each u in parentheses on the last line form a density over $\mathcal{U}$. Therefore:

$$\alpha(x,a) \leq \frac{\pi_b(a|x)}{\pi_b(a|x,u)} \leq \beta(x,a), \forall u \implies \alpha(x,a) \leq \pi_b(a|x)g(x,a,x') \leq \beta(x,a).$$

For the second claim:

$$\begin{aligned} g(x,a,x') &= \sum_{u\in\mathcal{U}} \left(\frac{p(u|x,a)P(x'|x,u,a)}{\hat{P}(x'|x,a)} \right) \frac{1}{\pi_b(a|x,u)} \\ &= \frac{1}{\hat{P}(x'|x,a)} \sum_{u\in\mathcal{U}} p_u(u) \frac{\pi_b(a|x,u)}{\pi_b(a|x)} P(x'|x,u,a) \frac{1}{\pi_b(a|x,u)} \\ &= \frac{1}{\hat{P}(x'|x,a)\pi_b(a|x)} \sum_{u\in\mathcal{U}} p_u(u) P(x'|x,u,a) \\ &= \frac{P(x'|x,a)}{\hat{P}(x'|x,a)\pi_b(a|x)}. \end{aligned}$$

So the true transition probabilities $P(x'|x,a)$ are exactly equal to $\pi_b(a|x)g(x,a,x')\hat{P}(x'|x,a)$ and the claim follows since $P(\cdot|x,a)$ is a density. □

A.4. Proof of Theorem 1

Proof. The inequality is by definition. We prove the equality of our reparameterization by $g(x,a,x')$. For any $\pi_b(a|x,u) \in \mathcal{B}_{xa}$:

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_{\pi_b}} \left[\frac{\pi_b(a|x)}{\pi_b(a|x,u)} f(x,a,x') \middle| x,a \right] &= \sum_{u\in\mathcal{U}} \sum_{x'\in\mathcal{X}} p(u|x,a)P(x'|x,u,a) \frac{\pi_b(a|x)}{\pi_b(a|x,u)} f(x,a,x') \\ &= \sum_{x'\in\mathcal{X}} f(x,a,x')\pi_b(a|x) \sum_{u\in\mathcal{U}} p(u|x,a)P(x'|x,u,a) \frac{1}{\pi_b(a|x,u)} \\ &= \sum_{x'\in\mathcal{X}} f(x,a,x')\pi_b(a|x)\hat{P}(x'|x,a)g(x,a,x') \\ &= \sum_{u\in\mathcal{U}} \sum_{x'\in\mathcal{X}} p(u|x,a)P(x'|x,u,a)\pi_b(a|x)g(x,a,x')f(x,a,x') \\ &= \mathbb{E}_{\mathcal{D}_{\pi_b}} \left[\pi_b(a|x)g(x,a,x')f(x,a,x') \middle| x,a \right] \end{aligned}$$

and by definition every $g(x,a,x') \in \tilde{\mathcal{B}}_{xa}$ corresponds to a $\pi_b(a|x,u) \in \mathcal{B}_{xa}$. So we've shown every value of the objective of the first minimization problem for some $\pi_b(a|x,u)$ corresponds to the value of the second minimization problem for some $g(x,a,x')$ and vice-versa. □

B. Experimental Details

B.1. Toy Environment

For our experiments, we use four different environments. The first is a toy MDP with 3 states, 2 actions, and a single binary unobserved state. The transition probabilities between the three states are illustrated in Appendix Figures 1 and 2 for $u = 0$ and $u = 1$ respectively. Entering the "Bad" state has a reward of -1, entering the "Neutral" state has a reward of 0, and entering the "Good" state has a reward of 1 and is absorbing. In this example, transitions do not depend on actions to emphasize how confounded estimates can nonetheless suggest certain policies have higher expected values than others.

