Counterfactual Data Augmentation using Locally Factored Dynamics

Silviu Pitis
Elliot Creager
Animesh Garg

University of Toronto · Vector Institute

{spitis, creager, garg}@cs.toronto.edu

Neural Information Processing Systems

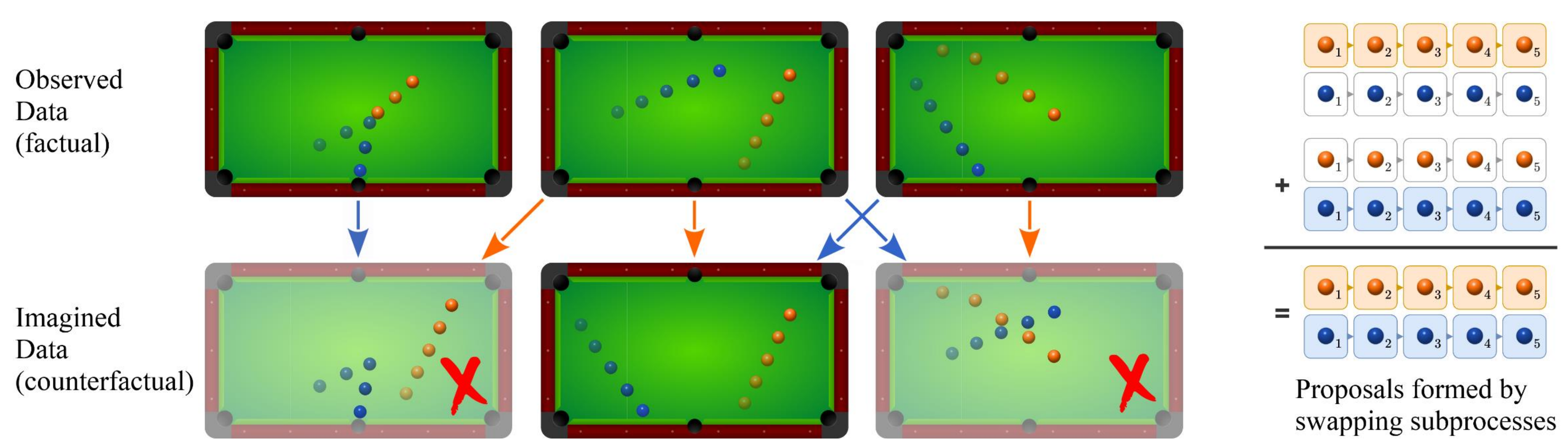
CoDA more than doubles sample efficiency in locally factored online and offline off-policy RL

Abstract

We leverage **local causal independence** to improve the sample efficiency of off-policy reinforcement learning. We do this by generating counterfactual experiences using a novel **Counterfactual Data Augmentation (CoDA)** algorithm and **Local Causal Modeling (LCM)** framework.

Contributions

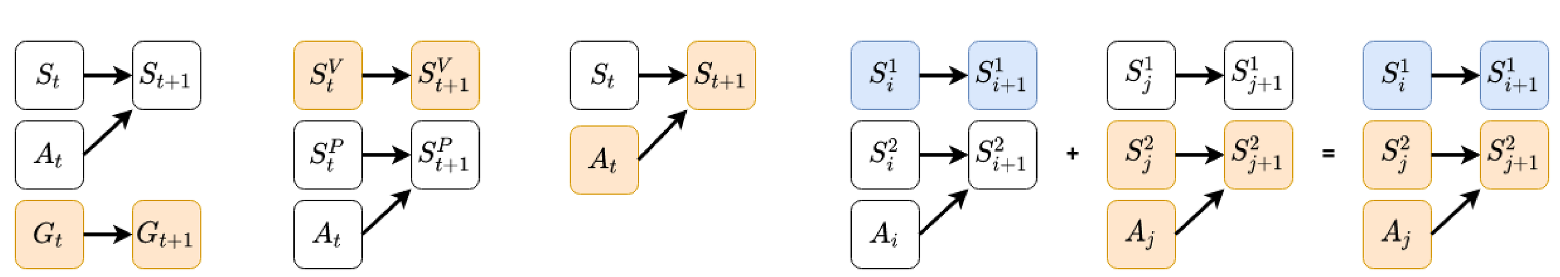
- LCMs, which condition a causal model on a local state set
- CoDA, which generalizes HER / domain randomization
- Locally conditioned causal structure discovery using a mask
- Significant performance improvements for off-policy reinforcement learning in locally factored tasks



