Split-RL: Local Conflict Resolution in Reinforcement Learning

Benjamin Fuhrer
NVIDIA
bfuhrer@nvidia.com

Chen Tessler
NVIDIA Research
ctessler@nvidia.com

Gal Dalal
NVIDIA Research
gdalal@nvidia.com

Abstract

Reinforcement learning (RL) policies are often trained by mixing contradictory feedback that is localized in space and time. For example, a robot may prioritize speed in open areas versus precision in tight gaps. We introduce Split-RL, a Gradient Boosting Trees (GBT) algorithm that uses guidance labels to locally isolate these conflicting signals. Standard neural networks (NNs) lack the structural mechanism to do this, as their parameters share feedback from all sources. While GBT provides the inductive bias to explicitly partition the state-space per objective, standard tree-fitting procedures aggregate contradictory gradient signals before they reach the leaves. Split-RL leverages this bias to integrate guidance labels into tree construction, routing these updates into disjoint state-space regions and preventing signal interference. Such labels can arise from heuristics, sensor-derived events, or existing rule-based systems, allowing Split-RL to incorporate rule-derived structure while retaining a learned policy. We theoretically show how early gradient aggregation loses objective-specific information and characterize when the Split-RL score isolates conflicting updates. We focus on constrained RL and offline RL with mixed datasets, each balancing two objectives that conflict locally. Across domains with spatial, rare-event, and temporal conflicts, Split-RL consistently outperforms existing methods by achieving low-cost under a fixed episodic budget while maintaining competitive rewards. In offline settings, Split-RL localizes the tradeoff between imitation and policy improvement, outperforming other methods and achieving near-expert performance across varying dataset sizes.

1 Introduction

In many reinforcement learning (RL) settings, a single policy must be trained from multiple signals whose importance varies across state or time. For instance, an autonomous vehicle may optimize speed on highways but prioritize conservative behavior in school zones. Similarly, a controller may balance drug dosage effectiveness while retaining safe levels across time. In these settings, the challenge is not just that objectives conflict, but that the conflict is *localized*: most states support compatible behavior, while specific, coherent regions require a different objective to dominate.

This setting exposes a mismatch with the inductive bias of standard parametric policies, such as neural networks (NNs). Because neural policies share parameters across states, an update driven by one region can also change the policy in another. When signals induce incompatible updates, such as a high-reward task update clashing with a critical safety-cost update, the averaged update can degrade individual objectives through gradient conflict [Yu et al., 2020, Liu et al., 2021, Sener and Koltun, 2018, Navon et al., 2022]. We refer to this failure mode as **premature gradient aggregation**: combining objective-specific signals before the learner has isolated the states in which they should control the update. This premature coupling results in two distinct issues: **signal dilution**, where the majority distribution washes out rare but critical regions, and **gradient interference**, where conflicting updates of similar magnitude cancel into a suboptimal compromise.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

These conflict regions are often easy to identify. A human operator, sensor-derived event, or simple rule-based heuristic can mark the preferred objective to optimize. This makes localized conflict a natural setting for combining rule-derived structure with learning: the rule specifies where an objective should dominate, while the learner determines the policy update within that region. Although these guidance labels provide a clear signal for localizing objectives, standard parametric models such as NNs lack a mechanism to exploit this information because their global updates aggregate signals regardless of local context. Gradient Boosting Trees (GBT) provides the structural machinery to explicitly partition the state-space, yet standard tree-fitting procedures fit only aggregated gradients, combining contradictory updates before they reach the leaves and preventing standard trees from isolating conflicting signals into disjoint regions.

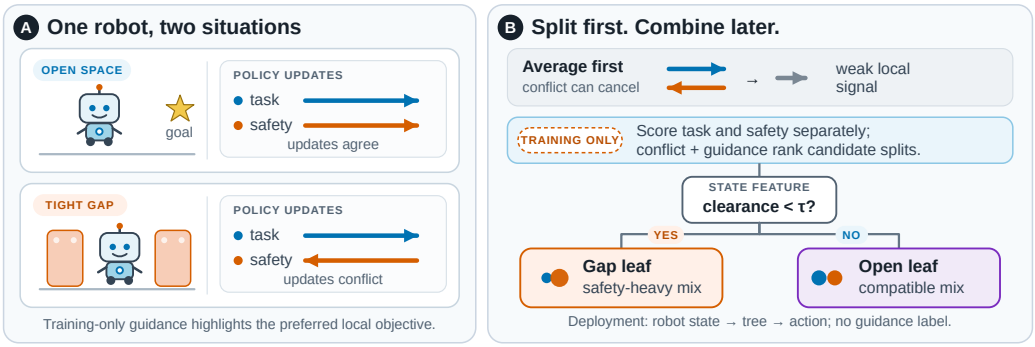


Figure 1: **Split-RL – Guided signal routing. (A)** A robot's task and safety updates align in open space but conflict in a tight gap. **(B)** Split-RL scores objectives separately and uses guidance during training to select tree splits before combining updates within leaves. The gap leaf emphasizes safety, while the open-space leaf retains a mixture of aligned updates.

To address these limitations, we build on recent advances in GBT-based RL [Fuhrer et al., 2025] to introduce **Split-RL**, a framework that uses guided supervision to control update sharing under localized objective conflict. Split-RL integrates guidance labels into the tree-split selection process to enforce structural isolation between conflicting objectives. By prioritizing partitions that route contradictory gradients into separate leaves, the algorithm incorporates rule-based logic into the learning process and shapes which samples share an update. This structural separation prevents the signal interference inherent in globally coupled models, ensuring that each objective dominates only within a localized region, as we illustrate in Figure 1.

This form of conflict appears across RL. In constrained RL, safety violations are often localized, but standard methods [Stooke et al., 2020, Achiam et al., 2017] control expected cost at the policy level rather than deciding which states should receive cost-sensitive updates. Similarly, offline RL requires balancing policy improvement against extrapolation error [Kumar et al., 2020, Kostrikov et al., 2022]. Because standard global regularization applies a uniform level of conservatism, it often washes out expert signals while failing to suppress suboptimal behaviors. Split-RL addresses these failures by allowing the preferred objective to be determined locally. This enables the model to satisfy stringent constraints and recover expert behaviors that are typically lost to the averaging effect of global regularization.

Our contributions are as follows:

1. **Combining rule-based logic with RL.** We formalize a guidance-based framework where rules identify the preferred objective. This lets the learner incorporate rule-based logic while retaining a learned policy.
2. **Split-RL.** We propose Split-RL[1] , a GBT-based algorithm that routes contradictory updates into specialized leaves to prevent premature aggregation. We show how early aggregation loses objective-specific information and when Split-RL isolates conflicting updates.
3. **State-of-the-art results in two distinct problem families.** Across three domains, only Split-RL meets the mean episodic cost budget in all 15 constrained scenarios. It achieves the highest reward within budget in 11. In the offline setting, Split-RL achieves the best return at every dataset size

[1]The Split-RL code is available at `https://github.com/NVlabs/split_rl`.

and reaches near-expert performance. Exposing the same labels to the baselines reproduces neither result.

2 Related Work

We position Split-RL relative to constrained RL, multi-objective learning and gradient conflict, GBT methods in RL, and offline learning from mixed-quality data.

Constrained RL. Constrained RL methods optimize rewards while controlling expected costs, typically falling into broad classes such as Lagrangian-based penalty updates [Stooke et al., 2020, Tessler et al., 2019], trust-region or interior-point strategies [Achiam et al., 2017, Liu et al., 2020, Yang et al., 2022], and feasibility-maintaining methods through projection, shielding, action masking, or barrier functions [Yang et al., 2020b, Alshiekh et al., 2018, Huang and Ontañón, 2020, Cheng et al., 2019]. Broader research examines safety-preserving transformations, reachability analysis, and extreme value theory for rare violations [Sootla et al., 2022, Yu et al., 2022, García and Fernández, 2015, Brunke et al., 2022, Gu et al., 2024, Wachi et al., 2024, Zhang et al., 2025, Gao et al., 2025]. Split-RL is complementary to these optimization-focused frameworks: rather than enforcing safety through global multipliers or projections, it uses guidance labels to structurally isolate contradictory updates at the state-space level.

