Reproducing and Extending Counterfactual Data Augmentation: A Study on Causal Identifiability and Stability in Reinforcement Learning

Sajan Kumar, Shilpa Noushad, and Pratyush Uppuluri

Purdue University, West Lafayette, USA

Abstract. We present a reproducibility-focused reimplementation and extension of CTRL (Counterfactual-based Reinforcement Learning), a causal reinforcement learning method based on counterfactual data augmentation. Beyond reproducing CartPole-SD (CartPole with the Simple Dataset protocol of [3]), we run a controlled validation matrix over counterfactual fraction, noise level, dataset size, and generator quality, then test transfer to LunarLander, MuJoCo, and D4RL-style offline settings. The empirical pattern is consistent across runs: counterfactual augmentation is conditionally useful, not uniformly superior. In CartPole, it can improve clean returns, especially with larger datasets and stronger generators, but noisy-evaluation gains are modest. In cross-domain settings, outcomes are mixed and currently budget-limited. Our claim is therefore scoped: counterfactual augmentation can help offline RL in specific regimes, but reliability depends on data regime, generator fidelity, and evaluation protocol. By introducing a comparative analysis against a non-causal Base-S world model, we identify a critical 'coverage-versus-bias' tradeoff where excessive augmentation can amplify transition inaccuracies, a failure mode particularly evident in balance-heavy tasks. We further demonstrate that Bellman-score selection is insufficient to overcome these biases in high-variance regimes. Finally, we fill a significant gap in the community by providing a verified, ground-up open-source implementation of the CTRL architecture to facilitate further research in causal RL.

Keywords: Causal Reinforcement Learning, Counterfactual Data Augmentation, Offline Reinforcement Learning, Structural Causal Models, Robustness

1 Introduction

Reinforcement Learning (RL) has shown remarkable success in simulated settings, but its real-world adoption in domains such as healthcare, robotics, and industrial control remains limited by costly data collection and unsafe exploration. Improving **sample efficiency and exploration** has thus become a central goal in practical RL, motivating approaches that learn effectively from small, fixed datasets while preserving theoretical convergence guarantees.

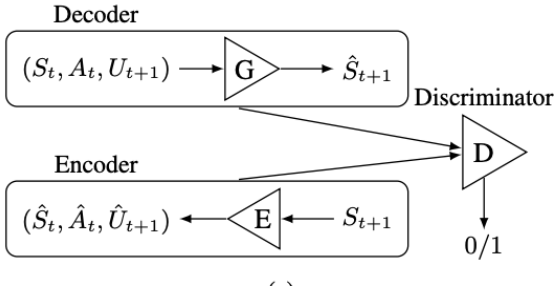

Fig. 1. Generator G, Encoder E, and Discriminator D in CTRL [3].

The CTRL framework (*Sample-Efficient Reinforcement Learning via Counterfactual Based Data Augmentation*, NeurIPS 2020) introduces a **Structural Causal Model (SCM)** to represent environment dynamics as

$$S_{t+1} = f(S_t, A_t, U_{t+1}), \tag{1}$$

where U_{t+1} denotes exogenous noise. By inferring this latent variable for each observed transition and reusing it to compute **counterfactual next states** $S'_{t+1} = f(S_t, a', U_{t+1})$ for alternate actions, CTRL generates additional, causally consistent experiences without new interactions—enabling **sample-efficient "imaginative exploration"** that broadens data coverage by asking *"what if the agent had taken action a' instead of a?"* The paper shows that if f is monotone in U, counterfactual outcomes are identifiable (Theorem 1 in [3]) and that Q-learning trained on this augmented dataset converges to the optimal value function Q^* (Theorem 2 in [3]). Together, these results make CTRL particularly compelling—uniting causal validity and reinforcement learning optimality within a single, data-efficient framework.

To learn this causal mechanism, CTRL employs a **Bidirectional Conditional GAN (BiCoGAN)[2]** that jointly trains an encoder, generator, and discriminator as in Figure 1, ensuring consistency between causal and inverse mappings, enabling inference of latent noise and reconstruction of next states. With its theoretical grounding and simplicity, CTRL offers a practical, scalable framework to evaluate sample efficiency and exploration on benchmark tasks like *Cartpole* and *LunarLander* [1].

In this work, we provide a ground-up reproduction and empirical extension of the `CTRL` framework. Due to the absence of official reference code, we implement the Bidirectional Conditional GAN (BiCoGAN) architecture from scratch to infer latent exogenous noise and generate counterfactual transitions. Beyond reproduction on *CartPole-SD*, we extend analysis to *LunarLander-v3*, MuJoCo control tasks, and D4RL-style offline settings to test where gains transfer and where they fail.

Our contributions are as follows:

- **Ground-Up Reproduction:** A from-scratch implementation of the `CTRL` pipeline (SCM data generation, BiCoGAN training, and offline D3QN+CQL), with run artifacts. D3QN+CQL here refers to Dueling Double Deep Q-Network + Conservative Q-Learning.
- **Controlled Ablation Matrix:** Targeted CartPole studies over Counterfactual(CF) mixing ratio, evaluation noise, dataset size, and generator quality to isolate when CF helps and when it harms.
- **Baseline Differentiation:** Implementation of a non-causal `Base-S` world model to isolate the specific advantages of causal identifiability over probabilistic augmentation.
- **Mechanistic Analysis:** Identification of a **coverage-versus-bias trade-off**, documenting how excessive augmentation amplifies model error in sensitive regions, leading to non-monotonic performance.
- **Cross-Domain Diagnostics:** Extension to LunarLander, MuJoCo, and D4RL-style settings to test external validity and to identify task regimes where CF(Counterfactual) transfer is unstable.