Given 3 factual samples, knowledge of the local causal structure lets us mix and match factored subprocesses to form counterfactual samples. The first proposal is rejected because one of its factual sources (the blue ball) is not locally factored. The third proposal is rejected because it is not itself factored. The second proposal is accepted.

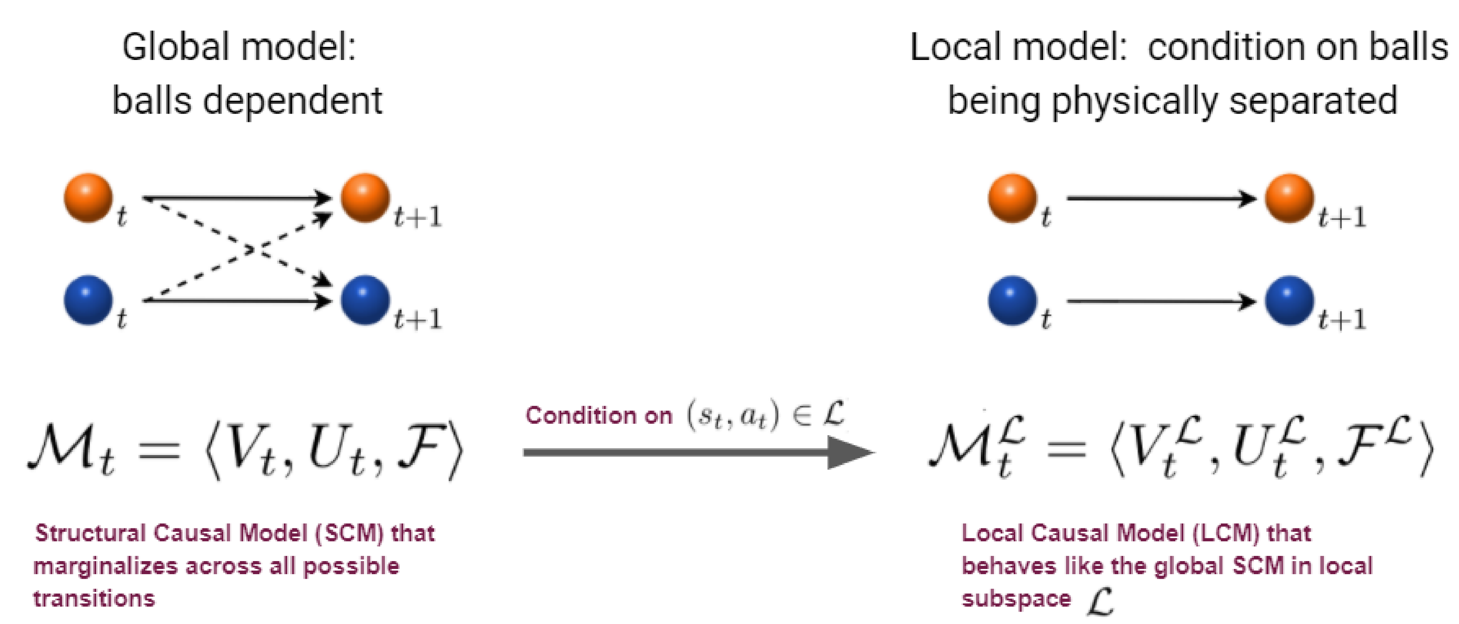
The CoDA Algorithm

Whenever a pair of transitions can be factored into the same two locally independent components, we can mix-and-match the components to form new causally valid, counterfactual samples. We call this **Counterfactual Data Augmentation (CoDA)**. This generalizes "relabeling" techniques that use global factorizations of the state, including Hindsight Experience Replay (HER) [1].