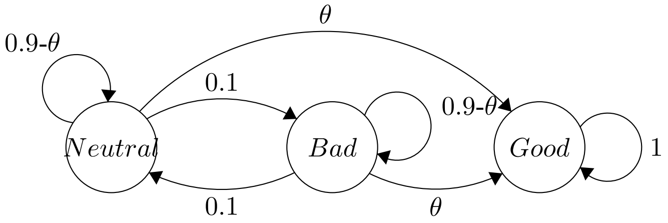

Figure 1. Transitions for unobserved state $u = 0$ (regardless of action)

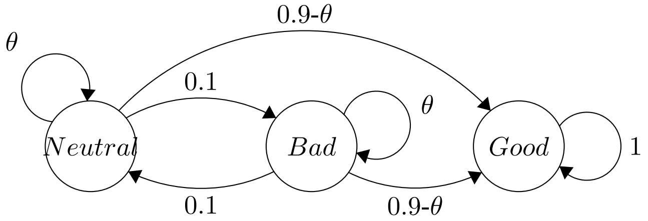

Figure 2. Transitions for unobserved state $u = 1$ (regardless of action)

The dynamics are parameterized by $\theta \in [0, 0.9]$. If $\theta = 0.45$, the transitions are unconfounded. If θ is closer to 0 or 0.9, then the transitions are strongly confounded.

We also parameterize the behavior policy over two possible actions $\mathcal{A} = \{0, 1\}$:

$$\pi_b(a = 0|x, u = 0) = 1 - \phi$$
$$\pi_b(a = 1|x, u = 0) = \phi$$
$$\pi_b(a = 0|x, u = 1) = \phi$$
$$\pi_b(a = 1|x, u = 1) = 1 - \phi$$

The parameter $\phi \in [0, 1]$ controls the correlation between a and u. For ϕ closer to zero, then action $a = 1$ occurs more often when $u = 1$. For ϕ closer to one, then action $a = 0$ occurs more often when $u = 0$. For $\phi = 0.5$, the policy is not confounded (essentially a randomized control trial). The two parameters ϕ and θ will imply corresponding values of Γ and Δ for the Policy and Transition Confounding assumptions.

For our experiments, we set $\phi = 0.25$ and $\theta = 0.1$ which is a moderate amount of policy confounding and a large amount of transition confounding. We use a horzion of $T = 5$ and the evaluation policy: $\pi_e(\cdot|x) = (0.3, 0.7), \forall x$.

B.2. Adding Confounding to OPE Benchmarks

The other three environments we take from (Voloshin et al., 2019), who make their benchmarks available online at `https://github.com/clvoloshin/OPE-tools`: Graph, Discrete-MC, and Gridworld. First, we modify these to introduce unobserved confounding. We implement transformations which take an environment and a behavior policy, and return a new confounded environment and policy. We implement two such transformation, one based on the rewards, and one based on a value function.

The reward transformation finds all transitions $P(x'|x, a)$ such that $R(x, a, x') > 0$ and those such that $R(x, a, x') \leq 0$. Then, we generate two new transition matrices $P(x'|x, u = 0, a)$ and $P(x'|x, u = 1, a)$. For the $u = 0$ matrix, we upweight transitions with a positive R and downweight transitions with a negative R. For the $u = 1$ matrix, we do the opposite. Likewise, for the behavior policy. For $R(x, a, x') > 0$, we upweight the probability of $\pi_b(a|x, u = 0)$ relative to $\pi_b(a|x)$ and downweight the probabilities of $\pi_b(a|x, u = 0)$ for $R(x, a, x') \leq 0$. The opposite is done for $u = 1$.

For the value transformation, we take any value function V, for example the optimal value function or the value function of a uniformly-random policy. For each state x, we find the action with the largest and smallest value of $P(\cdot|x, a)^T V(\cdot)$, call them $a_{\max}$ and $a_{\min}$. We then generate two new transition matrices for $u = 0$ and $u = 1$. For $u = 0$, we shift the transitions for x towards that of $a_{\max}$. For $u = 1$ we shift towards $a_{\min}$. For the behavior policy, when $u = 0$ we increase the probability of $\pi_b(a_{\max}|x)$ and decrease the probability of other actions and vice-versa for $u = 1$.

For both strategies, we make sure that the confounded behavior policy has a non-zero probability of taking each action from each state to guarantee overlap.

