Redistribution-based Cost Inference Improves Sparse Safe Offline RL

Ebenezer Gelo, Geraud Nangue Tasse, Steven James, Benjamin Rosman
University of the Witwatersrand
Johannesburg, South Africa
ebenezer.gelo1@students.wits.ac.za

***Abstract*—Safe offline RL typically assumes access to dense per-step cost annotations, but in practice supervisors provide only trajectory-level stop-feedback: a binary signal at the first unsafe transition, with no per-step attribution. We frame this as a temporal credit assignment problem and propose the Redistribution-based Cost Inference (RCI) framework, which converts sparse stop-feedback into dense per-step costs via return decomposition, then trains a constrained offline policy on the augmented dataset. We show that return-equivalent redistribution preserves the feasible policy set and the optimal Lagrangian in a CMDP, establishing that the transformation is lossless in theory while yielding better-conditioned cost critic learning in practice. Experiments on highway driving and robotic manipulation demonstrate substantially lower violation rates than sparse and classifier-based baselines, with robustness to heterogeneous dataset compositions and label noise.**

I. Introduction

Safe reinforcement learning (RL) is central to deploying autonomous agents in high-stakes domains such as autonomous driving, clinical decision support, and robotic manipulation [1], [2]. The predominant formalism, the Constrained Markov Decision Process (CMDP) [3], encodes safety as a constraint on expected cumulative cost, but standard CMDP methods presuppose access to dense, per-step cost annotations—a requirement that is prohibitively expensive to satisfy in practice.

The offline RL paradigm exacerbates this gap. Agents trained on fixed historical datasets avoid the risks of online exploration [4]–[6], yet they inherit the safety profile of the behavior policy and face unreliable value and cost estimates under distributional shift [7]. Critically, offline datasets collected without safety instrumentation lack the cost labels required by existing constrained methods, and post-hoc expert annotation at the per-step level is intractable at scale.

A more realistic supervision signal is *trajectory-level stop feedback*: a binary label indicating whether an episode was halted by a safety monitor upon detecting a violation. Such signals arise naturally from human oversight and automated safety systems, which operate by terminating execution rather than scoring individual transitions. However, each unsafe trajectory contributes only a single cost label at its termination point, providing no direct information about which earlier actions precipitated the violation. Recovering per-step costs from this signal is a temporal credit-assignment problem [8]–[10], compounded here by distributional shift and the one-shot nature of offline data collection.

We address this problem with the **Redistribution-based Cost Inference (RCI)** framework, which converts sparse trajectory-level stop labels into dense per-step cost estimates via return decomposition, then trains safe policies using standard constrained offline RL. RCI is explicitly modular: the redistribution stage accepts any return-equivalent decomposition method (instantiated here with RUDDER [8], though GRD [9] applies directly), and the policy learning stage accepts any constrained offline RL algorithm (instantiated with BCQ-Lagrangian [4], though CPQ or CDT substitute without modification). We evaluate on highway driving and simulated robot control tasks across datasets generated by unsafe, random, and mixed behavior policies, demonstrating that RCI substantially reduces constraint violation rates while preserving task performance comparable to an unconstrained baseline.

II. Preliminaries

We model a task as a Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ is the transition dynamics, $r(s,a)$ is the reward function, and $\gamma \in [0,1)$ is the discount factor. A policy $\pi(a|s)$ generates trajectories $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ with cumulative return $R(\tau) = \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$, and the objective is to find a policy that maximizes $\mathbb{E}[R(\tau)]$ [11].

In *offline* RL, the agent learns from a fixed dataset $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$ of trajectories collected by a behavior policy μ, without any further environment interaction. This avoids unsafe online exploration but introduces *distributional shift*: the learned policy π may select actions outside the support of μ, rendering value estimates unreliable—a phenomenon known as extrapolation error. Offline RL methods address this by constraining the learned policy to remain close to the dataset distribution or by penalising out-of-support actions [4], [7], [12].

A. Constrained Markov Decision Processes

A constrained Markov decision process (CMDP) extends the MDP framework by introducing a cost function and a safety budget. Formally, a CMDP is given by $(\mathcal{S}, \mathcal{A}, P, r, c, \gamma, d)$, where $c(s,a)$ is a cost function representing constraint violations and d is an allowable threshold. The agent's objective is to maximize $\mathbb{E}[R(\tau)]$ subject to $\mathbb{E}[C(\tau)] \leq d$, where

$C(\tau) = \sum_{t=0}^{T} \gamma^t c(s_t, a_t)$ [3]. This formulation balances performance and safety; for example, an autonomous vehicle should minimize accidents while still reaching its destination efficiently. A common approach to solving CMDPs is Lagrangian relaxation, where the problem is converted to maximizing $\mathbb{E}[R(\tau) - \lambda C(\tau)]$ with the Lagrange λ adjusted until the cost constraint is satisfied [1], [2].