Our results support a scoped claim: causal counterfactual augmentation is conditionally beneficial, but outcomes are sensitive to data regime, generator quality, and evaluation protocol.

For improved accessibility, a comprehensive list of abbreviations and key terms used throughout this work is provided in Table 4 in the Appendix.

Related Work:

This paper is positioned as a *reproduction-and-analysis study*, not a new algorithmic proposal. We build on the SCM formulation of CTRL [3] and the underlying causal framework [4]. We also relate our findings to broader data-augmentation baselines such as CoDA [5] and RAD [7], and to model-based world-model perspectives [6].

To situate this work in the post-2020 literature, we explicitly acknowledge direct successors to counterfactual augmentation (MoCoDA, ACAMDA, CA-IAC, RoCoDA) [12,13,14,15], survey/framework references [16,17,18], and foundational causal RL/representation works [22,19,20,21].

2 Methodology

We closely follow the theoretical framework presented in [3] and reproduce their experimental setup as faithfully as possible. Since the authors do not provide an official GitHub repository, and several implementation details in the paper remain ambiguous, we take particular care to avoid unintended deviations. To mitigate bias, we conducted multiple independent experiments to validate that our choices remain aligned with the underlying theory and with the intended

behavior described by the authors. Nevertheless, we acknowledge that the lack of publicly available code and the inherent under specification in parts of the original description may lead to unavoidable discrepancies.

In our study, we implement Algorithm 1 from [3] and replicate the core CTRL components: environment dynamics, SCM assumptions, and counterfactual augmentation. We first reproduce the CartPole-SD setting and then extend evaluation beyond the original scope to `LunarLander-v3`, MuJoCo, and D4RL-style offline checks to test how behavior changes under more complex domains and broader validation protocols.

2.1 Which CTRL Variant We Implement and Why

Lu et al. [3] present two complementary procedures. **Part 1** (Algorithm 1 in [3], our Algorithm 1) learns a *general* policy by first estimating the SCM with BiCoGAN, generating counterfactual data for all alternative actions, and then performing offline Q-learning on the augmented dataset $\tilde{\mathcal{D}}$. **Part 2** learns a *policy-specific* generalization in which counterfactual generation is conditioned on the policy under refinement, requiring an inner loop that interleaves CF generation with policy updates.

We implement **Part 1 (general-policy variant)** for three reasons. (i) *Offline compatibility*: Part 1 separates SCM estimation from policy learning, which matches our offline D3QN+CQL pipeline; Part 2 entangles CF generation with the current policy iterate and is closer in spirit to a model-based online procedure. (ii) *Reproducibility*: with no official reference code, the cleaner two-stage decomposition of Part 1 reduces implementation ambiguity and makes the SCM and the RL components independently auditable. (iii) *Diagnostic value*: Part 1 lets us isolate the effect of the augmented dataset $\tilde{\mathcal{D}}$ itself, which is exactly the lever our coverage-versus-bias analysis (Sec. 3.5) targets. A policy-conditional Part 2 study is left for future work.

2.2 Identifiability via Monotonicity in U: Practical Meaning

Theorem 1 in [3] requires the structural function $f(s_t, a_t, \cdot)$ to be *strictly monotone in the exogenous noise* U_{t+1}. Operationally this means: for every fixed (s_t, a_t), the map $u \mapsto f(s_t, a_t, u)$ is one-to-one, so each observed next state s_{t+1} pinpoints a unique latent u_{t+1}. This is the condition that makes the encoder E in BiCoGAN well-defined as a noise-inversion operator rather than a generic associative regressor, and it is what licenses reusing the inferred $\hat{u}_t$ for an alternative action a_{cf} in Eq. (1). Without it, two different latents could explain the same s_{t+1}, the inferred $\hat{u}_t$ is no longer a property of the transition, and counterfactual states drift away from causally consistent outcomes.

Does it hold in our environments?

- **CartPole-SD:** the clean dynamics are deterministic and the dominant noise channel is additive Gaussian observation noise $s' = s'_{\text{clean}} + 0.05\,\eta$, which is

componentwise strictly monotone in η. Action-execution noise enters through the CartPole physics and may break strict monotonicity in edge regions, but the operative noise channel during state generation is monotone, and this is consistent with CartPole being the regime where we observe the clearest CF gains.
- **LunarLander-v3:** stochasticity enters through wind, turbulence, and contact-dependent terms that interact with state in nonlinear, non-componentwise ways, so $f(s, a, \cdot)$ is generally not monotone in a single scalar latent. The high-variance, often-negative LunarLander outcomes are consistent with a violated or only partially-satisfied condition.
- **MuJoCo (Ant, HalfCheetah, Hopper, Walker2d):** contact-rich dynamics produce non-smooth, non-monotone responses to perturbations. We expect monotonicity to be more strongly violated on balance-heavy tasks (Hopper, Walker2d), and indeed those are the tasks where CF augmentation degrades performance in Sec. 3.3.