Multi-objective RL and gradient conflict. Multi-objective RL studies problems with vector-valued returns or learning signals, where policies are optimized under scalarizations, preferences, or Pareto criteria [Roijers et al., 2013, Hayes et al., 2022]. However, in shared parametric models, disparate objectives can induce incompatible updates. Consequently, multi-task and multi-objective learning research focuses on loss balancing, gradient normalization, and gradient projection techniques, such as PCGrad [Yu et al., 2020], to mitigate interference [Chen et al., 2018, Lin et al., 2026, Sener and Koltun, 2018, Navon et al., 2022, Liu et al., 2021, He et al., 2024]. In RL specifically, interference under shared policies has been shown to significantly degrade control performance [Liu et al., 2023, Mao and Zhang, 2026]. Architectural routing methods such as soft modularization and Mixture-of-Experts provide neural alternatives for assigning inputs to specialized subnetworks [Yang et al., 2020a, Ma et al., 2018]. Split-RL offers a distinct alternative by using the partitioning structure of decision trees. Rather than applying auxiliary gradient projections or training a separate routing network, Split-RL integrates guidance labels directly into the tree-building process to isolate contradictory updates within disjoint regions of the state-space.

GBT in RL. GBT is a standard supervised-learning model [Mason et al., 1999, Friedman, 2002], but its use in RL is limited. Prior work includes boosting function approximators for Q-learning [Abel et al., 2016], boosting ensemble policies for contextual bandits [London et al., 2023], and weak-learning reductions for RL [Brukhim et al., 2022]. Fuhrer et al. [2025] recently introduced GBRL, a GBT framework for actor-critic RL. Split-RL builds on it but changes the role of tree construction: guidance-specific targets are scored separately during split selection and combined only after the routing structure is fixed. This lets the model partition the state-space based on conflicting objectives rather than just fitting an aggregated gradient.

Offline RL from mixed-quality data. Offline RL learns from fixed datasets containing trajectories with varying quality. A central challenge is to improve upon the behavior policy while avoiding extrapolation error [Fujimoto et al., 2019, Kumar et al., 2019, Wu et al., 2019, Kumar et al., 2020, Kostrikov et al., 2022]. In mixed-quality datasets, the optimal balance between behavior cloning (BC) and improvement is often state-dependent [Cai et al., 2023, Wolczyk et al., 2024]. Minimalist approaches like TD3+BC [Fujimoto and Gu, 2021] and SPOT [Wu et al., 2022] weight a BC term using a global coefficient, while DICE-based methods re-weight trajectories based on expert-likeness [Kim et al., 2022, Jang et al., 2023]. Recent work has explored "imitating the good and avoiding the bad" using expert and undesirable demonstrations [Hoang et al., 2025, Xudong et al., 2024, Jiang et al., 2025]. Split-RL provides a complementary structural mechanism: it uses data provenance as a guidance label to decide locally whether to apply BC or RL updates during tree construction.

3 Preliminaries

This section establishes the formal framework for RL with multiple objectives and defines the additive ensemble structure used for policy parameterization with GBT.

3.1 RL with Multiple Objectives

We consider an agent interacting with an environment modeled as a fully observable infinite-horizon discounted MDP, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mathbf{r}, \gamma)$, where $\mathbf{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^M$ is a vector-valued reward function. At each time step t, the agent observes state $\mathbf{s}_t$, selects an action $a_t \sim \pi(\cdot \mid s_t)$ according to the policy π, and transitions to $\mathbf{s}_{t+1} \sim P(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$ while receiving the reward vector $\mathbf{r}_t = \mathbf{r}(\mathbf{s}_t, \mathbf{a}_t)$.

We consider K learning objectives $J_1(\pi_F), \ldots, J_K(\pi_F)$. The number of reward components M need not equal the number of objectives K: some objectives are driven by the environment, such as the maximization of rewards, while others are defined independently of the reward, such as BC. For any reward component $m \in \{1, \ldots, M\}$, let $G_t^{(m)} = \sum_{\ell=0}^{\infty} \gamma^\ell r_{t+\ell}^{(m)}$

be its discounted return. The advantage is $A_\pi^{(m)}(s,a) = Q_\pi^{(m)}(s,a) - V_\pi^{(m)}(s)$, where

$$Q_\pi^{(m)}(s,a) = \mathbb{E}_\pi\Big[G_t^{(m)} \mid s_t = s, a_t = a\Big], \qquad V_\pi^{(m)}(s) = \mathbb{E}_\pi\Big[G_t^{(m)} \mid s_t = s\Big].$$

We use GBT policies because tree construction partitions the state space before assigning leaf updates, a natural mechanism for isolating localized conflicts. Although this formulation is stated for general K, in this work we focus on the binary case, where $K = 2$.

3.2 GBT Policies

Split-RL builds on the GBRL framework of Fuhrer et al. [2025]. Following the functional-gradient view of boosting [Mason et al., 1999, Friedman, 2002], an additive tree ensemble parameterizes the policy:

$$F_T(s) = F_0(s) + \sum_{t=1}^{T} \eta_t f_t(s), \tag{1}$$

where each f_t is a decision tree with vector-valued leaves, $\eta_t > 0$ is a step size, and F_T are action logits or distribution parameters of a policy we denote by π_{F_T}. At boosting round t, the functional gradient for a single objective J is expressed as:

$$\nabla_{F_{t-1}} J(\pi_{F_{t-1}}) = \mathbb{E}_{\pi_{F_{t-1}}}\left[\nabla_{F_{t-1}} \log \pi_{F_{t-1}}(a \mid s)\, A(s,a)\right]. \tag{2}$$

The next tree is then chosen by projecting this functional gradient onto the tree class, and the ensemble is updated additively:

$$f_t = \arg\min_f \left\|\nabla_{F_{t-1}} J(\pi_{F_{t-1}}) - f\right\|_2^2, \qquad F_t(s) = F_{t-1}(s) + \eta_t f_t(s). \tag{3}$$

Crucially, tree construction determines the state-space partition before assigning an update to each leaf. Standard GBT fitting expects a single gradient target per sample. With multiple objectives, their gradients must therefore be combined before candidate splits are scored. This aggregation can reduce the split signal precisely when the objectives disagree, as we show below.

Premature aggregation. Let L, R be the children of a candidate split ψ, and let $\mu_k(A) = \frac{1}{|A|}\sum_{i \in A} g_{i,k}$ be the mean gradient of objective k over A. Objective k contributes to ψ the signal $u_k(\psi;\mathcal{N}) = \big(\sqrt{|L|}\,\mu_k(L),\ \sqrt{|R|}\,\mu_k(R)\big)$. Scoring each objective separately with an L2 score gives $S_k(\psi;\mathcal{N}) = \|u_k(\psi;\mathcal{N})\|_2^2$, whereas standard GBT fitting combines the gradients with weights $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$, into a single target before any split is scored: $S_{\text{agg}}(\psi;\mathcal{N}) = \left\|\sum_k \alpha_k u_k(\psi;\mathcal{N})\right\|_2^2$.

Proposition 1 (Early aggregation loses objective-specific information)**.** *For any such weights,*

$$\textstyle\sum_k \alpha_k S_k(\psi;\mathcal{N}) - S_{\text{agg}}(\psi;\mathcal{N}) = \sum_{k<\ell} \alpha_k \alpha_\ell \|u_k(\psi;\mathcal{N}) - u_\ell(\psi;\mathcal{N})\|_2^2 \geq 0.$$

The gap is exactly the pairwise disagreement between objectives: it vanishes when they agree and grows as they diverge. Split-RL therefore keeps objective-specific gradients separate during tree construction. The identity follows by expanding the aggregated update's squared norm. We give the proof and the cosine split-score analogue in Appendix A.

4 Split-RL: Localized Signal Isolation via Guided Partitioning

At each boosting round t, each objective J_k provides a per-sample gradient vector $g_{i,k} \in \mathbb{R}^d$, $k \in \{1, \ldots, K\}$. Guidance is supplied by a label function $y : \mathcal{S} \to \{1, \ldots, K\}$, assigning each state to an objective.