III. Related Work

a) Offline Safe Reinforcement Learning.: Offline safe RL combines the CMDP formalism with offline training on fixed datasets [13], [14]. Conservative offline RL methods such as BCQ [4] address distributional shift by constraining the learned policy to remain close to the behavior policy. Constrained extensions such as COPO [13] and CPQ [14] incorporate Lagrangian penalties or conservative cost critics into this framework. A key limitation in previous studies is the assumption that per-step cost annotations are available in the dataset, an assumption we relax.

b) Cost Inference from Sparse Feedback.: In practice, safety supervision is often sparse. [10] propose TraCeS, which learns a dense safety scoring model from sparse trajectory labels using a multiplicative decomposition, but requires online labeler queries during training for iterative refinement. Similarly, [15] introduce RLSF, which transforms segment-level feedback into classification targets, but also relies on interactive labeler access during online learning. These settings differ fundamentally from ours: the dataset is fixed, the labeler cannot be queried post-hoc, and all safety supervision must be incorporated offline.

c) Return Decomposition and Credit Assignment.: Return decomposition methods address credit assignment in sparse reward RL by training sequence models to predict cumulative returns and redistributing terminal signals backward through time [8], [9]. These methods guarantee return equivalence: the redistributed per-step signals sum to the original trajectory return. Our work transfers this principle to the cost and constraint domain, establishing that return-equivalent cost redistribution preserves the feasible policy set and the optimal Lagrangian in a CMDP.

IV. Redistribution-based Cost Inference

We consider an offline dataset $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$ of trajectories where each trajectory $\tau = \{(s_0, a_0, r_0), \ldots, (s_T, a_T, r_T)\}$ is labeled by an expert function $\mathcal{F}(\tau)$ returning the index of the first unsafe transition, or the empty set if the trajectory is safe. This produces a sparse cost signal:

$$c^{\text{sparse}}(s_t, a_t) = \begin{cases} 1, & \text{if } t \in \mathcal{F}(\tau); \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The objective is to learn a policy π maximizing expected return $\mathbb{E}_\pi[R(\tau)]$ subject to $\mathbb{E}_\pi[C(\tau)] \leq d$ for a specified budget d, using only $\mathcal{D}$ and sparse cost annotations, within the CMDP framework with soft constraints [3], [16]. The central challenge is that stop-feedback provides trajectory-level information—indicating that something went wrong—but not dense per-step annotations of which actions caused the failure, creating a severe credit assignment problem [8].

We propose Redistribution-based Cost Inference (RCI), a modular framework that decomposes safe offline RL into three stages: (1) trajectory-level stop-feedback collection, (2) return-decomposition-based cost inference, and (3) constrained offline policy learning. The key insight is that credit assignment for costs and policy learning for safety are algorithmically distinct problems that benefit from separate treatment. This modularity allows any return-equivalent decomposition method and any constrained offline RL algorithm to be substituted independently.

a) Stage 1: Expert Feedback.: For each trajectory τ_i, an expert provides binary feedback: $C(\tau_i) = 1$ if the trajectory contains an unsafe transition, and $C(\tau_i) = 0$ otherwise. Unsafe trajectories are truncated at the first violation point t^*. This labeling protocol reflects how safety monitoring operates in practice—automated systems or human supervisors issue stop commands at the first detected violation [17]—and is far more scalable than dense per-step annotation.

b) Stage 2: Return Decomposition.: We train a sequence model $\hat{C}(s_{0:t}, a_{0:t})$ to predict episodic cost $C(\tau)$ from trajectory prefixes by minimizing $\mathcal{L} = \mathbb{E}_{\tau\sim\mathcal{D}}[(\hat{C}(s_{0:T}, a_{0:T}) - C(\tau))^2]$. Dense per-step costs are then obtained by differencing successive predictions:

$$\tilde{c}_t = \hat{C}(s_{0:t}, a_{0:t}) - \hat{C}(s_{0:t-1}, a_{0:t-1}) + \delta_t, \quad \hat{C}(s_{0:-1}, a_{0:-1}) = 0, \tag{2}$$

where $\delta_T = C(\tau) - \hat{C}(s_{0:T}, a_{0:T})$ at the terminal step (with $\delta_t = 0$ for $t < T$) is a compensation term ensuring exact return equivalence (Definition 1). The intuition is straightforward: if the model's cost prediction increases sharply at timestep t, that transition contains information critical to predicting eventual failure. Following prior work [8], we instantiate $\hat{C}$ as an LSTM, though any sequence architecture (e.g., Transformers) would suffice.

c) Theoretical Results.: A redistributed cost $\tilde{c}$ is return-equivalent to the sparse cost if $\sum_{t=0}^{T} \tilde{c}_t = C(\tau)$ for every trajectory (Definition 1). The costs in Equation (2) satisfy this by construction via a telescoping sum (Lemma 1). This property cascades through the constrained optimization (formal statements and proofs are provided in Appendix A):

Proposition 1 (Constraint Equivalence)**.** *If $\tilde{c}$ is return-equivalent to c^{sparse}, then for any policy π:*

$$\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \tilde{c}_t\right] = \mathbb{E}_{\tau\sim\pi}\left[C(\tau)\right]. \tag{3}$$

Theorem 1 (Policy Invariance)**.** *Let $\mathcal{M}^{\text{sparse}}$ and $\mathcal{M}^{\text{redist}}$ denote CMDPs differing only in instantaneous costs. Then the feasible policy sets are identical and $\pi^*_{\text{sparse}} = \pi^*_{\text{redist}}$. Moreover, for any $\lambda \geq 0$:*

$$\mathcal{L}(\pi, \lambda; \tilde{c}) = \mathcal{L}(\pi, \lambda; c^{\text{sparse}}), \tag{4}$$

so the Lagrangian saddle point (π^*, λ^*) coincides under both formulations.