(a) Goal relabeling (b) Visual augmentation (c) Dyna (model-based) (d) Counterfactual Data Augmentation by swapping locally independent subspaces **(ours)**

Local Causal Models (LCMs)



- We use LCMs to formalize the notion of local independence
- The LCM for a subspace $\mathcal{L} \subset \mathcal{S}$ of the state space can be derived from a Structural Causal Model (SCM) by restricting the ranges of the state variables to $\mathcal{L}$
- The LCM can be used to simplify causal reasoning about interventions or counterfactuals that are within the bounds of local subspace $\mathcal{L}$.

Defining a Local Model with a Mask

To detect local independence, we adapt the Set Transformer [3] and use the product of its attention masks as a locally conditioned model of feature-wise causal dependencies in the forward dynamics. This extends the global mask-based causal discovery of GraN-DAG [2].

Example

Given a state that describes 2 pool balls with 8 features, the model outputs a binary mask representing feature-wise causal dependencies. When the balls are physically separated (i.e., locally independent), the mask will look like the one shown on the right.

Input:

```
[[ 1.23, -0.73,  1.31,  1.07],
 [-0.6 ,  2.51, -1.51, -0.89]]
```

Output:

```
[[1., 1., 1., 1., 0., 0., 0., 0.],
 [1., 1., 1., 1., 0., 0., 0., 0.],
 [1., 1., 1., 1., 0., 0., 0., 0.],
 [1., 1., 1., 1., 0., 0., 0., 0.],
 [0., 0., 0., 0., 1., 1., 1., 1.],
 [0., 0., 0., 0., 1., 1., 1., 1.],
 [0., 0., 0., 0., 1., 1., 1., 1.],
 [0., 0., 0., 0., 1., 1., 1., 1.]]
```

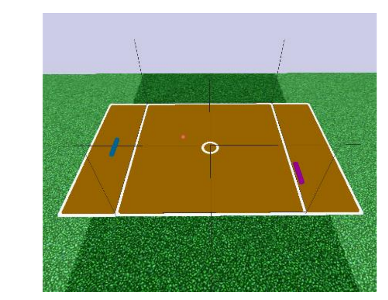
Algorithm 1 Mask-based Counterfactual Data Augmentation (CoDA)

function CODA(transition t1, transition t2):
 s1, a1, s1' ← t1
 s2, a2, s2' ← t2
 m1, m2 ← MASK(s1, a1), MASK(s2, a2)
 D1 ← COMPONENTS(m1)
 D2 ← COMPONENTS(m2)
 d ← random sample from (D1 ∩ D2)
 s̃, ã, s̃' ← copy(s1, a1, s1')
 s̃[d], ã[d], s̃'[d] ← s2[d], a2[d], s2'[d]
 D̃ ← COMPONENTS(MASK(s̃, ã))
 return (s̃, ã, s̃') **if** d ∈ D̃ **else** ∅

function MASK(state s, action a):
 Returns $(n+m) \times (n)$ matrix indicating if the n next state components (columns) locally depend on the n state and m action components (rows).

function COMPONENTS(mask m):
 Using the mask as the adjacency matrix for $\mathcal{G}^{\mathcal{L}}$ (with dummy columns for next action), finds the set of connected components $C = \{C_j\}$, and returns the set of independent components $D = \{\mathcal{G}_i = \bigcup_k \mathcal{C}_k^i \mid \mathcal{C}^i \subset \text{powerset}(C)\}$.