At a node $\mathcal{N}$, the prevalence of label k is $p_k(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \mathbb{I}[y(s_i) = k]$. For a candidate split ψ, Split-RL computes a per-objective split score $S_k(\psi; \mathcal{N})$ and combines the scores by local label prevalence:

$$\bar{S}(\psi; \mathcal{N}) = \sum_{k=1}^{K} p_k(\mathcal{N})\, S_k(\psi; \mathcal{N}).$$

For example, in offline RL, the BC score is computed purely from expert-labeled samples and the policy-improvement score from exploratory samples.

To resolve localized conflicts, we evaluate how effectively a candidate split ψ separates labels when partitioning $\mathcal{N}$ into children L and R. Letting $n_{A,k}$ be the number of label-k samples in subset A, we use the categorical sum-of-squares error:

$$\mathrm{SSE}(A) = |A| - \frac{1}{|A|} \sum_{k=1}^{K} n_{A,k}^2, \qquad \mathrm{SSE}(\emptyset) = 0.$$

We define the normalized label impurity after split ψ as

$$H(\psi; \mathcal{N}) = \frac{\mathrm{SSE}(L) + \mathrm{SSE}(R)}{\mathrm{SSE}(\mathcal{N}) + \epsilon}.$$

$H(\psi; \mathcal{N})$ measures how effectively a split isolates guidance labels: $H = 0$ for a label-pure split and $H \to 1$ for a completely mixed split.

However, explicitly forcing purity is potentially harmful when gradients are aligned. To balance this, we define the node conflict coefficient

$$\rho(\mathcal{N}) = \left[1 - \frac{\left\| \sum_{k=1}^{K} \mu_k(\mathcal{N}) \right\|_2^2}{\sum_{k=1}^{K} \|\mu_k(\mathcal{N})\|_2^2} \right]_+,$$

where $\mu_k(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} g_{i,k}$ is the mean gradient vector for objective k. The coefficient $\rho(\mathcal{N})$ is high when objective gradients cancel and low when they align, making the purity regularization conflict-dependent.

The Split-RL split score is then:

$$S_{\mathrm{Split}}(\psi; \mathcal{N}) = \bar{S}(\psi; \mathcal{N}) \left(1 - \mathrm{clip}_{[0,1]} \left(\lambda_0 \rho(\mathcal{N}) H(\psi; \mathcal{N}) \right) \right), \qquad \lambda_0 \geq 0. \tag{4}$$

Here, the single hyperparameter λ_0 controls the purity regularization strength. We now show the threshold at which a label-pure split overtakes an impure one.

Proposition 2 (Regularization threshold for a purifying split)**.** *Let $\rho(\mathcal{N}) > 0$ and let ψ_{pure} satisfy $H(\psi_{\mathrm{pure}}; \mathcal{N}) = 0$ and $\bar{S}(\psi_{\mathrm{pure}}; \mathcal{N}) > 0$. Then ψ_{pure} scores at least as high as every impure candidate at $\mathcal{N}$ whenever*

$$\lambda_0 > \max_{\substack{\psi:\, H(\psi;\mathcal{N})>0, \\ \bar{S}(\psi;\mathcal{N}) > \bar{S}(\psi_{\mathrm{pure}};\mathcal{N})}} \frac{1 - \bar{S}(\psi_{\mathrm{pure}}; \mathcal{N}) / \bar{S}(\psi; \mathcal{N})}{\rho(\mathcal{N}) H(\psi; \mathcal{N})}.$$

The maximum is over an empty set when no impure candidate has a higher raw score, so any $\lambda_0 \geq 0$ suffices in that case.

Once the tree topology is fixed, Split-RL computes the final leaf update as a weighted average across labels:

$$v(\mathcal{L}) = \sum_{k=1}^{K} p_k(\mathcal{L})\, \mu_k(\mathcal{L}),$$

where $\mu_k(\mathcal{L})$ is the mean gradient for objective k. Because aggregation now occurs within a leaf, its label purity determines how closely the applied update follows a particular objective.

Proposition 3 (Purer leaves track the guided objective)**.** *For any label j,*

$$\|v(\mathcal{L}) - \mu_j(\mathcal{L})\|_2 \leq \sum_{k \neq j} p_k(\mathcal{L}) \, \|\mu_k(\mathcal{L}) - \mu_j(\mathcal{L})\|_2,$$

so a leaf that is pure for j applies exactly $\mu_j(\mathcal{L})$.

Together, these propositions characterize how Split-RL preserves objective-specific signals during tree construction and turns purer partitions into updates that are better aligned with the guided objective. We prove all three propositions in Appendix A.

5 Experiments

We evaluate Split-RL in settings where objective conflict is localized and observable. In constrained RL, the signals are reward and cost, and guidance labels mark states where cost-sensitive updates should dominate. In offline RL, the signals are BC and policy improvement, and labels indicate expert versus non-expert provenance. All label rules are fixed before training and depend only on the observable state or transition provenance. Full environment definitions, label rules, costs, and hyperparameters are in Appendices B.1–C.

5.1 Constrained RL

We implement Split-RL within PPO [Schulman et al., 2017], adding a cost critic for constrained tasks. We compare against neural CPO [Achiam et al., 2017], CUP [Yang et al., 2022], IPO [Liu et al., 2020], and PPO-Lagrangian [Stooke et al., 2020], and include a GBT PPO-Lagrangian baseline to isolate the effect of guided partitioning.

MiniGrid. We construct three controlled MiniGrid tasks [Chevalier-Boisvert et al., 2023] to isolate localized signal conflict. DynamicCrossing creates a reward–cost routing conflict around a lava barrier with a safe opening, where a moving obstacle near the opening adds timing difficulty. FragileCrossing creates a context-dependent hazard, where ice is dangerous only while the agent carries an object. Corner creates a sparse localized conflict by placing one reward behind a narrow lava entrance.

Traffic signal control. We create traffic-signal control scenarios using SUMO-RL [Alegre, 2019]. We use single-intersection, grid, and arterial networks, and introduce bus, premium, and convoy priority events. These scenarios create localized priority conflicts within ordinary traffic: the controller must sometimes sacrifice throughput to serve a specific vehicle class. Special vehicles are injected in half of the episodes, producing infrequent event-based cost signals.

Building energy management. We create reserve-control scenarios using CityLearn [Nweye et al., 2024]. A central agent controls battery storage across three buildings with solar generation over one month of hourly data. The scenarios create temporal conflicts between electricity-cost minimization and reserve maintenance, with labels marking deterministic reserve-critical windows that require sustained anticipatory behavior.

Ablations and interpretability. We test whether Split-RL's advantage comes from access to the guidance label by giving the same label to all five constrained-RL baselines, either as an extra observation feature (*Label-in-Obs*) or as a loss mask selecting each sample's objective (*Masked*). We ablate the label-purity regularization strength λ_0 in CityLearn Solar Ramp Reserve to isolate the role of the purity penalty. To test robustness to imperfect guidance, we corrupt the label on MiniGrid: each label is independently redrawn uniformly at random with probability p before it reaches split selection. Finally, we inspect the trained trees on MiniGrid, whose states are easy to visualize as a grid. At tree construction, each leaf stores the proportion of training samples per label. We run evaluation episodes, pass every visited state through the ensemble, and average the stored label proportions across the leaves reached, which we call the *routing density*.

5.2 Offline RL

The offline experiments test whether Split-RL can imitate expert transitions while using non-expert transitions for coverage and policy improvement. Split-RL uses transition provenance as the guidance

label to switch between BC targets for expert data and policy-improvement targets for non-expert data. For policy improvement, we use Advantage-Weighted Regression (AWR) [Peng et al., 2019], which weights supervised regression by exponentiated advantages to provide a stable, off-policy actor-critic target that is a natural fit for decision trees. We evaluate Split-RL across varying dataset sizes of Atari RAM [Bellemare et al., 2013]. Baselines include CQL [Kumar et al., 2020], BCQ [Fujimoto et al., 2019], Decision Transformer (DT) [Chen et al., 2021], and AWR. To examine the role of structural partitioning, we also compare Split-RL against three update-sharing ablations: (i) **All**, where both BC and AWR targets are computed and averaged for every transition; (ii) **Masked**, where objectives are strictly partitioned by label, applying BC only to expert transitions and AWR only to non-expert; and (iii) **Expert-Only BC**, where AWR is applied globally but BC is restricted exclusively to expert transitions. For all experiments, we report the expert-normalized return.