Crucially, these guarantees hold regardless of the sequence model's prediction accuracy due to the compensation term δ_T. While both formulations define the same optimal policy, redistributed costs provide dense supervision throughout trajectories, yielding a better-conditioned regression target for the cost critic compared to sparse terminal signals (Remark 2).

d) Stage 3: Policy Optimization.: Given the dense cost-augmented dataset $\mathcal{D}_{\text{dense}} = \{(s_t, a_t, r_t, \tilde{c}_t)\}$, we train a constrained offline RL policy. In our instantiation, we employ BCQ-Lagrangian [4], [18], which addresses distributional shift through a VAE-based behavior cloning constraint and enforces safety via an adaptive Lagrange multiplier. The Bellman target becomes $y = r(s,a) + \gamma \max_{a'}[Q_{\theta'}(s',a') - \lambda\, \tilde{c}(s',a')]$, where λ is updated as $\lambda \leftarrow \max(0, \lambda + \alpha_\lambda(C_{\text{batch}} - d))$ based on batch-average cost relative to the budget d. This choice is well-motivated but not essential to RCI.

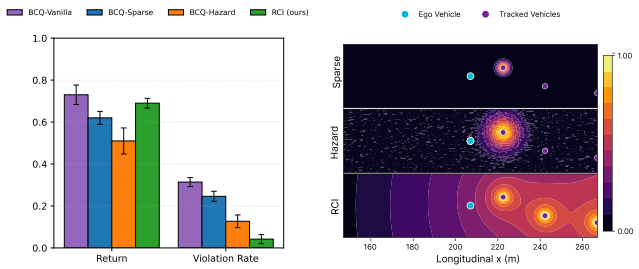
V. Experiments

We evaluate RCI on two benchmark environments: `HighwayEnv` [19], where an ego vehicle navigates highway traffic while avoiding collisions, and `Safe-FetchReach` [20], where a 7-DOF robotic arm reaches target positions while avoiding a spherical hazard region. For each environment we generate 5,000 offline episodes using PPO-trained behavioral policies that optimize task reward while disregarding safety [21], producing aggressive driving with frequent collisions and hazard-ignoring reaching behavior. An automated evaluator identifies the first unsafe transition in each trajectory and truncates all subsequent steps, mimicking expert intervention [17]: a transition is labeled unsafe when the ego approaches within 0.2 units of another vehicle in `HighwayEnv`, or when the end-effector enters the spherical hazard region $\mathcal{H} = \{p \in \mathbb{R}^3 : \|p - h\|_2 \le r\}$ in `Safe-FetchReach`.

Baselines and Protocol. We compare against three baselines under identical BCQ-Lagrangian architectures [4]: (1) *Reward-Only*, which ignores safety costs and sets the budget to infinity; (2) *Sparse*, which uses the raw terminal cost label without redistribution; and (3) *Hazard*, a two-head binary classifier $\tilde{c}(s_t, a_t) = P_1(s_t, a_t) + P_2(s_t, a_t)$, where P_1 predicts whether a state-action pair appears in any unsafe trajectory and P_2 predicts the unsafe termination point specifically, trained with focal loss to address class imbalance [22]. For RCI we instantiate RUDDER-based redistribution with an LSTM sequence model [8]. For each method we sweep the safety budget d over the 10th-50th percentiles of the dataset's trajectory cost distribution in increments of 10, train three independent policies per configuration, and evaluate the lowest-violation policy over 1,000 online episodes using ground-truth safety labels.

A. Safe Navigation

Figure 1 reports normalized return $R_{\text{norm}} = (R - R_{\min})/(R_{\max} - R_{\min})$ pooled across all evaluated policies and seeds just as in [18], and violation rate on `HighwayEnv` under the mixed-composition dataset (equal-proportioned PPO and uniform random rollouts). RCI cuts violation rate compared to Sparse and Hazard while maintaining task return; a two-sample t-test against the unconstrained baseline yields $t = 0.9962$, $p = 0.3483$, indicating no statistically significant return penalty.



(a) `HighwayEnv` results (b) Spatial cost landscapes

Fig. 1. Performance on `HighwayEnv`. (a) Normalized return and violation rate across baselines and RCI; bars show means over 5 seeds with standard error. The two-sample t-test between RCI and BCQ-Vanilla yields $t = 0.9962$, $p = 0.3483$. (b) Learned cost critics as heatmaps over road geometry for Sparse, Hazard, and RCI (top to bottom); warmer colors indicate higher predicted cumulative cost.