The takeaway is that identifiability is a *sufficient condition*, not a free lunch. Our coverage-versus-bias finding can be read as the empirical signature of a domain where the monotonicity premise is partially or fully violated, so the BiCoGAN encoder learns an associative inverse rather than a causal noise inversion, and aggressive CF mixing then injects model error rather than support.

2.3 Stochastic CartPole-SD Environment and Evaluation Protocols

We follow the SD(Simple Dataset) setting from [3] and explicitly document the environment details used in our implementation. The underlying CartPole dynamics remain unchanged; we use the same 11-action parameterization, additive action noise, and noisy observations. The agent selects among 11 discrete actions $a \in \{0, \ldots, 10\}$, which are mapped to a continuous control value $a_{\text{cont}} = a/10$ and further perturbed by Gaussian execution noise $\tilde{a} = a_{\text{cont}} + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.05^2)$. The resulting applied force is $F = (2\tilde{a} - 1)\, F_{\max}$ with $F_{\max} = 10$. Termination is evaluated on the *clean* next state prior to adding any stochastic noise, i.e., whenever $|x'_{\text{clean}}| > 2.4$ or $|\theta'_{\text{clean}}| > 12°$. After termination is checked, the environment returns a noisy observation $s' = s'_{\text{clean}} + 0.05\,\eta$, with $\eta \sim \mathcal{N}(0, I_4)$, subsequently clipped to the usual CartPole bounds $x' \in [-4.8, 4.8]$ and $\theta' \in [-0.418, 0.418]$. Dataset generation samples random discrete actions, applies noisy actions $\tilde{a}$, and records $(s, a, \tilde{a}, r, s')$ until termination.

Clean vs. noisy evaluation. To avoid ambiguity in the results section:

- **Clean** refers to evaluation on standard `CartPole-v1` (no SD observation noise in environment state transitions).
- **CTRL/noisy** refers to evaluation on our SD environment with built-in observation noise ($\sigma_s = 0.05$) and 11-action mapping; additional action-evaluation noise is set to zero unless stated.

2.4 Offline D3QN + CQL Training with Real and Counterfactual Data

After training BiCoGAN (Algorithm 2), we construct the offline RL dataset by combining real SD transitions with counterfactual (CF) transitions generated via the learned (G, E) pair. For each real transition (s_t, a_t, r_t, s_{t+1}), we first infer the transition-specific latent exogenous variable using the encoder and then *reuse that inferred latent* for alternative actions:

$$\hat{u}_t = E(s_{t+1}), \qquad \hat{s}' = G(s_t, a_{\text{cf}}, \hat{u}_t), \qquad a_{\text{cf}} \neq a_t.$$

Each CF(Counterfactual) transition inherits reward r_t and uses a done flag based on the clean termination thresholds applied to $\hat{s}'$.

We evaluate two regimes:

- **Real-only**: the offline dataset consists solely of the clean SD transitions.
- **Real + CF (k per real)**: each real transition is augmented with k counterfactual transitions generated from BiCoGAN.

All states and next states are standardized using the real dataset mean and standard deviation. We train an offline Dueling Double Deep Q-Network (D3QN) agent as in Algotithm 3 with a Conservative Q-Learning (CQL) penalty, following the update structure of [3]. The value network is trained using Double DQN targets and soft target updates.

Reproducibility. All experiments, result plots, and evaluation logs used in this section can be fully reproduced using our publicly released notebook available at `Counterfactual-RL Experiments`. This includes the specific configurations used for the `CartPole-SD` and `LunarLander-v3` environments.

2.5 Experiments on LunarLander-v3

For our second environment, we use the standard `LunarLander-v3` from Gymnasium, with default physics (gravity, wind, and turbulence parameters), and no modifications to the underlying dynamics.

BiCoGAN SCM for LunarLander. We reuse the same Structural Causal Model (SCM) framework developed for CartPole-SD: a bidirectional conditional GAN that learns forward dynamics $G(s_t, a_t, u_t)$ and an encoder $E(s_{t+1})$ for recovering latent exogenous structure. The architecture, loss functions, and training schedule are identical to the CartPole version; only the state and action dimensions differ.

Base-S World Model. To compare SCM-based counterfactual generation with a simpler probabilistic world model, we implement *Base-S*, a Gaussian next-state-and-reward model following the formulation in the CTRL paper. Base-S conditions on the current state and a one-hot action vector and produces

$$(\mu_{s'}, \mu_r, \log \sigma^2_{s'}, \log \sigma^2_r),$$

the means and log-variances of a diagonal Gaussian distribution over next state and reward. Samples are drawn via reparameterization, and the model is trained using a Gaussian negative log-likelihood loss:

$$\mathcal{L}_{\text{NLL}} = \mathbb{E}\left[\mathcal{N}(s_{t+1}; \mu_{s'}, \sigma^2_{s'}) + \mathcal{N}(r_t; \mu_r, \sigma^2_r)\right].$$

Training Base-S. We train Base-S for 20 epochs using Adam with learning rate 10^{-3}, weight-normalized MLP layers, and minibatch NLL over both next state and reward. This follows the training setup stated in the CTRL paper and provides a baseline generative model for producing counterfactual transitions. The trained Base-S model is then used to generate synthetic rollouts for offline RL, enabling a direct comparison with the BiCoGAN SCM approach.