6 Results

We report the full results for constrained and offline RL in Appendix D. Constrained RL results are averaged over 15 seeds per configuration and offline results over 50 evaluation episodes (Appendix C).

6.1 Constrained RL

Unlike standard constrained RL, which maximizes reward subject to a fixed episodic cost budget, Split-RL uses a per-state guidance label indicating when the cost signal should dominate the local update. This results in Split-RL being the only method feasible in all scenarios, as shown in Table 1.

Table 1: **Constrained RL summary.** Domain columns count feasible scenarios; i.e., mean episodic cost below the limit. The final column counts scenarios where a method has the highest reward among feasible methods; ties are counted for all tied methods.

Method	MiniGrid	SUMO-RL	CityLearn	Feasible	Best feasible reward
CPO	0/3	7/9	0/3	7/15	0/15
CUP	2/3	8/9	0/3	10/15	0/15
IPO	0/3	8/9	0/3	8/15	0/15
PPO-Lag	1/3	4/9	0/3	5/15	3/15
PPO-Lag (GBT)	**3/3**	3/9	0/3	6/15	1/15
Split-RL (Ours)	**3/3**	**9/9**	**3/3**	**15/15**	**11/15**

MiniGrid. Split-RL matches the best feasible operating point on DynamicCrossing and gives the strongest reward–cost profile on FragileCrossing. Corner is the main exception: Split-RL remains safe, but PPO-Lagrangian (GBT) attains a higher feasible reward. These results suggest that Split-RL is most useful when feasibility requires a state- or context-dependent switch between progress-seeking and cost-sensitive updates. When a conservative feasible policy is already easy to represent, as in Corner, the benefit of guided local routing is smaller.

SUMO-RL. Split-RL reliably produces feasible priority-aware policies, achieving feasibility in all nine network–event pairs, the lowest cost in four, and the highest feasible reward in every Arterial scenario (Appendix D). As shown in Figure 2, Split-RL maintains low cost even when priority events are rare and remains feasible across all tested conflict probabilities in the premium scenario.

CityLearn. CityLearn exposes the clearest reward–feasibility tradeoff. Split-RL is feasible in all three scenarios, while no baseline is feasible in any.

Across all three domains, Split-RL substantially outperforms PPO-Lagrangian (GBT), showing that the tree function class alone does not explain the gains.

Does Split-RL have access to privileged information? Giving the baselines the label does not produce a systematic gain (Table 2). Under either condition, every baseline changes by at most one feasible scenario except CPO under Masked, which gains four. These gains are not consistent across scenarios: masking may reduce competing contributions to CPO's constrained update, stabilizing high-variance runs and trading reward for feasibility in some scenarios while degrading others.

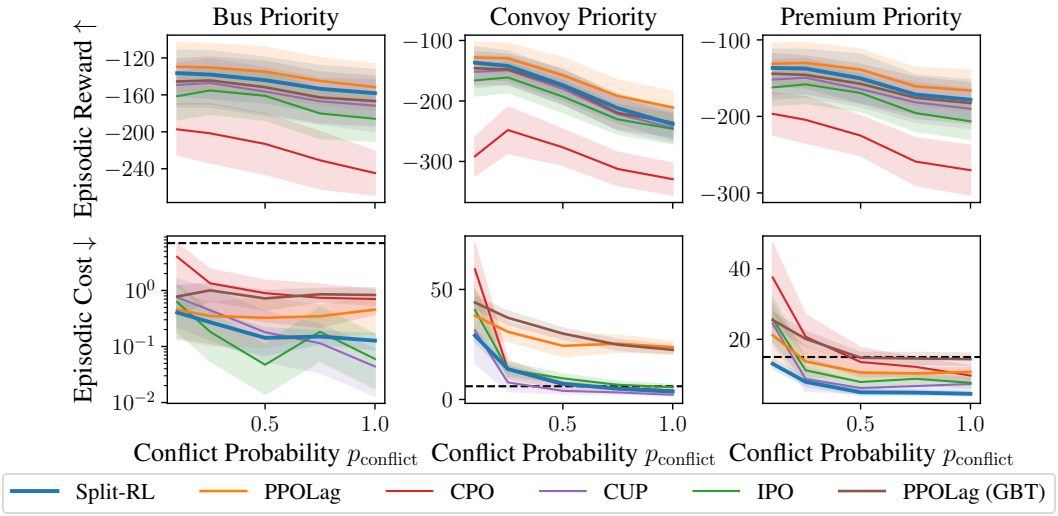


Figure 2: **Effect of priority-event frequency in SUMO-RL 4x4Grid network.** Mean episodic reward (top) and episodic cost (bottom) as conflict probability p_{conflict} increases. Shading shows ± 1 std. Dashed lines mark cost limits. Split-RL keeps cost low even when conflicts are rare, while maintaining competitive throughput.

PPO-Lag (GBT), which shares Split-RL's tree backbone and receives the same labels, loses a feasible scenario under both conditions. Full reward–cost results are in Appendix F.3. Overall, label access does not reproduce Split-RL's advantage.

Table 2: **Does access to the label alone explain Split-RL's gap?** Feasible scenario counts out of 15 (cost $\leq$ limit, Table 11) for each baseline under three training conditions.

Algorithm	No-label	Label-in-Obs	Masked
CPO	7/15	7/15	11/15
CUP	10/15	9/15	9/15
IPO	8/15	9/15	9/15
PPOLag	5/15	5/15	6/15
PPOLag (GBT)	6/15	5/15	5/15

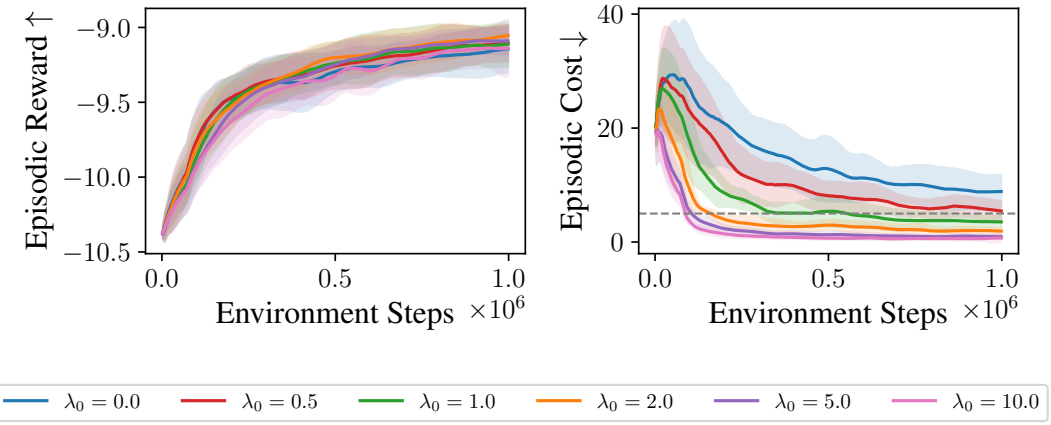


Figure 3: **Label-purity regularization ablation in CityLearn Solar Ramp Reserve.** Mean episodic reward and cost over training for different λ_0. Shading: ± 1 std. Dashed line: cost limit. Nonzero λ_0 substantially reduces cost by penalizing label-mixing splits in conflict windows.

Does the purity penalty matter? Figure 3 shows that it does. Without the penalty ($\lambda_0 = 0$), or with a small λ_0, Split-RL remains above the cost limit in Solar Ramp Reserve. Once λ_0 is sufficiently large, it becomes feasible without a loss in reward. This mirrors the threshold behavior in Proposition 2: increasing λ_0 can make a label-pure split overtake an impure one when the objectives conflict.

How robust is Split-RL to label noise? Performance is stable up to $p = 0.25$: cost stays near its noise-free value, and reward is essentially unchanged across the whole sweep (Figure 4). Cost rises sharply once $p \geq 0.5$. This matches the failure mode Split-RL targets: cost is the rare but critical signal that the labels protect from dilution. An uninformative label produces arbitrary routing rather than a fallback to unguided training.