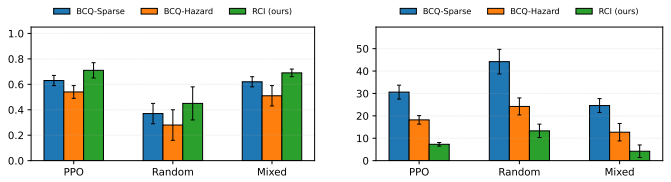
Learned spatial risk structure. Figure 1(b) shows learned cost critics evaluated over the `HighwayEnv` road geometry. Sparse concentrates all cost at the terminal unsafe configuration, with near-zero values elsewhere, even in states already dangerously close to other vehicles. Hazard's classifier captures near-terminal danger in a small neighborhood but decays rapidly. RCI produces globally coherent spatial structure: cost increases gradually as the ego approaches traffic, with high-cost regions extending backward along approach corridors. The redistributed costs treat vehicle proximity as a graded risk signal modulated by temporal context, confirming that the dense supervision recovers meaningful temporal causality from a single terminal bit.

We report two additional experiments on `HighwayEnv` that probe RCI's behavior under realistic deployment conditions: varying data sources and imperfect supervision.

Dataset Composition We evaluate RCI across three behavioral policies with distinct exploration patterns: (i) *PPO*, task-optimized policies producing goal-directed rollouts concentrated around high-reward regions; (ii) *Random*, uniform random action sampling; and (iii) *Mixed*, equal-proportion combinations of PPO and Random rollouts, reflecting realistic offline datasets that aggregate multiple data sources.

RCI's violation reduction holds across all three regimes (Figure 2), indicating that the redistribution mechanism does not depend on the behavior policy producing structured or near-optimal exploration.

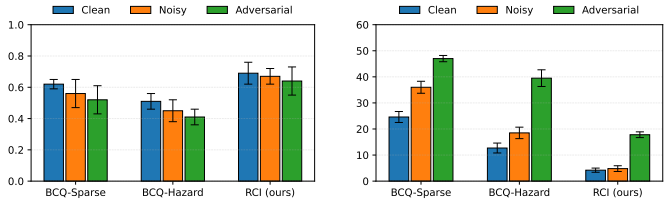
Label Noise We simulate two corruption regimes. *Noisy labels*: for each unsafe trajectory, the stop-point index is shifted by δ steps, where δ is uniformly drawn from $[-15, 15]$ in increments of 5, subject to trajectory bounds. This perturbs the termination label earlier (false positive) or later (false



(a) Normalized return (b) Violation rate

Fig. 2. Dataset composition ablation on `HighwayEnv`. Bars show means over 3 seeds with standard error; each policy is evaluated for 1,000 episodes per seed using ground-truth violations. Safety budget and model selection rule are held fixed across methods.

negative) relative to the true unsafe transition, simulating limited-precision human or automated annotation. *Adversarial labels*: safety labels of a random 20% of trajectories are flipped: unsafe trajectories are relabeled safe (label removed), and safe trajectories are relabeled unsafe with a stop label placed at a uniformly random timestep. Cost budget d is fixed to the value selected under clean supervision to isolate the effect of corruption.



(a) Normalized return (b) Violation rate

Fig. 3. Label-noise ablation on `HighwayEnv`, with cost budget d fixed to the value selected in the main results (Figure 1). Error bars show standard error across 3 seeds; evaluation uses 1,000 episodes and ground-truth safety events.

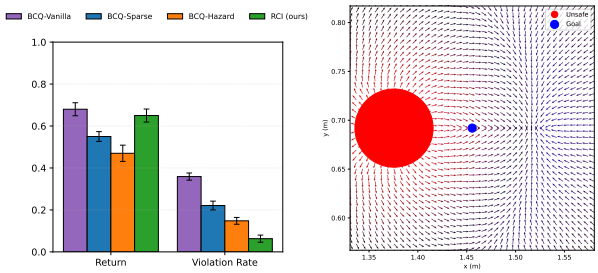
Returns remain stable across noise regimes (Figure 3a) since termination-time noise alters constraint signals without affecting logged rewards. Violation rates on Sparse and Hazard respond more sharply to misaligned supervision than RCI (Figure 3b), consistent with the smoothing effect of return decomposition over per-step label errors.

B. Safe Continuous Control

On `Safe-FetchReach` (Figure 4), we observe the same trend: RCI achieves competitive returns while substantially reducing violation rates. The vector field shows strong repulsive behavior away from the hazard region under a restrictive budget, demonstrating that spatially coherent avoidance emerges from trajectory-level stop-feedback alone, without access to dense cost annotations.

VI. Conclusion

RCI converts trajectory-level stop-feedback into dense per-step costs via return decomposition, enabling constrained offline policy learning from the supervision modality physical AI deployments actually produce. The transformation preserves the feasible policy set and Lagrangian saddle point of the



(a) `Safe-FetchReach` results (b) Policy vector field

Fig. 4. Performance on `Safe-FetchReach`. (a) Normalized return and violation rate; bars show means over 5 seeds with standard error. (b) Vector field of RCI policy under a restrictive safety budget; arrows show action directions, color encodes critic value estimates.

underlying CMDP, and reduces violation rates roughly fivefold on two physical-safety domains without statistically significant return penalties. Stop-feedback is informationally sufficient for safe offline policy learning; richer supervision modalities are not prerequisites.