3 Results

Main framing. We report results in two tiers: (i) controlled CartPole-SD offline ablations to study mechanism-level behavior, and (ii) cross-domain validation on LunarLander, MuJoCo, and D4RL-style offline evaluation for external validity. Throughout this section, *clean* means evaluation on standard `CartPole-v1`; *CTRL/noisy* means evaluation on the SD environment with observation noise and 11-action control. Unless otherwise noted, values are mean returns; exploratory settings with smaller seed counts are explicitly marked in tables.

3.1 CartPole Offline Matrix (Controlled Analysis)

Table 1. CartPole-SD offline evaluation (mean return ± std over seeds; main settings use $n = 30$, exploratory edge fractions use $n = 10$).

Agent	CF frac	Clean	CTRL (noisy)
Rainbow	–	37.20 ± 44.37 ($n = 30$)	N/A
D3QN (real only)	0	33.14 ± 10.23 ($n = 30$)	17.99 ± 1.49 ($n = 30$)
D3QN + CF	0.02	37.77 ± 21.21 ($n = 10$)	17.15 ± 1.47 ($n = 10$)
D3QN + CF	0.05	34.25 ± 10.87 ($n = 30$)	17.82 ± 1.46 ($n = 30$)
D3QN + CF	0.10	35.60 ± 20.73 ($n = 30$)	17.73 ± 1.37 ($n = 30$)
D3QN + CF	0.25	33.87 ± 12.45 ($n = 30$)	18.37 ± 1.60 ($n = 30$)
D3QN + CF	0.50	44.66 ± 23.30 ($n = 10$)	18.47 ± 1.53 ($n = 10$)

On clean CartPole, CF(Counterfactual) augmentation improves over real-only D3QN in several settings (notably $f = 0.10$ and exploratory $f = 0.50$), but gains are non-monotonic in CF fraction and have wide dispersion in some regimes. On CTRL/noisy evaluation, all D3QN variants remain tightly clustered around ~ 18, indicating limited robustness separation.

Key takeaways from the CartPole matrix.

- CF(Counterfactual) augmentation can improve clean-return performance, but the effect is not monotonic in CF fraction.
- Noisy evaluation remains tightly clustered across D3QN variants, indicating limited robustness gains in this setting.
- Rainbow shows high variance in this offline protocol, so the main causal comparison is between D3QN real-only and D3QN+CF variants.
- Exploratory edge fractions ($f = 0.02$, $f = 0.50$) use fewer seeds and should be interpreted as trend indicators, not final estimates.

3.2 Ablations: When Does CF Help?

Dataset-size ablation is shown in Figure 2. We define **small** as 100 episodes (1669 transitions), **medium** as 250 episodes (4149 transitions), and **large** as 500 episodes (8543 transitions). CF(Counterfactual) is regime-dependent: it hurts in small/medium data but helps in large data (clean means: small 35.20 vs real 44.83; medium 31.37 vs real 35.15; large 42.38 vs real 35.07).

Figure 3 summarizes CF-quality and BiCoGAN-quality effects. Raw CF(Counterfactual) gives the highest clean score in this matrix (55.80), while aggressive filtering/subsampling reduces gains. Generator quality is also monotonic in this study (weak 31.01, medium 34.52, strong 41.68 clean mean), confirming that downstream RL performance is sensitive to SCM quality.

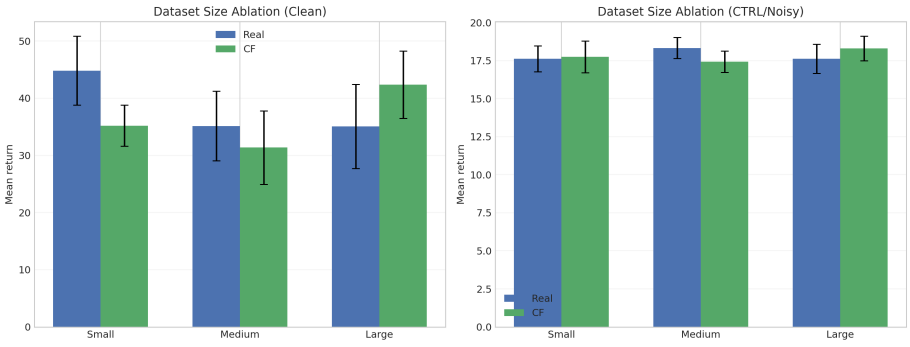

Fig. 2. Dataset-size ablation: CF vs real-only under clean and CTRL/noisy evaluation. CF benefit appears in the large-data regime only.