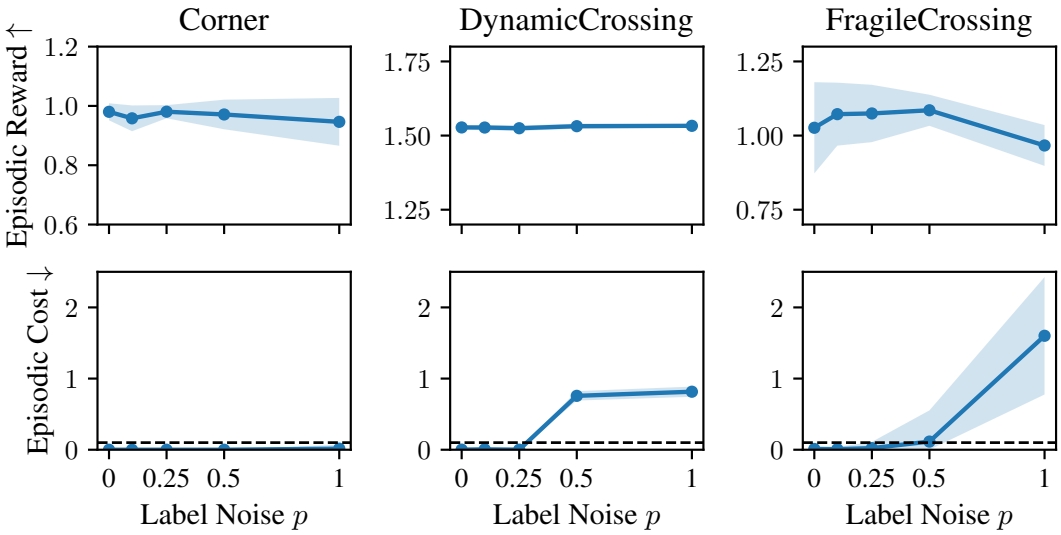


Figure 4: **Robustness to label noise.** Mean episodic reward and cost (± 1 std) on MiniGrid. Dashed line: cost limit. Split-RL's performance is stable up to $p = 0.25$.

Do the trees route by conflict structure? In FragileCrossing, cost-labelled states (Figure 5b) reach leaves with high cost-label density (Figure 5c). The trees recover the conflict structure from the observation alone, without the label at evaluation. Appendix E shows the other two environments.

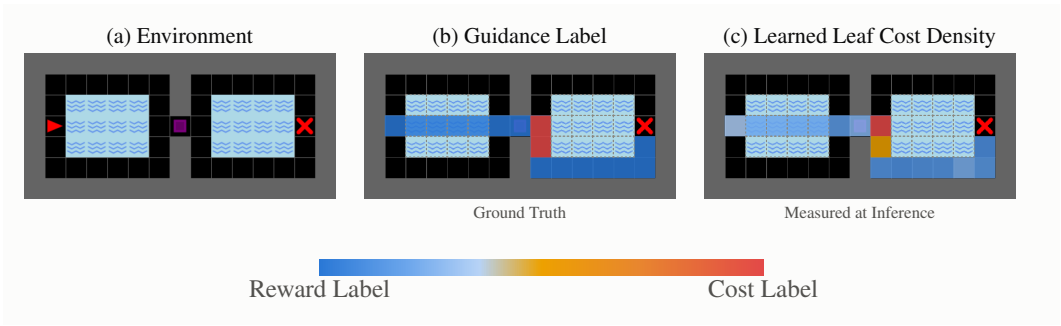


Figure 5: **Guided routing density under policy rollouts (FragileCrossing).** Fraction of cost-labelled training samples in the leaves reached by each visited state, over evaluation episodes. Higher density (red) on labelled-cost states shows the ensemble routes by conflict structure.

6.2 Offline RL

As shown in Table 3, Split-RL achieves the highest expert-normalized return on average, outperforming DT by 43% and GBT AWR by 86%. It is near-expert from 50K transitions onward, while standard baselines struggle with the imitation–improvement tradeoff.

Table 3: **Atari RAM offline learning: data-budget profile.** Mean ± std of expert-normalized return across 11 Atari games at each offline dataset size. **Bold** = best per column. Split-RL produces the best return at every dataset size and exhibits near-expert performance on average.

		Dataset Size						
	Method	**10K**	**20K**	**50K**	**100K**	**200K**	**Full**	**Avg.**
NN	CQL	0.27 ± 0.27	0.24 ± 0.23	0.27 ± 0.30	0.35 ± 0.32	0.42 ± 0.28	0.42 ± 0.29	0.33
	DT	0.52 ± 0.27	0.58 ± 0.32	0.71 ± 0.35	0.73 ± 0.35	0.65 ± 0.35	0.69 ± 0.40	0.65
	AWR	0.43 ± 0.37	0.53 ± 0.33	0.58 ± 0.31	0.60 ± 0.37	0.59 ± 0.30	0.67 ± 0.26	0.57
	BCQ	0.35 ± 0.36	0.34 ± 0.29	0.24 ± 0.33	0.22 ± 0.27	0.18 ± 0.25	0.33 ± 0.33	0.28
GBT	AWR	0.48 ± 0.30	0.57 ± 0.30	0.52 ± 0.32	0.48 ± 0.32	0.50 ± 0.35	0.46 ± 0.28	0.50
	Split-RL (Ours)	$\mathbf{0.75 \pm 0.32}$	$\mathbf{0.86 \pm 0.26}$	$\mathbf{0.93 \pm 0.13}$	$\mathbf{1.02 \pm 0.05}$	$\mathbf{1.02 \pm 0.04}$	$\mathbf{0.99 \pm 0.08}$	**0.93**

As shown in Table 4, where the BC target is applied matters, but it is not sufficient. Applying BC to all samples hurts performance because it clones non-expert behavior. Restricting BC to expert samples gives the strongest AWR+BC baseline, while the fully masked variant is too restrictive. Split-RL still performs best. Masking prevents the two objectives from acting on the same transition, but only Split-RL also keeps their split scores separate during tree construction, preventing premature aggregation.

Table 4: **BC target ablation.** Mean ± std of expert-normalized return across 11 Atari games. *All*: both signals on all samples; *Expert*: BC on expert samples only; *Masked*: BC on expert, AWR on non-expert exclusively. **Bold** = best.

	NN: AWR + BC			**GBT: AWR + BC**			**Split-RL (Ours)**
	All	Expert	Masked	All	Expert	Masked	
Return	0.68 ± 0.28	0.81 ± 0.29	0.59 ± 0.42	0.68 ± 0.29	0.88 ± 0.25	0.52 ± 0.35	$\mathbf{0.99 \pm 0.08}$

7 Limitations

Split-RL assumes guidance labels are deterministic and aligned with the objective conflict. A label unrelated to the conflict, or one missing hidden context, can prevent effective learning. It also targets conflicts that are localized and identifiable, so where objectives conflict globally and uniformly, explicit partitioning may offer little advantage over Lagrangian or projection-based methods. The framework is also tied to GBT policies and structured observations: GBT excels on the tabular inputs studied here, but extending Split-RL to pixel-based inputs would likely require a hybrid architecture in which a representation learner extracts features for the tree-based policy. Finally, our experiments use $K = 2$ objectives. Because each split is binary, routing K label categories into separate leaves requires a tree depth that grows with K.

8 Conclusion

We introduced Split-RL, a framework that reframes localized objective conflict as a problem of update sharing. By integrating guidance labels into tree construction, Split-RL routes contradictory signals into disjoint regions of the state-space before aggregation. Our results show that this structural isolation is useful across distinct problem families: Split-RL resolves the imitation–improvement tradeoff in offline RL and achieves consistent feasibility in constrained environments. Crucially, it incorporates rule-based logic without replacing the learned policy: guidance labels define where different objectives control the update, while the learner still optimizes the policy within those regions. When conflicts are localized, deciding which samples share an update can be as important as optimizing the objective itself.

Acknowledgments and Disclosure of Funding

This work was funded by NVIDIA, where all authors are employed. The authors received no third-party funding in support of this work during the past 36 months, and have no financial relationships outside the submitted work that could be perceived as influencing it.

References

David Abel, Alekh Agarwal, Fernando Diaz, Akshay Krishnamurthy, and Robert E. Schapire. Exploratory gradient boosting for reinforcement learning in complex domains, 2016. URL https://arxiv.org/abs/1603.04119.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 22–31. JMLR.org, 2017.