Three limitations matter. RCI inherits standard offline RL coverage requirements: datasets skewed toward unsafe trajectories risk overly conservative policies, and sparse safe coverage may leave policy optimization underspecified. The redistribution mechanism captures statistical, not causal, associations between trajectory prefixes and violations, identifying risky situations rather than causally hazardous actions. The framework is currently restricted to single binary constraints; multi-constraint extension via per-channel decomposition is straightforward but raises open questions about balancing competing objectives. Future directions include graduated severity feedback, uncertainty-aware redistribution under distributional shift, and more expressive sequence architectures.

References

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained Policy Optimization," May 2017.

[2] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll, "A review of safe reinforcement learning: Methods, theory and applications," 2024. [Online]. Available: https://arxiv.org/abs/2205.10330

[3] E. Altman, "Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program," *Mathematical Methods of Operations Research*, vol. 48, no. 3, pp. 387–417, Dec. 1998.

[4] S. Fujimoto, D. Meger, and D. Precup, "Off-Policy Deep Reinforcement Learning without Exploration," Aug. 2019.

[5] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020.

[6] S. Fujimoto and S. S. Gu, "A Minimalist Approach to Offline Reinforcement Learning," Dec. 2021.

[7] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 1179–1191.

[8] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, "RUDDER: Return Decomposition for Delayed Rewards," Sep. 2019.

[9] Y. Zhang, Y. Du, B. Huang, Z. Wang, J. Wang, M. Fang, and M. Pechenizkiy, "Interpretable Reward Redistribution in Reinforcement Learning: A Causal Approach," Nov. 2023.

[10] S. M. Low and A. Kumar, "TraCeS: Trajectory Based Credit Assignment From Sparse Safety Feedback," Apr. 2025.
[11] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
[12] Y. Wu, G. Tucker, and O. Nachum, "Behavior Regularized Offline Reinforcement Learning," Nov. 2019.
[13] N. Polosky, B. C. D. Silva, M. Fiterau, and J. Jagannath, "Constrained Offline Policy Optimization," in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, Jun. 2022, pp. 17 801–17 810.
[14] H. Xu, X. Zhan, and X. Zhu, "Constraints Penalized Q-learning for Safe Offline Reinforcement Learning," Apr. 2022.
[15] S. R. Chirra, P. Varakantham, and P. Paruchuri, "Safety through feedback in Constrained RL," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, Nov. 2024.
[16] J. García and F. Fernández, "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015.
[17] S. Poletti, A. Testolin, and S. Tschiatschek, "Learning Constraints From Human Stop-Feedback in Reinforcement Learning," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '23. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2023, pp. 2328–2330.
[18] Z. Liu, Z. Guo, H. Lin, Y. Yao, J. Zhu, Z. Cen, H. Hu, W. Yu, T. Zhang, J. Tan, and D. Zhao, "Datasets and Benchmarks for Offline Safe Reinforcement Learning," Jun. 2023.
[19] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.
[20] R. de Lazcano, K. Andreas, J. J. Tai, S. R. Lee, and J. Terry, "Gymnasium robotics," http://github.com/Farama-Foundation/Gymnasium-Robotics, 2024. [Online]. Available: http://github.com/Farama-Foundation/Gymnasium-Robotics
[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017.
[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

APPENDIX A
SIGNAL PRESERVATION

Algorithm 1 summarizes the complete RCI framework. The framework takes as input an offline dataset $\mathcal{D}$, a labeler $\mathcal{L}$ (human or automated), a return decomposition algorithm $\mathcal{A}_{\text{decomp}}$, a constrained offline RL algorithm $\mathcal{A}_{\text{OSRL}}$, and a cost budget d. It outputs a safe policy π.

A critical question still arises: does this redistribution preserve the original supervision signal? In the offline setting, where no additional feedback can be queried, any transformation of the cost signal must guarantee that the resulting constrained optimization problem yields the same optimal policy. We now establish this formally.

Definition 1 (Return-Equivalent Cost Redistribution)**.** A redistributed cost function $\tilde{c} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is return-equivalent to the sparse cost if for every trajectory τ, the sum of redistributed costs equals the episodic cost: $\sum_{t=0}^{T} \tilde{c}_t = C(\tau)$.

Lemma 1 (Return Equivalence)**.** *The redistributed costs defined in Equation* (2) *satisfy Definition 1 for any sequence model* $\hat{C}$.