B.3. OPE Benchmark Details

We provide details on the three environments from (Voloshin et al., 2019). In general, the behavior policy is created by applying the transformations described above, and the evaluation policy is explicitly chosen to have a higher value as we explain in the main paper. The parameters of the OPE benchmarks and the choice of policies and horizon is documented here. We make some minor modifications where necessary which are preceded by the symbol (*).

ope-graph

For the Graph benchmark, we use the following parameters:

max_length: 4
make_pomdp: False
transitions_deterministic: False
sparse_rewards: False
stochastic_rewards: False

For π_b we start with $\pi_b(\cdot|x) = [0.6, 0.4], \forall x$ and then we generate a confounded policy using the reward transformation. We use $\pi_e(\cdot|x) = [0.3, 0.7]$. We use discount factor $\gamma = 0.99$ and horizon $T = 4$.

ope-mc

For the Discrete-MC benchmark, (*) we add an addition parameter "slippage" to make the transitions stochastic. We the use the parameter values:

n_left: 10
n_right: 10
slippage: 0.25

For π_b we start with $\pi_b(\cdot|x) = [0.6, 0.4], \forall x$ and then we generate a confounded policy using the value transformation. We use $\pi_e(\cdot|x) = [0.15, 0.85], \forall x$. We use discount factor $\gamma = 0.99$ and horizon $T = 20$.

ope-gridworld

For the Gridworld benchmark, (*) we use a smaller 4x4 grid, but otherwise keep the environment identical. We use the parameters value:

slippage: 0.05

For π_b we start with $\pi_b(\cdot|x) = [0.4, 0.1, 0.1, 0.4], \forall x$ and then we generate a confounded policy using the value transformation. We use $\pi_e(\cdot|x) = [0.4, 0.1, 0.4, 0.1], \forall x$. We use discount factor $\gamma = 0.99$ and horizon $T = 8$.

B.4. Runtime and Packages

All experiments were performed on a ThinkPad T470p Laptop with 32 GB of RAM.

Our implementation of FQE took on average, 0.2, 0.2, 0.7, and 0.3 seconds per iteration for the toy, Graph, Discrete-MC, Gridworld environments respectively.

Our confounded-FQE bound took on average, 0.2, 0.2, 1.2, and 0.8 seconds per iteration.

Our robust MDP bound took on average, 0.1, 0.3, 1.6, and 2.2 seconds per iteration.

The packages used for the code were:
Anaconda Python 3.8 (`https://www.anaconda.com/products/individual`)
Gurobi 9.1.1 (`https://www.gurobi.com/products/gurobi-optimizer/`).

C. Additional Experiments and Robustness Checks

C.1. Additional Figure 2 Results

Figure 2 in the main paper only includes the the ope-graph and ope-mc environments. For completeness we provide the remaining results from the toy and ope-gridworld environments. See Appendix Figure 3.