Lucas N. Alegre. SUMO-RL. https://github.com/LucasAlegre/sumo-rl, 2019.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

Nataly Brukhim, Elad Hazan, and Karan Singh. A boosting approach to reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 33806–33817. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/daf8364f0715a41a469c677c0adc4754-Paper-Conference.pdf.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Yuanying Cai, Chuheng Zhang, Li Zhao, Wei Shen, Xuyun Zhang, Lei Song, Jiang Bian, Tao Qin, and Tie-Yan Liu. Td3 with reverse kl regularizer for offline reinforcement learning from mixed datasets. In *IEEE International Conference on Data Mining (Best Student Paper Runner-Up Awards)*, January 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3387–3395, 2019.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002. ISSN 0167-9473. doi: 10.1016/S0167-9473(01)00065-2. Nonlinear Methods and Data Mining.

Benjamin Fuhrer, Chen Tessler, and Gal Dalal. Gradient boosting reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 17960–17985. PMLR, 2025.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2019.

Shiqing Gao, Yihang Zhou, Shuai Shao, Haoyu Luo, Yiheng Bing, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. Extreme value policy optimization for safe reinforcement learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 18772–18793. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gao25v.html.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL https://jmlr.org/papers/v16/garcia15a.html.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Yun He, Xuxing Chen, Jiayi Xu, Renqin Cai, Yiling You, Jennifer Cao, Minhui Huang, Liu Yang, Yiqun Liu, Xiaoyi Liu, et al. Multibalance: Multi-objective gradient balancing in industrial-scale multi-task recommendation system. *arXiv preprint arXiv:2411.11871*, 2024.

Huy Hoang, Tien Mai, Pradeep Varakantham, and Tanvi Verma. Learning what to do and what not to do: Offline imitation from expert and undesirable demonstrations. *arXiv preprint arXiv:2505.21182*, 2025.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*, 2020.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Youngsoo Jang, Geon-Hyeong Kim, Jongmin Lee, Sungryull Sohn, Byoungjip Kim, Honglak Lee, and Moontae Lee. SafeDICE: Offline safe imitation learning with non-preferred demonstrations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=toEGuA9Qfn.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *arXiv preprint arXiv:2305.09304*, 2023.

Xingyu Jiang, Ning Gao, Xiuhui Zhang, Hongkun Dou, and Yue Deng. Value-aligned behavior cloning for offline reinforcement learning via bi-level optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=elTJBP7Fbv.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BrPdX1bDZkQ.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Baijiong Lin, Weisen Jiang, Feiyang Ye, Yu Zhang, Pengguang Chen, Ying-Cong Chen, Shu Liu, Ivor W. Tsang, and James T. Kwok. Dual-balancing for multi-task learning. *Neural Networks*, 195:108317, 2026. ISSN 0893-6080. doi: 10.1016/j.neunet.2025.108317.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

Vincent Liu, Han Wang, Ruo Yu Tao, Khurram Javed, Adam White, and Martha White. Measuring and mitigating interference in reinforcement learning. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup, editors, *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 781–795. PMLR, 22–25 Aug 2023. URL https://proceedings.mlr.press/v232/liu23a.html.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4940–4947, 2020.

Ben London, Levi Lu, Ted Sandler, and Thorsten Joachims. Boosted off-policy learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5614–5640. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/london23a.html.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018. doi: 10.1109/ITSC.2018.8569938.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1930–1939, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220007. URL https://doi.org/10.1145/3219819.3220007.

Yihuan Mao and Chongjie Zhang. A bayesian fast-slow framework to mitigate interference in non-stationary reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=MmjW4VGKbh.

Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pages 16428–16446, 2022.

Kingsley Nweye, Kathryn Kaspar, Giacomo Buscemi, Tiago Fonseca, Giuseppe Pinto, Dipanjan Ghose, Satvik Duddukuru, Pavani Pratapa, Han Li, Javad Mohammadi, Luis Lino Ferreira, Tianzhen Hong, Mohamed Ouf, Alfonso Capozzoli, and Zoltan Nagy. Citylearn v2: energy-flexible, resilient, occupant-centric, and carbon-aware management of grid-interactive communities. *Journal of Building Performance Simulation*, 0(0):1–22, 2024. doi: 10.1080/19401493.2024.2418813. URL https://doi.org/10.1080/19401493.2024.2418813.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Takuma Seno and Michita Imai. d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315):1–20, 2022. URL http://jmlr.org/papers/v23/22-0017.html.

Aivar Sootla, Alexander I Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David H Mguni, Jun Wang, and Haitham Ammar. Sauté rl: Almost surely safe reinforcement learning using state augmentation. In *International Conference on Machine Learning*, pages 20423–20443. PMLR, 2022.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

Akifumi Wachi, Xun Shen, and Yanan Sui. A survey of constraint formulations in safe reinforcement learning. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8262–8271. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/913. URL https://doi.org/10.24963/ijcai.2024/913. Survey Track.

Maciej Wolczyk, Bartłomiej Cupiał, Mateusz Ostaszewski, Michał Bortkiewicz, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Fine-tuning reinforcement learning models is secretly a forgetting mitigation problem. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53039–53078. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wolczyk24a.html.

Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:31278–31291, 2022.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. In *International Conference on Learning Representations*, 2019.

Gong Xudong, Feng Dawei, Kele Xu, Yuanzhao Zhai, Chengkang Yao, Weijia Wang, Bo Ding, and Huaimin Wang. Iterative regularized policy optimization with imperfect demonstrations. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 55547–55568. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xudong24a.html.

Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35:9111–9124, 2022.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33:4767–4777, 2020a.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020b.

Dongjie Yu, Haitong Ma, Shengbo Li, and Jianyu Chen. Reachability constrained reinforcement learning. In *International conference on machine learning*, pages 25636–25655. PMLR, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Changxin Zhang, Xinglong Zhang, Yixing Lan, Hao Gao, and Xin Xu. Game-theoretic constrained policy optimization for safe reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17990–18004, 2025. doi: 10.1109/TNNLS.2025.3586603.

A Theoretical Properties of Split-RL

In this section, we formalize three properties of Split-RL. The first shows that early aggregation can lose objective-specific information; we establish the L2 identity and then give the equivalent comparison for the cosine split score used in our experiments. The second shows how the regularization strength λ_0 can make a purifying split win when conflict is present. The third shows that purer leaves make the applied update follow the guided objective update more closely.

Let L, R be the children of a candidate split ψ at node $\mathcal{N}$, and let $\mu_k(A) = \frac{1}{|A|}\sum_{i\in A} g_{i,k}$ be the mean gradient of objective k over A. Objective k contributes to ψ the signal

$$u_k(\psi;\mathcal{N}) = \big(\sqrt{|L|}\,\mu_k(L),\ \sqrt{|R|}\,\mu_k(R)\big),$$

scored separately by $S_k(\psi;\mathcal{N}) = \|u_k(\psi;\mathcal{N})\|_2^2$, while aggregating the gradients with weights $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$, before scoring gives

$$S_{\text{agg}}(\psi;\mathcal{N}) = \Big\|\sum_{k=1}^{K} \alpha_k u_k(\psi;\mathcal{N})\Big\|_2^2.$$

Below, $\sum_{k<\ell}$ ranges over distinct unordered pairs of objectives, so each pair is counted once.