3.3 Cross-Domain Validation

Cross-domain results are mixed: CF(Counterfactual) improves LunarLander mean and 2/4 MuJoCo tasks (Ant, HalfCheetah), but degrades 2/4 MuJoCo tasks

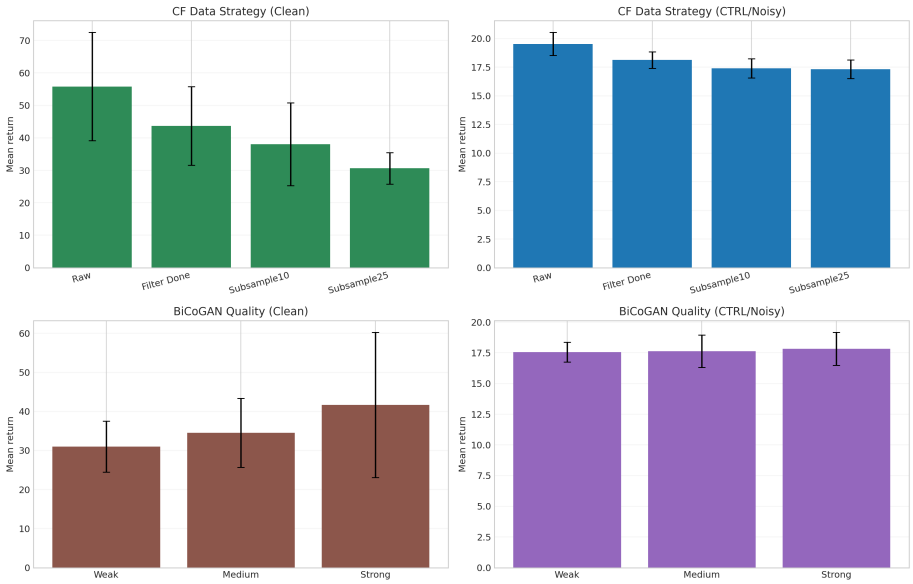

Fig. 3. CF strategy and BiCoGAN-quality ablations. Clean performance depends strongly on augmentation/generator quality; noisy returns remain closely clustered.

(Hopper, Walker2d). D4RL numbers are reported as budget-limited baseline checks (one CQL configuration per dataset at 100k steps, raw returns) and are not yet sufficient for a strong causal-augmentation claim.

Task-level interpretation and statistical caveat. The MuJoCo split is consistent with task sensitivity: Ant and HalfCheetah are more tolerant to local model error, while Hopper and Walker2d are balance-heavy and can fail under small transition inaccuracies. We therefore treat these cross-domain outcomes as diagnostic rather than definitive, because current budgets use limited seeds on several tasks ($n = 2$ or $n = 3$), and D4RL reporting is currently a low-budget sanity pass rather than a normalized-score benchmark.

3.4 Focused Diagnostic: Bellman-Selected CF on LunarLander-v3

To isolate one failure mode, we evaluate a Bellman-score selector that keeps only the highest-scoring counterfactual per transition. This diagnostic uses a different protocol from the cross-domain matrix and is reported separately.

Even with Bellman-score selection (Table 3), counterfactual augmentation did not improve *LunarLander-v3* in our runs. Both Base-S and BiCoGAN variants remained high-variance and underperformed the Real-only baseline. Lunar lander experiments can be observed and run with code at `Counterfactual-RL LunarLander Experiments`.

Table 2. Cross-domain summary with task-level outcomes and run budgets.

Domain	Task/Setting	Budget	Outcome
LunarLander offline	real vs CF	250 dataset eps, 400 DQN epochs, $n = 8$	-158.25 vs -144.31
MuJoCo SAC	Ant-v4 (real vs aug)	200k steps, 20 eval eps, $n = 2$	857.32 → 1330.07
MuJoCo SAC	HalfCheetah-v4 (real vs aug)	200k steps, 20 eval eps, $n = 3$	3974.77 → 5246.36
MuJoCo SAC	Hopper-v4 (real vs aug)	200k steps, 20 eval eps, $n = 3$	1785.47 → 763.75
MuJoCo SAC	Walker2d-v4 (real vs aug)	200k steps, 20 eval eps, $n = 2$	2692.83 → 1945.52
D4RL CQL (baseline)	halfcheetah-medium-v2	100k steps, 10 eval eps, $n = 2$	-336.17
D4RL CQL (baseline)	hopper-medium-v2	100k steps, 10 eval eps, $n = 2$	17.80
D4RL CQL (baseline)	walker2d-medium-v2	100k steps, 10 eval eps, $n = 3$	10.86

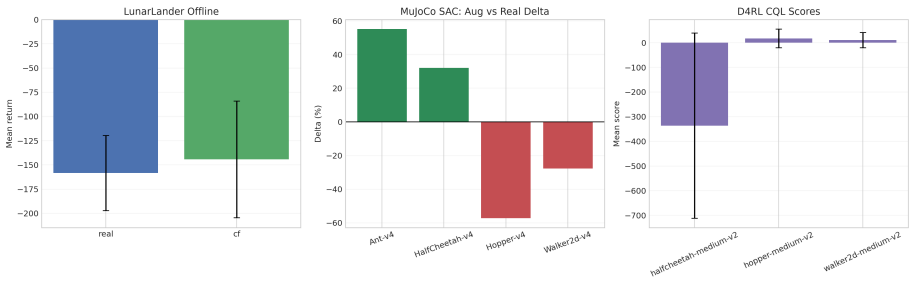

Fig. 4. External-validity summary across LunarLander, MuJoCo, and D4RL.

3.5 Interpretation and Next Methodological Steps

Interpretation (hypothesis, not proof). Our non-monotonic CartPole gains are consistent with a coverage-versus-bias tradeoff: limited CF(Counterfactual) can improve support, while excessive CF can amplify model error in synthetic regions. This interpretation is consistent with the high variance observed in LunarLander and with a possible Bellman-selection feedback loop where biased high-value CF samples are repeatedly over-selected [3,8].

