

1 **[R2, R3] Amount of augmented data and sample efficiency** R2 questioned the effective sample size of CoDA, and
2 R3 asked why more CoDA samples don't always increase performance. In response, we emphasize the distinction
3 between (1) the number of raw data samples and (2) the usefulness of those samples, and discuss some of the difficulties
4 associated with making precise statements about the latter. The exponential increase in *raw data samples* that results
5 from permuting different subspaces (our Remark 3.1) is plain: given m sets (subspaces), each with n unique elements
6 (subsamples), their Cartesian product has cardinality n^m . E.g. Permuting triplets $\{A1♣, B2♦, C3♠, D4♥\}$ forms
7 $4^3 = 64$ unique samples: $(\{A1♣, A1♦, A1♠, A1♥, A2♣, \dots\})$. This is all we meant by Remark 3.1: that within
8 a local subspace, we might produce an exponential number of raw data samples (if there are duplicate subsamples,
9 the permuted samples will contain duplicates). This does not necessarily mean that sample efficiency increases by a
10 commensurate amount (we observe in the Batch RL experiment that too much CoDA data can hurt). Several reasons
11 prevent us from making a stronger statement than Remark 3.1. First is distributional shift, too much of which can cause
12 off-policy algorithms to diverge (the same effect is observed with HER [1], Fig 6). Unfortunately we are not aware of
13 any good theoretical approach to quantifying the effect of distributional shift in RL (cf. [17] and arxiv:2003.07305).
14 Second is the fact that CoDA is model-agnostic: it occurs at the data level, and can be applied to models with different
15 inductive biases. We agree with R2: if the model already has the local SCM built-in as an inductive bias, and learns by
16 estimating each of the univariate conditionals as posited, CoDA will not help. Such a model is not generally available
17 in the RL context, and in our paper we feed CoDA data into standard agents that use fully connected networks with
18 no built-in notion of independence / subspaces. In this case, CoDA data prevents the model from learning spurious
19 dependencies across independent subspaces, and helps with sample efficiency, as we demonstrate empirically. We note
20 that to the extent that non-causal dependencies (between features in the state at time t) are useful for the prediction task
21 (policy or value), especially in the low-data regime, CoDA data might help the model learn these non-causal relations;
22 thus, we believe CoDA will be helpful even when using a factored model such as a GNN (perhaps fruitful future work).

23 **[R2] "Minimality" and Project Scope** We agree our "intuitive" explanation of minimality might mislead in the way
24 described, and will append something to the effect of "while holding all other nodes constant" to better reflect the formal
25 definition at L89-91, which we think is precise/precludes R2's counterexample. Note that causal graphs in our paper
26 are shallow (bipartite), so the $V_1 \rightarrow V_2 \rightarrow V_3$ example would not arise in our context. Overall, rather than unifying
27 RL/Causal literatures, we show a broad application of causal techniques yielding empirical sample efficiency in RL.

28 **[R1, R3] Causal discovery method** R3 notes that our choice of Transformer is a bit ad hoc (we agree) and that there is
29 no guarantee of finding the causal graph (we agree). We restricted our discussion of causal discovery to the appendix
30 because it is orthogonal to our main contributions [LCMs, Alg 1, empirical (see L44-55)] and is, as R3 correctly points
31 out, somewhat ad hoc in that alternatives are not as well explored as we would have liked. We note, however, that the
32 use of a network mask (thresholded absolute Jacobian) for (global) causal discovery is a known, theoretically-grounded
33 technique (see GraN-DAG [39], cited at L217), and that all we are doing is additionally conditioning the mask on the
34 local context (s_t, a_t) . This might done with any context-dependent model (e.g., the mixture of experts model we tried
35 in the Appdx), but we chose a Transformer because it was the first model we tried that worked well. As we note at
36 L290, this component can likely be improved in future work (as evidenced by the gap between learned CoDA and
37 ground truth CoDA in Fig. 4). We will add the figure R1 requested to the paper for the Batch RL experiment, but we
38 unfortunately could not complete it in time for the rebuttal. We note that a random mask fails (see Fig 11 in Appdx), and
39 have observed anecdotally that using a poorly trained Transformer model for CoDA results in poor agent performance.

40 **[R1, R3] When do local factorizations (subspaces at bottom of page 3) arise?** We see that our examples at the
41 end of Remark 3.4 are poorly worded and will revise. We expect that local factorizations will arise when things
42 are physically separate, as we observe in our various examples (two-armed robot, pool balls, sprites, pong paddles,
43 gripper and block), which we posit is very common (detection of physical separation should also transfer well across
44 different objects, as demonstrated in our Fetch experiments). The "mental ignorance" comment at the end of Remark
45 3.4 references a common scenario in multi-agent settings, where agent A's actions are known to be independent of agent
46 B's actual thoughts (but not agent A's belief about agent B's thoughts), and other true facts that agent A is ignorant of.

47 **[R3] Model-based baseline.** MBPO baseline was omitted for Spriteworld only because the experiment used a different
48 codebase (see submitted code), and we only implemented the stronger MBPO model in the Batch RL code. Anecdotally,
49 we tested a couple models (e.g. LSTM) in Spriteworld, but they performed poorly; we will add a standard (non-
50 Transformer) model in the revision. We did not try the delta state trick; this is a helpful suggestion (thanks!) that we
51 believe will help the learned CoDA mask as well. Note that CoDA + MBPO were complementary in the Batch RL case.

52 **[R3] Precision (subgraphs/mechanisms, model-free, reproducibility)** Thanks for these comments. We will do our
53 best to clarify in the revision. For now, we note that (1) mechanism and subgraph are more or less interchangeable (see
54 definition L148); neither is meant to include the structural equation, (2) we think our usage of model-free (defined at
55 L66) is standard in RL, insofar as we reuse subsamples from the environment rather than rolling out a forward dynamics
56 model, and (3) code was provided for reproducibility (but that is no excuse for missing details, which we will add).