Proposition 1. *For any such weights,*

$$\textstyle\sum_k \alpha_k S_k(\psi;\mathcal{N}) - S_{\text{agg}}(\psi;\mathcal{N}) = \sum_{k<\ell} \alpha_k\alpha_\ell \,\|u_k(\psi;\mathcal{N}) - u_\ell(\psi;\mathcal{N})\|_2^2 \geq 0.$$

Proof. Write $u_k = u_k(\psi;\mathcal{N})$. By definition,

$$\sum_{k=1}^{K} \alpha_k S_k(\psi;\mathcal{N}) = \sum_{k=1}^{K} \alpha_k \|u_k\|_2^2, \qquad S_{\text{agg}}(\psi;\mathcal{N}) = \left\|\sum_{k=1}^{K} \alpha_k u_k\right\|_2^2.$$

Expanding the aggregated score gives

$$S_{\text{agg}}(\psi;\mathcal{N}) = \sum_{k=1}^{K} \alpha_k^2 \|u_k\|_2^2 + 2\sum_{k<\ell} \alpha_k\alpha_\ell\, u_k^\top u_\ell.$$

Therefore,

$$\sum_{k=1}^{K} \alpha_k S_k(\psi;\mathcal{N}) - S_{\text{agg}}(\psi;\mathcal{N}) = \sum_{k=1}^{K} \alpha_k(1-\alpha_k)\|u_k\|_2^2 - 2\sum_{k<\ell} \alpha_k\alpha_\ell\, u_k^\top u_\ell.$$

Now expand the right-hand side we want:

$$\sum_{k<\ell} \alpha_k\alpha_\ell \,\|u_k - u_\ell\|_2^2 = \sum_{k<\ell} \alpha_k\alpha_\ell\big(\|u_k\|_2^2 + \|u_\ell\|_2^2 - 2u_k^\top u_\ell\big).$$

For a fixed k, the term $\|u_k\|_2^2$ appears once for every $\ell \neq k$, and its total coefficient is

$$\sum_{\ell\neq k} \alpha_k\alpha_\ell = \alpha_k \sum_{\ell\neq k} \alpha_\ell = \alpha_k(1-\alpha_k),$$

because $\sum_{\ell=1}^{K} \alpha_\ell = 1$. Hence

$$\sum_{k<\ell} \alpha_k\alpha_\ell\big(\|u_k\|_2^2 + \|u_\ell\|_2^2\big) = \sum_{k=1}^{K} \alpha_k(1-\alpha_k)\|u_k\|_2^2.$$

Substituting this back yields

$$\sum_{k<\ell} \alpha_k\alpha_\ell \,\|u_k - u_\ell\|_2^2 = \sum_{k=1}^{K} \alpha_k(1-\alpha_k)\|u_k\|_2^2 - 2\sum_{k<\ell} \alpha_k\alpha_\ell\, u_k^\top u_\ell,$$

which is exactly $\sum_{k=1}^{K} \alpha_k S_k(\psi;\mathcal{N}) - S_{\text{agg}}(\psi;\mathcal{N})$. □

Cosine split score. Let $d_k = (\sum_{i\in\mathcal{N}} \|g_{i,k}\|_2^2)^{1/2}$. Scoring ψ by the cosine similarity between the gradients and the child means they are fitted with gives

$$S_k^{\cos}(\psi;\mathcal{N}) = \frac{\sum_{A\in\{L,R\}} \sum_{i\in A} g_{i,k}^\top \mu_k(A)}{d_k \big(\sum_{A\in\{L,R\}} |A|\, \|\mu_k(A)\|_2^2\big)^{1/2}} = \frac{\|u_k(\psi;\mathcal{N})\|_2}{d_k},$$

using $\sum_{i\in A} g_{i,k}^\top \mu_k(A) = |A|\,\|\mu_k(A)\|_2^2$ in the numerator. Scoring the aggregated gradients $\sum_k \alpha_k g_{i,k}$ in the same way gives $S_{\text{agg}}^{\cos}(\psi;\mathcal{N}) = \|\sum_k \alpha_k u_k(\psi;\mathcal{N})\|_2/D$, where $D = (\sum_{i\in\mathcal{N}} \|\sum_k \alpha_k g_{i,k}\|_2^2)^{1/2}$. Writing $u_k = u_k(\psi;\mathcal{N})$ and expanding the square as in Proposition 1,

$$D^2\big(S_{\text{agg}}^{\cos}\big)^2 = \Big(\sum_{k=1}^K \alpha_k \|u_k\|_2\Big)^2 - \Delta_\psi, \qquad \Delta_\psi = 2\sum_{k<\ell} \alpha_k\alpha_\ell \big(\|u_k\|_2\|u_\ell\|_2 - u_k^\top u_\ell\big) \geq 0.$$

Angular disagreement reduces the squared aggregated numerator by Δ_ψ. As in the L2 case, premature aggregation can conceal objective-specific disagreement through cancellation. Scoring objectives separately preserves their split signals during tree construction. Both comparisons concern the score before the label-purity penalty, which we turn to next.

Split-RL penalizes splits that mix guidance labels. Let $H(\psi;\mathcal{N})$ be the normalized label impurity after ψ and $\rho(\mathcal{N})$ the node conflict coefficient, both defined in Section 4, and let $\bar{S}(\psi;\mathcal{N}) = \sum_k p_k(\mathcal{N}) S_k(\psi;\mathcal{N})$ combine the per-objective scores by label prevalence $p_k(\mathcal{N})$. The selected split maximizes

$$S_{\text{Split}}(\psi;\mathcal{N}) = \bar{S}(\psi;\mathcal{N})\big(1 - \text{clip}_{[0,1]}(\lambda_0 \rho(\mathcal{N}) H(\psi;\mathcal{N}))\big), \qquad \lambda_0 \geq 0.$$

Call ψ *label-pure* if $H(\psi;\mathcal{N}) = 0$ and *impure* otherwise. The following result gives the regularization strength at which a pure candidate outscores every impure one.

Proposition 2. *Let $\rho(\mathcal{N}) > 0$ and let ψ_{pure} satisfy $H(\psi_{\text{pure}};\mathcal{N}) = 0$ and $\bar{S}(\psi_{\text{pure}};\mathcal{N}) > 0$. Then ψ_{pure} scores at least as high as every impure candidate at $\mathcal{N}$ whenever*

$$\lambda_0 > \max_{\substack{\psi:\, H(\psi;\mathcal{N})>0,\\ \bar{S}(\psi;\mathcal{N})>\bar{S}(\psi_{\text{pure}};\mathcal{N})}} \frac{1 - \bar{S}(\psi_{\text{pure}};\mathcal{N})/\bar{S}(\psi;\mathcal{N})}{\rho(\mathcal{N}) H(\psi;\mathcal{N})}.$$

Proof. Write $m(\psi) = 1 - \text{clip}_{[0,1]}\big(\lambda_0 \rho(\mathcal{N}) H(\psi;\mathcal{N})\big) \in [0,1]$, so that

$$S_{\text{Split}}(\psi;\mathcal{N}) = \bar{S}(\psi;\mathcal{N})\, m(\psi).$$

Since $H(\psi_{\text{pure}};\mathcal{N}) = 0$ we have $m(\psi_{\text{pure}}) = 1$ and $S_{\text{Split}}(\psi_{\text{pure}};\mathcal{N}) = \bar{S}(\psi_{\text{pure}};\mathcal{N})$. Fix an impure candidate ψ. As $\bar{S}(\psi;\mathcal{N}) = \sum_k p_k(\mathcal{N}) S_k(\psi;\mathcal{N}) \geq 0$, if $\bar{S}(\psi;\mathcal{N}) \leq \bar{S}(\psi_{\text{pure}};\mathcal{N})$ then

$$S_{\text{Split}}(\psi;\mathcal{N}) = \bar{S}(\psi;\mathcal{N})\, m(\psi) \leq \bar{S}(\psi;\mathcal{N}) \leq \bar{S}(\psi_{\text{pure}};\mathcal{N}) = S_{\text{Split}}(\psi_{\text{pure}};\mathcal{N}).$$

Otherwise $\bar{S}(\psi;\mathcal{N}) > \bar{S}(\psi_{\text{pure}};\mathcal{N}) > 0$, and $S_{\text{Split}}(\psi_{\text{pure}};\mathcal{N}) > S_{\text{Split}}(\psi;\mathcal{N})$ holds exactly when

$$\text{clip}_{[0,1]}\big(\lambda_0 \rho(\mathcal{N}) H(\psi;\mathcal{N})\big) > 1 - \frac{\bar{S}(\psi_{\text{pure}};\mathcal{N})}{\bar{S}(\psi;\mathcal{N})}.$$

The right-hand side lies in $(0,1)$, so $\lambda_0 > \big(1 - \bar{S}(\psi_{\text{pure}};\mathcal{N})/\bar{S}(\psi;\mathcal{N})\big)/\big(\rho(\mathcal{N}) H(\psi;\mathcal{N})\big)$ suffices. Taking the maximum over such ψ gives the stated condition. □

Proposition 2 isolates the role of the regularization strength λ_0. If a pure split is already competitive in raw score, no penalty is needed. Otherwise, increasing λ_0 suppresses impure candidates whose raw score advantage is not large enough to overcome the conflict-weighted impurity penalty.