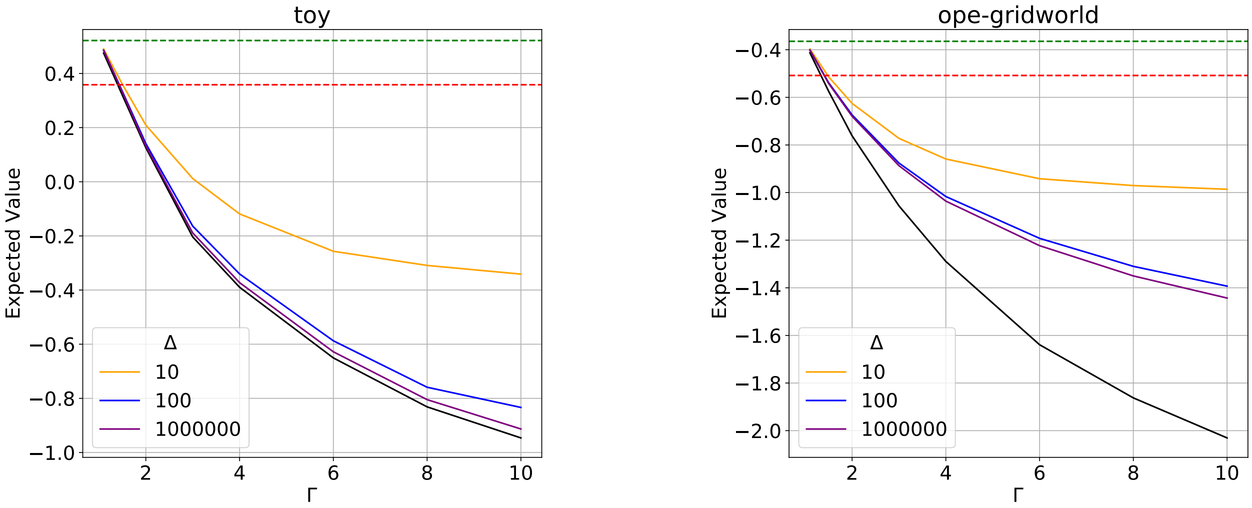

Figure 3.

C.2. NKYB Comparison with the Same Γ

The comparison with NKYB in the main paper has to deal with the fact that our approach and their approach use different sensitivity models. In particular, our sensitivity model uses the bounds, $\forall x, a, u$:

$$\frac{1}{\Gamma} \leq \left(\frac{\pi_b(a|x,u)}{1-\pi_b(a|x,u)}\right) \Big/ \left(\frac{\pi_b(a|x)}{1-\pi_b(a|x)}\right) \leq \Gamma$$

whereas NKYB uses the bounds, $\forall x, a, u, u'$:

$$\frac{1}{\Gamma} \leq \left(\frac{\pi_b(a|x,u)}{1-\pi_b(a|x,u)}\right) \Big/ \left(\frac{\pi_b(a|x,u')}{1-\pi_b(a|x,u')}\right) \leq \Gamma.$$

Notice that for the same value of Γ, there's is more restrictive than ours. It's roughly, but not exactly, a quadratic relationship. We handle this in the main paper by calculating the smallest valid value of Γ for each environment with respect to each sensitivity model. We call these the true sensitivity parameters. However, this means that our results could simply reflect difference between the sensitivity models for the particular confounded environments we use. Therefore, I additionally provide a comparison where we simply use the same value of Γ. Because this is quite favorable for NKYB, this serves as a useful robustness check. See Table 1 for the results.

env	Nominal	Γ	NKYB	Ours
toy	0.5207	2	0.3876	0.3590
		4	0.2507	0.1947
		10	0.0137	-0.0232
ope-graph	0.7008	2	0.4588	0.6001
		4	0.2756	0.4310
		10	0.0761	0.0875
ope-mc	-15.6941	2	-72.6004	-15.7450
		4	-107.1074	-15.9043
		10	-122.7673	-16.2092
ope-gridworld	-0.3588	2	-0.5207	-0.3922
		4	-0.7072	-0.4277
		10	-1.4157	-0.4650

Table 1.

Unlike for the true sensitivity parameters, the NKYB bounds are now comparable for the toy and ope-graph environments. They are actually between 6 and 8 percentage points better for the toy environment. In the ope-graph environment they are worse than our bounds, but only by 20 percentage points for low values of Γ and only by a tiny amount for $\Gamma = 10$. This is much better performance than for the true sensitivity parameters. However, for the slightly more complicated ope-mc and ope-gridworld environments, there bounds are still vastly worse, even given the favorable comparison. These environments are simply too sensitive to persistent confounders to achieve reasonable lower bounds.

The ope-mc case is particularly bad. Worryingly, when bootstrapping the data the lower bounds vary between -10 and -160. Not only is this an extreme amount of variance, but -10 is not a valid lower bound: it's larger than the nominal value. There may be issues involving the training of the neural network for the NKYB bounds that make the results less reliable.

C.3. Sensitivity to Confounder Distribution

In the main text, we fix the parameter $p = 0.5$. In this section, we explore sensitivity to this assumption by recomputing our lower bounds for $p \in \{0.1, 0.4, 0.45\}$. Appendix Figure 4 shows the case when $p = 0.1$. In this case, the distribution of the unobserved confounder is highly imbalanced, with $u = 0$ showing up 90% of the time. As is clear from the figure, such imbalance actually reduces uncertainty significantly. Unless there's reason to believe that the unobserved variable usually takes a single fixed value, this setting of the parameter will give overly optimistic bounds.