Algorithm 1 RCI Framework

Input: Dataset $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$, Labeler $\mathcal{L}$, Return decomposition algorithm $\mathcal{A}_{\text{decomp}}$, Constrained Offline RL algorithm $\mathcal{A}_{\text{OSRL}}$, Cost budget d
Output: Safe policy π
for each trajectory $\tau_i \in \mathcal{D}$ **do**
  $F(\tau_i) \leftarrow \mathcal{L}(\tau_i)$ ▷ Feedback Collection
  $C(\tau_i) \leftarrow \begin{cases} 1 & \text{if } F(\tau_i) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$
  for $t = 0, 1, \ldots, T_i$ **do**
    $c^{\text{sparse}}(s_t, a_t) \leftarrow \begin{cases} 1 & \text{if } t \in F(\tau_i) \\ 0 & \text{otherwise} \end{cases}$
  end for
end for
$\hat{C} \leftarrow$ Train sequence model with $\mathcal{A}_{\text{decomp}}$ on $\{(\tau_i, C(\tau_i))\}$ ▷ Return Decomposition
for each trajectory $\tau_i \in \mathcal{D}$ **do**
  for $t = 0, 1, \ldots, T_i$ **do**
    $\tilde{c}_t \leftarrow \hat{C}(s_{0:t}, a_{0:t}) - \hat{C}(s_{0:t-1}, a_{0:t-1})$ ▷ with $\hat{C}(s_{0:-1}, a_{0:-1}) = 0$
  end for
  $\delta_i \leftarrow C(\tau_i) - \hat{C}(s_{0:T_i}, a_{0:T_i})$ ▷ Return-equivalence correction
  $\tilde{c}_{T_i} \leftarrow \tilde{c}_{T_i} + \delta_i$
end for
$\mathcal{D}_{\text{dense}} \leftarrow \{(s_t, a_t, r_t, \tilde{c}_t)\}$ ▷ Constrained Policy Learning
$\pi \leftarrow \mathcal{A}_{\text{OSRL}}(\mathcal{D}_{\text{dense}}, d)$
return π

Proof. Summing across all timesteps yields a telescoping series:

$$\sum_{t=0}^{T} \tilde{c}_t = \sum_{t=0}^{T} \left[\hat{C}(\tau_{0:t}) - \hat{C}(\tau_{0:t-1}) \right] + \delta_T \tag{5}$$

$$= \hat{C}(\tau_{0:T}) - \hat{C}(\tau_{0:-1}) + \delta_T \tag{6}$$

$$= \hat{C}(\tau_{0:T}) + \left[C(\tau) - \hat{C}(\tau_{0:T}) \right] = C(\tau). \tag{7}$$

The boundary condition $\hat{C}(\tau_{0:-1}) = 0$ and the compensation term δ_T ensure exact equality regardless of prediction accuracy. □

This return equivalence property is essential in the offline setting where we cannot query for additional supervision. Unlike interactive learning where the agent can request more safety labels [10], [15], offline learning must extract maximum information from fixed labels. Return equivalence ensures that the inferred costs preserve the original supervision signal without introducing systematic bias, while redistributing it temporally to guide per-step decision-making.

We now show that return equivalence implies policy invariance in the constrained MDP.

Proof. By Lemma 1, return equivalence implies $\sum_{t=0}^{T} \tilde{c}_t = C(\tau)$ for every trajectory τ. Since this equality holds pointwise, the result follows by linearity of expectation. □

Proof. By Proposition 1, $\mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{T}\tilde{c}_t] = \mathbb{E}_{\tau\sim\pi}[C(\tau)]$ for any π, so the feasible sets are identical. Both CMDPs share identical state spaces, action spaces, transition dynamics, reward functions, and discount factors, and maximize the same expected return over identical feasible sets; hence their optimal solutions coincide. □

Since constrained MDPs are commonly solved via Lagrangian relaxation [1], [3], we verify that policy invariance extends to this optimization framework.

a) Lagrangian Equivalence.: For any policy π and multiplier $\lambda \geq 0$, the Lagrangian evaluated using sparse costs equals the Lagrangian evaluated using return-equivalent redistributed costs. Consequently, the saddle point (π^*, λ^*) is identical for both formulations.

Proof. By Proposition 1:

$$\mathcal{L}(\pi, \lambda; \tilde{c}) = \mathbb{E}_{\tau\sim\pi}[R(\tau)] - \lambda\left(\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\tilde{c}_t\right] - d\right) \tag{8}$$

$$= \mathbb{E}_{\tau\sim\pi}[R(\tau)] - \lambda\left(\mathbb{E}_{\tau\sim\pi}[C(\tau)] - d\right) = \mathcal{L}(\pi, \lambda; c^{\text{sparse}}). \tag{9}$$

Since the Lagrangian is identical for all (π, λ), the saddle point problem yields the same solution under both cost specifications. □

These results collectively establish that RCI preserves supervision at multiple levels: trajectory-level (Lemma 1), policy-level (Proposition 1), constraint-level, and optimization-level (Theorem 1). The transformation changes only the temporal allocation of costs—from sparse terminal signals to dense per-step signals—while preserving the aggregate constraint information and the set of optimal policies.

Remark 1. The compensation term δ_T ensures that these guarantees hold regardless of the sequence model's prediction accuracy. While prediction errors affect the distribution of costs across timesteps and may degrade credit assignment quality, they do not compromise supervision preservation. Under perfect prediction, the compensation term vanishes and the redistributed cost exactly quantifies how much each transition increased the probability of eventual violation.

Remark 2. Theorem 1 establish that the Lagrangian and optimal policy are identical under sparse and redistributed costs. However, practical algorithms do not optimize the true Lagrangian directly—they estimate cost expectations using learned critics. Under sparse costs, the cost critic observes non-zero signal only at terminal failure points, making it difficult to learn accurate cost Q-values for earlier states. This is a supervised learning problem with extreme sparsity. Redistributed costs provide dense supervision throughout trajectories, yielding a better-conditioned regression target for the critic. Both formulations define the same optimal policy, but dense costs enable more accurate critic estimation within a fixed sample budget.