Results: Batch RL

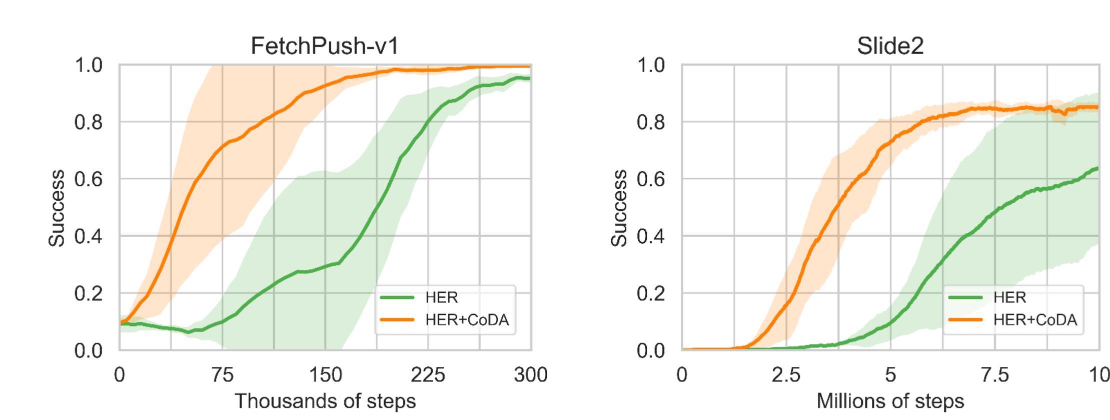


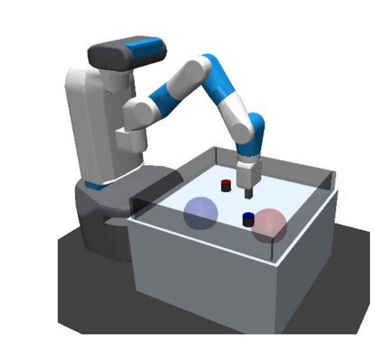
- Continuous control **Pong** environment
- Expand a batch dataset by (1) training our mask model on the provided data, then (2) using the model to generate CoDA data.
- Using CoDA data gives **2x effective data size**, and gives a **3x performance boost** at small data sizes

$\lvert\mathcal{D}\rvert$ (1000s)	Real data 1R	MBPO 1R:1M	Ratio of Real:CoDA [:MBPO] data (ours) 1R:1C	1R:3C	1R:5C	1R:3C:1M
25	13.2 ± 0.7	18.5 ± 1.5	43.8 ± 2.8	40.9 ± 2.5	38.4 ± 4.9	$\mathbf{46.8} \pm 3.1$
50	22.8 ± 3.0	36.6 ± 4.3	66.6 ± 3.8	64.4 ± 3.1	62.5 ± 3.5	$\mathbf{70.4} \pm 3.8$
75	43.2 ± 4.9	46.0 ± 4.7	73.4 ± 2.8	$\mathbf{76.7} \pm 2.6$	75.0 ± 3.4	74.6 ± 3.2
100	63.0 ± 3.1	66.4 ± 4.9	77.8 ± 2.0	$\mathbf{82.7} \pm 1.5$	76.6 ± 3.0	73.7 ± 2.9
150	77.4 ± 1.2	72.6 ± 5.6	82.2 ± 1.8	$\mathbf{85.8} \pm 1.4$	84.2 ± 1.0	79.7 ± 3.6
250	78.2 ± 2.7	77.9 ± 2.4	85.0 ± 2.9	$\mathbf{87.8} \pm 1.8$	87.0 ± 1.0	78.3 ± 4.9

Results: Goal-Conditioned RL

- CoDA obtains **state-of-the-art results** in `FetchPush` and **doubles sample efficiency of HER** in challenging `Slide2`
- Here we use the same heuristic mask based on physical separation (objects independent if >10cm apart)





References

[1] Andrychowicz et al. Hindsight Experience Replay. NeurIPS 2017.
[2] Lachapelle et al. Gradient-based Neural DAG Learning. ICLR 2020.
[3] Lee et al. Set Transformer [...]. ICML 2019.

Links

15m talk: https://bit.ly/coda_ool
Code: https://bit.ly/coda_github
arXiv: https://arxiv.org/abs/2007.02863