We find that the smallest expected values tend to occur when p is between 0.4 and 0.45, see Appendix Figures 5 and 6. However, these bounds tend to be very close to the default $p = 0.5$ case. For the toy, ope-mc, and ope-gridworld environments there is virtually zero difference. The only exception is ope-graph, where the bounds get looser for all Γ and Δ. The change is small but non-zero: slightly less than -0.1.

(Ding & VanderWeele, 2016) gets bounds on a treatment effect following a very similar strategy, but their bounds avoid treating p as a parameter and only feature analogs of Γ and Δ. It is non-trivial to adapt their approach to the MDP setting, but relaxing the assumption of fixed p would be valuable for future work.

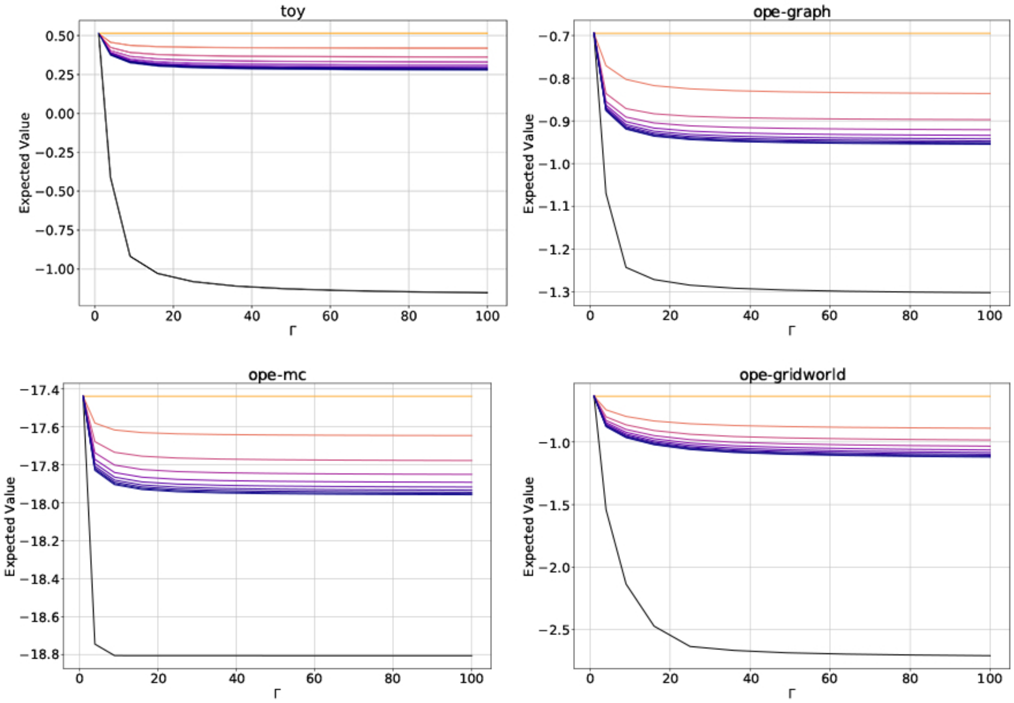

Figure 4. $p = 0.1$

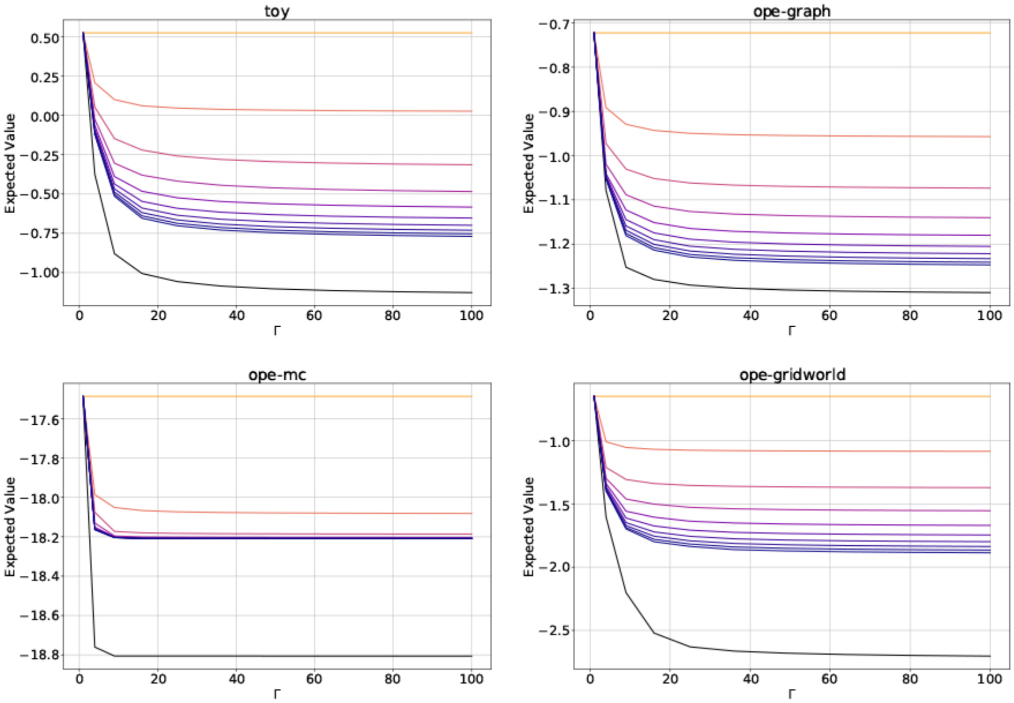

Figure 5. $p = 0.4$

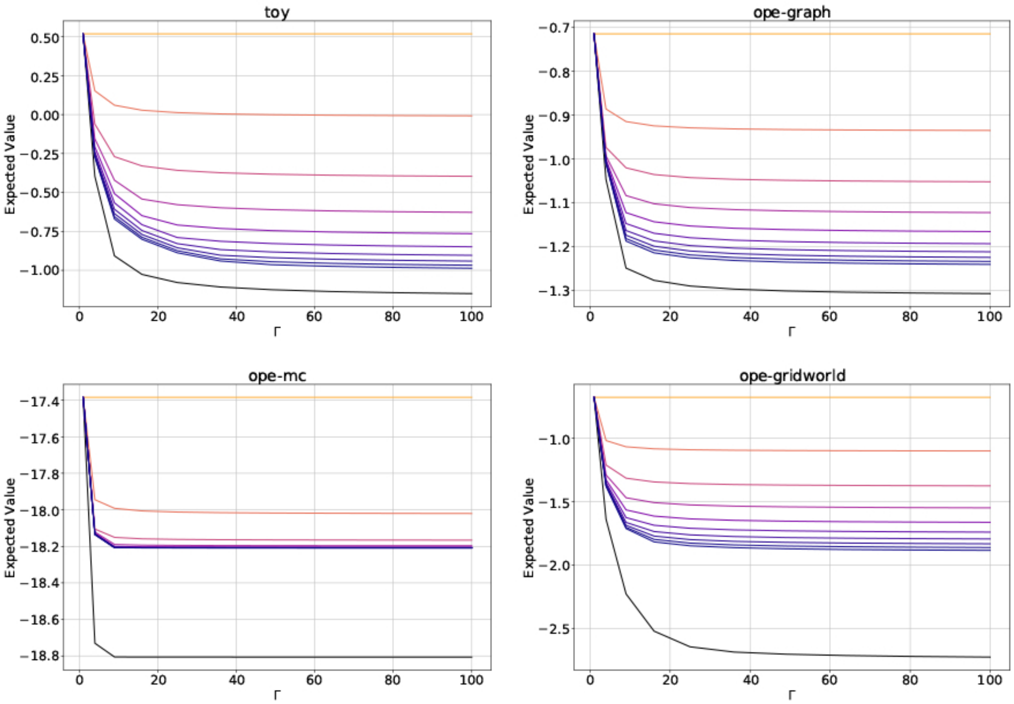

Figure 6. $p = 0.45$

References

Ding, P. and VanderWeele, T. J. Sensitivity analysis without assumptions. *Epidemiology (Cambridge, Mass.)*, 27(3):368, 2016.

Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.