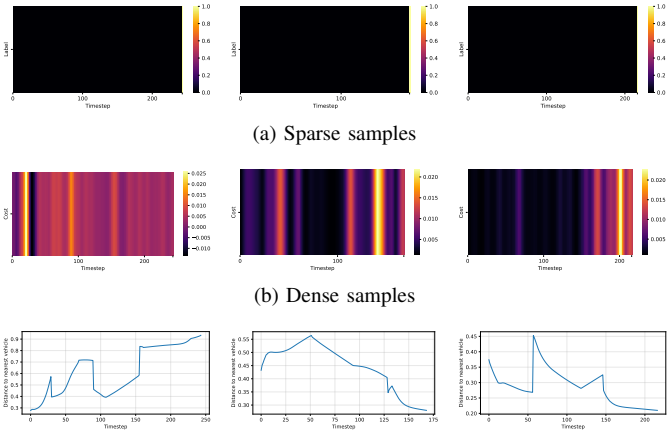
APPENDIX B
QUALITATIVE ANALYSIS

This section provides qualitative evidence supporting the quantitative results presented in Figure 1 and Figure 4. We visualize spatial cost landscapes learned by each method, examine policy trajectories under different safety budgets, and present vector fields of learned policies in the manipulation domain.

A. `HighwayEnv`

A central question motivating our approach is whether return decomposition produces meaningful dense cost signals that identify safety-critical steps. We analyze the quality of redistributed costs through visualization and quantitative assessment.

Figure 5 presents the cost redistribution process on randomly sampled trajectories from `HighwayEnv`. Figure 5a shows the sparse trajectory-level label (a single cost of 1 at the terminal unsafe step), while the Figure 5b shows the dense per-step costs inferred by redistribution via return decomposition.



(a) Sparse samples

(b) Dense samples

(c) Distance to nearest vehicle

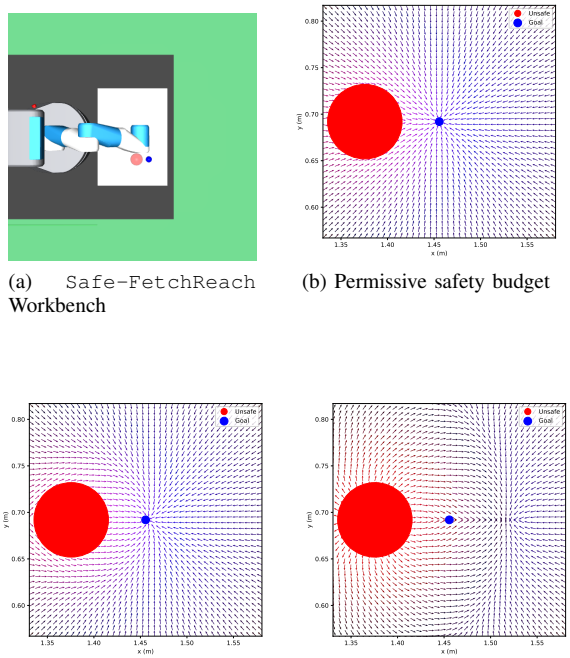
Fig. 5. Cost inference on three sampled trajectories from `HighwayEnv`. (5a) shows the sparse stop-feedback labels (a single terminal cost at the unsafe step). (5b) shows the dense per-step costs inferred via return decomposition, with higher values assigned to precursor actions that contributed to the eventual violation. (5c) shows the distance to the nearest vehicle over time for the same trajectories; note that each trajectory is truncated at the first unsafe transition, as recorded in the offline dataset.

a) Policy Trajectories Under Different Safety Budgets.: Under a restrictive budget ($d = P_{10}$), the policy maintains large following distances and exits the highway entirely; under a balanced budget ($d = P_{30}$), it sustains forward progress while respecting distance constraints. These rollouts confirm that redistributed costs yield interpretable safety–performance trade-offs without retraining the cost inference model.

Figure 7 presents vector fields of the learned RCI policies in `Safe-FetchReach` across three safety budgets from the empirical percentile sweep. Arrows depict the policy's action directions in a two-dimensional projection of the state space, and color encodes the critic's value estimates.

(a) $d = P_{10}$ (b) $d = P_{30}$

Fig. 6. Trajectories from RCI's learned policy in `HighwayEnv` under restrictive (6a) and balanced (6b) safety budgets d, where P_q denotes the q^{th} percentile of the dataset's episodic cost distribution.



(a) `Safe-FetchReach` Workbench (b) Permissive safety budget

(c) Balanced safety budget (d) Restrictive safety budget

Fig. 7. Vector fields of RCI policies in `Safe-FetchReach` across three safety budgets. Arrows depict policy actions and color encodes the critic's value estimates. (7b) Higher budgets permit direct goal approach. (7c) Intermediate budgets balance hazard avoidance with goal-reaching. (7d) Restrictive budgets enforce strong hazard avoidance at the expense of task success.