Proposed methodological changes (not yet evaluated). We propose three concrete upgrades for the next revision cycle. First, replace unconstrained GAN generators with monotonic or identifiable SCM parameterizations [24,25]. Second, replace fixed CF(Counterfactual) mixing with uncertainty-aware acceptance and discrepancy-aware augmentation [13,26,27]. Third, benchmark stronger offline RL baselines (for example, IQL and support-constrained variants) and report scaling behavior [23,28]. We also plan to add a non-causal augmentation baseline so causal benefit can be separated from generic data-augmentation effects.

Causal invariance diagnostic (planned extension). To separate causal mechanism learning from associative fitting, a direct next step is an invariance diagnostic under controlled environment shifts (for example, changing gravity or wind in *LunarLander-v3*) [10,11]. The intended plot compares one-step prediction error versus shift magnitude for BiCoGAN-SCM and Base-S.

Table 3. Final evaluation returns (20 episodes) using Bellman-selected counterfactuals.

Agent	CF Source	Mean Return ± Std
Real-only	None	-145.95 ± 55.89
Hybrid CTRL (best-CF)	Base-S	-326.99 ± 116.05
Hybrid CTRL (best-CF)	BiCoGAN	-262.48 ± 86.85

4 Conclusion and Future Work

This study reimplements CTRL and evaluates it under a substantially broader matrix than the original reproduction baseline. Our evidence supports a *conditional* conclusion: counterfactual augmentation can improve offline RL performance, but gains depend strongly on regime and are not universal.

In controlled CartPole analysis, CF(Counterfactual) data improves clean-return performance in several settings and shows stronger effect when dataset size and generator quality are higher. However, noisy/CTRL robustness gains are modest, and cross-domain transfer is mixed (MuJoCo has both wins and losses; D4RL remains underpowered at current budget and seed count).

Practical takeaway. Counterfactual augmentation should be treated as a high-variance lever that requires careful control of data quality, generator quality, and mixing ratio. It is promising, but not yet a drop-in replacement for standard offline RL pipelines.

Future work. To strengthen external validity, the next phase should prioritize compute-matched seed expansion and broader MuJoCo/D4RL coverage with normalized reporting. Methodological upgrades (monotonic/identifiable SCMs, uncertainty-aware CF selection, and stronger offline RL baselines) are detailed in the preceding diagnostic subsection.

Implications for Causal RL Research. Our results carry three takeaways that extend beyond the CTRL reproduction. **First, identifiability is necessary but not sufficient for downstream RL gain.** Theorem 1 of [3] guarantees that counterfactual outcomes are recoverable when f is monotone in U, yet in our LunarLander and MuJoCo experiments the condition is plausibly violated and CF augmentation either fails to help or actively hurts. Causal RL methods that inherit the same identifiability premise (MoCoDA, ACAMDA, CAIAC, RoCoDA [12,13,14,15]) should expect similar regime-dependence and benefit from explicitly reporting whether their causal assumptions hold in the target environments, beyond citation. **Second, generator fidelity, not causal structure per se, is the dominant lever in our matrix.** Generator-quality ablations move clean returns from 31.01 (weak) to 41.68 (strong), a larger swing than any CF mixing ratio produces. This suggests that the field's near-term progress is more likely to come from constrained, identifiable SCM parameterizations (*e.g.* Chen and Du [25]) than from new mixing heuristics on top of

unconstrained GAN generators. **Third, the coverage-versus-bias tradeoff is task-aware.** Balance-heavy control tasks (Hopper, Walker2d) tolerate transition error far less than locomotion tasks (Ant, HalfCheetah). A practical causal-RL deployment story therefore needs an upstream invariance or sensitivity check before CF augmentation is applied, of the kind we sketch in Sec. 3.5 [10,11].

Among recent successors, the work most closely aligned with our research direction is **ACAMDA** [13], which makes counterfactual generation *adversarial* and discrepancy-aware, directly attacking the bias side of the coverage-versus-bias tradeoff. Among the most recent identifiability-focused efforts, **Chen and Du** [25] on exogenous isomorphism is the closest theoretical match to our monotone-U critique, since it formalizes when the exogenous structure can be recovered up to isomorphism. We view both as the natural next baselines once seed budgets and offline-RL coverage are expanded.

Open-Source Implementation. To fill the existing gap in open-source causal RL frameworks, a verified, ground-up implementation of the `CTRL` architecture is provided. See our public notebook and code available at `Causal-RL-Study`.