The vector fields confirm that the learned policies adapt coherently to the imposed constraint. With a permissive budget (Figure 7b), the policy pursues the target directly, tolerating hazard proximity as the constraint permits risk-taking behavior. At an intermediate budget (Figure 7c), the policy steers around the hazard while maintaining progress toward the goal, demonstrating learned avoidance without complete task abandonment. Under a restrictive budget (Figure 7d), the policy exhibits strong repulsive behavior away from the hazard region, prioritizing constraint satisfaction even at the cost of task success. This smooth degradation from goal-pursuit to hazard-avoidance as the constraint tightens reflects the Lagrangian mechanism: as d decreases, the optimal multiplier λ^* increases, reweighting the objective toward cost minimization over reward maximization.

The value estimates confirm this interpretation, with lower values near the hazard region indicating that the critic has learned to associate proximity with elevated risk. That spatially coherent avoidance strategies emerge from trajectory-level stop-feedback alone—without access to dense cost annotations—validates that the redistributed costs carry genuine per-step safety semantics.

APPENDIX C
EXPERIMENTAL DETAILS

A. Benchmark Environments

Both environments share a common annotation protocol: if a safety violation occurs during data collection, the trajectory is truncated at the first unsafe step and assigned an episodic cost label $C(\tau) = 1$; otherwise $C(\tau) = 0$. This stop-feedback structure provides sparse supervision that indicates *that* a violation occurred, but not which earlier decisions contributed to it.

1) `HighwayEnv`: **Task.** The agent controls an ego vehicle on a multilane highway populated with traffic, aiming to maintain forward progress while avoiding unsafe maneuvers.

State space. Each state is a fixed-size array of kinematic features for the ego and $V-1$ nearby vehicles, including presence, relative position (x, y), velocities (v_x, v_y), and orientation $(\cos h, \sin h)$. Features are normalized and expressed in ego-centric coordinates, with zero-padding to maintain fixed dimensionality.

Action space. Actions are two-dimensional continuous controls $a = [a_{\text{throttle}}, \delta_{\text{steer}}]^\top \in [-1, 1]^2$, mapped to acceleration $\dot{v} \in [-5, 5]$ m/s^2 and steering $\delta \in [-0.785, 0.785]$ rad.

Reward function. We modify the native reward to emphasize forward velocity, simulating a safety-unaware ego vehicle:

$$R(s, a) = \alpha \cdot \frac{v - v_{\min}}{v_{\max} - v_{\min}}, \tag{10}$$

with $\alpha > 0$. The collision coefficient is set to zero, so collisions do not incur an explicit reward penalty; however, collisions still terminate the episode via the simulator's built-in termination condition. This ensures that reward-optimized behavioral policies can remain aggressive, while safety is enforced exclusively through the stop-feedback cost signal.

Unsafe criterion. A transition is labeled unsafe when the ego vehicle comes within distance 0.2 of another vehicle:

$$c^{\text{sparse}}(s, a) = \mathbf{1}\Big\{ \min_{j \in \{1,\dots,V-1\}} \|p_{\text{ego}} - p_j\|_2 \le 0.2 \Big\}. \tag{11}$$

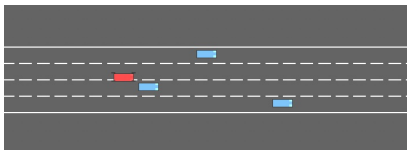


Fig. 8. `HighwayEnv`: a transition is labeled unsafe when the ego vehicle comes within distance 0.2 of another vehicle.

2) `Safe-FetchReach`: **Task.** A 7-DOF Fetch robot must move its end-effector to a sampled Cartesian target. We introduce a spherical hazard region to assess safety.

State space. Observations are tuples $(o, g_{\text{ach}}, g_{\text{des}})$, where $o \in \mathbb{R}^{10}$ encodes end-effector position, finger joint states, and

velocities, and $g_{\text{ach}}, g_{\text{des}} \in \mathbb{R}^3$ denote achieved and desired goals.

Action space. Actions are $a = [\Delta x, \Delta y, \Delta z, a_{\text{grip}}]^\top \in [-1, 1]^4$, with the first three dimensions mapped to Cartesian displacements and the fourth controlling the gripper (unused in this task).

Reward function. The dense shaping reward is:

$$R(s, a) = -\|g_{\text{ach}} - g_{\text{des}}\|_2. \tag{12}$$

Unsafe criterion. We define a spherical hazard region $\mathcal{H} = \{p \in \mathbb{R}^3 : \|p - h\|_2 \leq r\}$ with center h and radius r. A transition is labeled unsafe at the first step where the end-effector enters $\mathcal{H}$:

$$c^{\text{sparse}}(s, a) = \mathbf{1}\{g_{\text{ach}} \in \mathcal{H}\}. \tag{13}$$

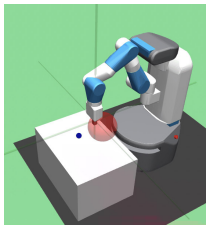


Fig. 9. `Safe-FetchReach`: a transition is labeled unsafe when the end-effector enters the spherical hazard region $\mathcal{H}$.