References

1. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym. arXiv preprint arXiv:1606.01540 (2016)
2. Jaiswal, A., AbdAlmageed, W., Wu, Y., Natarajan, P.: Bidirectional Conditional Generative Adversarial Networks. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018. LNCS, vol. 11363, pp. 216–232. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20893-6_14
3. Lu, C., Huang, B., Wang, K., Hernandez-Lobato, J.M., Zhang, K., Scholkopf, B.: Sample-Efficient Reinforcement Learning via Counterfactual-Based Data Augmentation. In: Advances in Neural Information Processing Systems, vol. 33, pp. 18750–18761 (2020)
4. Pearl, J.: Causality. Cambridge University Press, Cambridge (2009)
5. Pitis, S., Creager, E., Garg, A.: Counterfactual Data Augmentation using Locally Factored Dynamics. In: Advances in Neural Information Processing Systems, vol. 33, pp. 3976–3987 (2020)
6. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering Diverse Domains through World Models. arXiv preprint arXiv:2301.04104 (2023)
7. Laskin, M., Srinivas, A., Abbeel, P.: Reinforcement Learning with Augmented Data. In: Advances in Neural Information Processing Systems, vol. 33, pp. 19884–19895 (2020)
8. Kumar, A., Zhou, A., Tucker, G., Levine, S.: Conservative Q-Learning for Offline Reinforcement Learning. In: Advances in Neural Information Processing Systems, vol. 33 (2020)
9. Vygotsky, L.S.: Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)
10. Peters, J., Buhlmann, P., Meinshausen, N.: Causal Inference by Using Invariant Prediction: Identification and Confidence Intervals. Journal of the Royal Statistical Society: Series B **78**(5), 947–1012 (2016). https://doi.org/10.1111/rssb.12167
11. Scholkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., Mooij, J.M.: On Causal and Anticausal Learning. In: Proceedings of the 29th International Conference on Machine Learning (2012)
12. Pitis, S., Creager, E., Garg, A.: Model-Based Counterfactual Data Augmentation for Offline Reinforcement Learning. In: Advances in Neural Information Processing Systems, vol. 35 (2022)
13. Sun, Y., Wu, T., Yu, X., Lu, C., Zhang, K.: Adversarial Causal Model-Based Data Augmentation for Offline Reinforcement Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38(14), pp. 15653–15661 (2024). https://doi.org/10.1609/AAAI.V38I14.29442
14. Armengol Urpi, A., Hernandez-Lobato, J.M., Lu, C.: Counterfactual Augmentation via Action Influence in Offline Reinforcement Learning. In: Proceedings of the 41st International Conference on Machine Learning. PMLR, vol. 235, pp. 1709–1729 (2024)
15. Ameperosa, N., Curi, S., Janner, M., Garg, A.: RoCoDA: A Unified Framework for Invariance, Equivariance, and Causality in Robot Learning. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 17111–17118 (2025). https://doi.org/10.1109/ICRA55743.2025.11128694
16. Deng, Z., Zhang, D., Li, Y., Bareinboim, E.: Causal Reinforcement Learning: A Survey. Transactions on Machine Learning Research (2023)

17. Zeng, G., Li, X., Wang, Q., Zhang, K.: A Survey on Causal Reinforcement Learning. IEEE Transactions on Neural Networks and Learning Systems (2025). https://doi.org/10.1109/TNNLS.2024.3403001
18. Bareinboim, E., Zhang, J., Lee, S.: An Introduction to Causal Reinforcement Learning. Tech. rep., Causal AI Lab (2025). https://causalai.net/r65.pdf
19. Buesing, L., et al.: Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search. In: International Conference on Learning Representations (2019)
20. Huang, B., Lu, C., Wang, K., Zhang, K.: Learning Action-Sufficient State Representation for Control from Pixels. In: Proceedings of the 39th International Conference on Machine Learning (2022)
21. Huang, B., Lu, C., Zhang, K.: AdaRL: What, Where, and How to Adapt in Transfer Reinforcement Learning. In: International Conference on Learning Representations (2022)
22. Scholkopf, B., et al.: Toward Causal Representation Learning. Proceedings of the IEEE **109**(5), 612–634 (2021). https://doi.org/10.1109/JPROC.2021.3058954
23. Kostrikov, I., Nair, A., Levine, S.: Offline Reinforcement Learning with Implicit Q-Learning. In: International Conference on Learning Representations (2022)
24. Nasr-Esfahany, M., Madani, K., Shah, B.R.: Identifiability and Stability in Causal Structure Learning from Time Series Data. Entropy **25**(5), 761 (2023). https://doi.org/10.3390/e25050761
25. Chen, D., Du, Y.: Exogenous Isomorphism in Identifiable Causal Reinforcement Learning. In: Proceedings of the 42nd International Conference on Machine Learning (2025)
26. Lu, B., Fan, W., Ma, C., Cai, X., Wang, Y., Liu, Y.: OASIS: Offline Actor-Critic with Synthesis. arXiv preprint arXiv:2410.03679 (2024)
27. Li, Z., et al.: BECAUSE: Bilinear Causal Representation for Generalization under Hidden Confounding. In: Advances in Neural Information Processing Systems, vol. 37 (2024)
28. Zhang, S., So, O., Ahmad, H.M.S., Yu, E.Y., Cleaveland, M., Black, M., Fan, C.: ReFORM: Reflected Flows for On-support Offline RL via Noise Manipulation. In: International Conference on Learning Representations (2026)

A Appendix

Table 4. Key Abbreviations, Full Forms, and Descriptions

Abbreviation	Description and Context
CTRL	**Counterfactual-based Reinforcement Learning**: Derived from the original framework title, "Sample-Efficient Reinforcement Learning via Counterfactual-Based Data Augmentation".
CF	**Counterfactual**: A synthetic "what-if" transition generated for an alternative action by reusing inferred latent exogenous noise, used to broaden data coverage without new interactions.
CartPole-SD	**CartPole with a Simple Dataset**: The "SD" specifically refers to the "Simple Dataset" environment setting used for benchmarking in the original work.
MuJoCo	**Multi-Joint dynamics with Contact**: A physics engine used to simulate the continuous control tasks evaluated in this study.
D4RL-style	**Datasets for Deep Data-Driven Reinforcement Learning**: Refers to the standardized format and protocol for offline RL benchmarks used for cross-domain validation.
Base-S	**Baseline State-only world model**: A non-causal Gaussian model that predicts next-states and rewards, used here to isolate the specific benefits of causal identifiability.
SCM	**Structural Causal Model**: A mathematical framework used to represent environment dynamics as functions of states, actions, and exogenous noise [4, 13, 14].
BiCoGAN	**Bidirectional Conditional Generative Adversarial Network**: An architecture used to jointly train an encoder and generator to infer latent noise and reconstruct next states.
D3QN+CQL	**Dueling Double Deep Q-Network + Conservative Q-Learning**: The offline RL algorithm combination used to train the agent on the augmented dataset.
CF mixing ratio	**Counterfactual mixing ratio**: The fraction of synthetic counterfactual transitions added to the real dataset during training.
MuJoCo SAC	**Soft Actor-Critic on MuJoCo**: Refers to the use of the Soft Actor-Critic algorithm for evaluating augmentation on continuous MuJoCo control tasks.

Algorithm 1 Policy Learning via Counterfactual-Based Data Augmentation — Part 1 ([3])

Input: Observed triplets (S_t, A_t, S_{t+1}) from the offline dataset, for $t = 1, \ldots, T$.
Estimation of a general policy:
    2.1. Estimate the SCM in Eq. 1 using BiCoGAN.
    2.2. Generate counterfactual data for alternative actions according to the estimated SCM; denote the counterfactually augmented dataset by $\tilde{\mathcal{D}}$.
    2.3. Perform D3QN learning on $\tilde{\mathcal{D}}$ to obtain the learned policy π.

Note: This pseudocode corresponds to **Part 1** of the CTRL framework in [3], adapted to our implementation pipeline.

Algorithm 2 BiCoGAN Training Pipeline (Our Implementation)

Require: Dataset $\mathcal{D} = \{(s_t, a_t, \tilde{a}_t, r_t, s_{t+1})\}$, hyperparameters $(\alpha, \rho, \phi, \lambda_{\text{fwd}})$.
Ensure: Trained (G, E, D) models.
Stage 1: Pretrain Generator G
for epoch $= 1$ to N_{pre} **do**
  for minibatch $(s_t, a_t, \tilde{a}_t, s_{t+1})$ from $\mathcal{D}$ **do**
    Predict $\hat{s}_{t+1} = G(s_t, a_t, u = 0)$.
    Update G via $\text{MSE}(\hat{s}_{t+1}, s_{t+1})$.
  end for
end for
Stage 2: Adversarial BiCoGAN Training
for epoch $= 1$ to N_{GAN} **do**
  for minibatch $(s_t, a_t, \tilde{a}_t, s_{t+1})$ **do**
    Sample latent $u \sim \mathcal{N}(0, I)$.
    Generate fake $\hat{s}_{t+1} = G(s_t, a_t, u)$.
    Discriminator update:

$$\mathcal{L}_D = \text{BCE}(D(s_t, a_t, s_{t+1}), 1) + \text{BCE}(D(s_t, a_t, \hat{s}_{t+1}), 0)$$

    Gradient step on D.
    Encoder–Generator update:
    $(\hat{s}_t, \hat{a}_t, \hat{u}_t) = E(s_{t+1})$.
    $\hat{s}'_{t+1} = G(s_t, a_t, \hat{u}_t)$.
    Compute

$$\mathcal{L}_{GE} = \mathcal{L}_{\text{adv}} + \gamma_t \mathcal{L}_{\text{EFL}} + \lambda_{\text{fwd}} \mathcal{L}_{\text{fwd}}.$$

    Gradient step jointly on G and E.
  end for
end for

Algorithm 3 Offline D3QN + CQL Training (Real or Real+CF Data)

Require: Normalized dataset (S, A, R, S', D), hyperparameters γ, α_{CQL}, τ.
Ensure: Trained Q-network.

Initialize Q-network Q_θ and target network Q_{θ^-}.
for epoch $= 1$ to N_{epochs} **do**
  Shuffle dataset.
  for each minibatch (s, a, r, s', d) **do**
    Compute $Q_\theta(s, a)$ and $Q_\theta(s, \cdot)$.
    Compute Double DQN target:

$$y = r + \gamma Q_{\theta^-}\left(s', \arg\max_{a'} Q_\theta(s', a')\right)(1 - d).$$

    TD loss:

$$\mathcal{L}_{\text{TD}} = \text{Huber}(Q_\theta(s, a), y).$$

    CQL penalty:

$$\mathcal{L}_{\text{CQL}} = \alpha_{\text{CQL}}\left(\log \sum_{a'} e^{Q_\theta(s,a')} - Q_\theta(s, a)\right).$$

    Total loss:

$$\mathcal{L} = \mathcal{L}_{\text{TD}} + \mathcal{L}_{\text{CQL}}.$$

    Gradient update on Q_θ.
    Soft target update:

$$\theta^- \leftarrow \tau\,\theta + (1 - \tau)\theta^-.$$

  end for
  if epoch mod eval_every $= 0$ **then**
    Evaluate policy for 50 rollouts in the SD/CTRL evaluation environment.
  end if
end for