Once the topology is fixed, the leaf update combines the per-objective mean gradients by label prevalence,

$$v(\mathcal{L}) = \sum_{k=1}^K p_k(\mathcal{L})\, \mu_k(\mathcal{L}),$$

and $\mathcal{L}$ is *pure* for label j when $p_j(\mathcal{L}) = 1$. The following result bounds how far the applied update can fall from the update of the guided objective.

Proposition 3. *For any label j,*

$$\|v(\mathcal{L}) - \mu_j(\mathcal{L})\|_2 \leq \sum_{k \neq j} p_k(\mathcal{L}) \, \|\mu_k(\mathcal{L}) - \mu_j(\mathcal{L})\|_2,$$

so a leaf that is pure for j applies exactly $\mu_j(\mathcal{L})$.

Proof. Since $\sum_{k=1}^{K} p_k(\mathcal{L}) = 1$,

$$v(\mathcal{L}) - \mu_j(\mathcal{L}) = \sum_{k=1}^{K} p_k(\mathcal{L})\big(\mu_k(\mathcal{L}) - \mu_j(\mathcal{L})\big) = \sum_{k \neq j} p_k(\mathcal{L})\big(\mu_k(\mathcal{L}) - \mu_j(\mathcal{L})\big),$$

the term with $k = j$ being zero, and the triangle inequality gives the bound. If $\mathcal{L}$ is pure for j, then $p_k(\mathcal{L}) = 0$ for every $k \neq j$ and the right-hand side vanishes. □

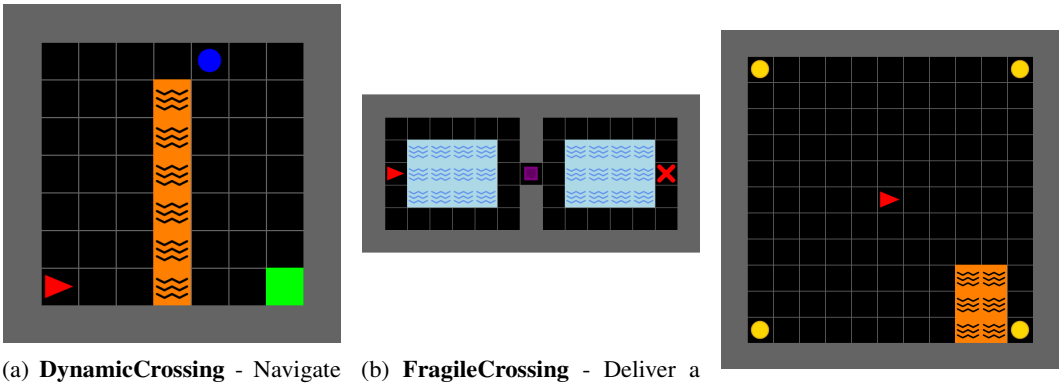
Proposition 3 connects the guidance labels to the actual update applied by the tree. As a leaf becomes purer, its aggregated update moves closer to the update of the guided objective context; in the pure case, the match is exact.

B Environment Details

In this section we describe the details of each environment used in the experiments.

B.1 MiniGrid Environment Details

The MiniGrid tasks are implemented within the MiniGrid framework [Chevalier-Boisvert et al., 2023] using agent-centric observations that provide a 7×7 grid field of view and discrete action spaces. Neural network baselines receive a flattened numeric encoding of the grid, while tree-based methods receive categorical observations where each cell is represented as a string token (e.g. `wall`, `lava_N`, `door_open`), enabling native categorical splits in the tree. Each environment defines a reward for task completion and a binary cost for constraint violations. The evaluation cost limit is $c \leq 0.1$ for all three environments (see Table 11). A visualization of the environments is shown in Figure 6.



(a) **DynamicCrossing** - Navigate around a lava river to reach the goal while avoiding a moving obstacle near the crossing.

(b) **FragileCrossing** - Deliver a heavy object to the drop zone; ice (light blue) is hazardous only while carrying.

(c) **Corner** - Collect four coins; the bottom-right corner is surrounded by lava with a single safe entrance.

Figure 6: **MiniGrid Environments.** Three custom environments with distinct safety-constraint structures: dynamic hazards, context-dependent constraints, and sparse localized hazards.

B.1.1 DynamicCrossing

The agent must reach a green goal square in the bottom-right corner of a room. A vertical river of lava (or walls) runs through the center of the grid with a single safe crossing point. Additionally, a moving obstacle (ball) traverses vertically near the crossing, requiring the agent to time its movements.

Action space. Discrete(4): turn left, turn right, move forward, wait.

Reward. The reward comprises potential-based shaping and a terminal bonus. Each step, $r = \phi(s') - \phi(s)$, where $\phi(s) = \text{clip}(1 - \|s - s_{\text{goal}}\|_2/d_{\max}, 0, 1)$. Upon reaching the goal, $r = 1 - 0.9 \times (\text{step_count}/\text{max_steps})$.

Cost. Binary: $c = 1$ if the agent attempts to move forward into any non-goal obstacle (lava, the moving ball, or a wall).

Label. $y(s) = \text{cost}$ when the cell directly in front of the agent is lava or the moving ball; $y(s) = \text{reward}$ otherwise.

Termination. The episode ends when the agent reaches the goal, collides with the moving obstacle, or exceeds the step limit. Stepping on lava incurs cost but does not terminate the episode.

B.1.2 FragileCrossing

Two rooms are separated by a central wall with a single doorway. The agent must pick up a heavy object in the doorway and deliver it to a drop zone (marked X) in the far room. Both rooms contain horizontal strips of ice that are safe when the agent is empty-handed, but hazardous when carrying the heavy object — exceeding a configurable number of steps on ice while carrying results in termination.

Action space. Discrete(5): turn left, turn right, move forward, pickup, drop.

Reward. Potential-based shaping from Manhattan distance along the mission path from agent to box to goal, plus a terminal bonus for delivery.

Cost. Binary: $c = 1$ when the agent steps on ice while carrying the heavy object.

Label. $y(s) = \text{cost}$ when the agent is carrying the object and is adjacent to or on ice; $y(s) = \text{reward}$ otherwise.

Termination. Successful delivery, exceeding the ice-step limit while carrying, or exceeding the total step limit.

B.1.3 Corner

The agent starts in the center of a 13×13 room and must collect four coins, one in each corner. Three corners are easily accessible, but the bottom-right corner is surrounded by lava with only a single narrow entrance from above.

Action space. Discrete(4): turn left, turn right, move forward, pickup.

Reward. $r = 0.25$ per coin collected; episode success when all four coins are gathered.

Cost. Binary: $c = 1$ when the agent is standing on a lava cell.

Label. $y(s) = \text{cost}$ when the agent is standing on or facing a lava cell; $y(s) = \text{reward}$ otherwise. The label is more conservative than the cost, firing before the agent steps onto lava.

Termination. All four coins collected or exceeding the step limit. Stepping on lava incurs cost but does not terminate the episode.

B.2 SUMO-RL Scenario Details

SUMO-RL [Alegre, 2019] is a traffic signal control framework built on the SUMO microsimulator [Lopez et al., 2018]. Control is decentralized; each signalized intersection is controlled by an independent agent that selects the next green phase from a discrete action set. Observations include

per-lane queue lengths, densities, and average speeds for all incoming and outgoing lanes. All intersections share a single policy (parameter sharing), so the policy must generalize across topological positions. We evaluate on three network topologies of increasing complexity (Figure 7): a single 4-way intersection, a symmetric 4×4 grid, and an asymmetric 4×4 arterial network.

All three SUMO-RL scenarios share the pressure reward $r_t = \sum_{l \in \mathcal{L}_{\text{out}}} n_l - \sum_{l \in \mathcal{L}_{\text{in}}} n_l$, where n_l is the number of vehicles currently on lane l (fewer vehicles queued on incoming lanes relative to outgoing lanes indicates better throughput), and differ only in cost, label, and injected vehicles. Evaluation cost limits are: Bus Priority $c \leq 7$, Convoy Priority $c \leq 6$, Premium Priority $c \leq 15$ (see Table 11). We denote the set of lanes served by the current green phase as $\mathcal{S}$, and q_j the queue fraction on lane j. Injection parameters are listed in Table 5.

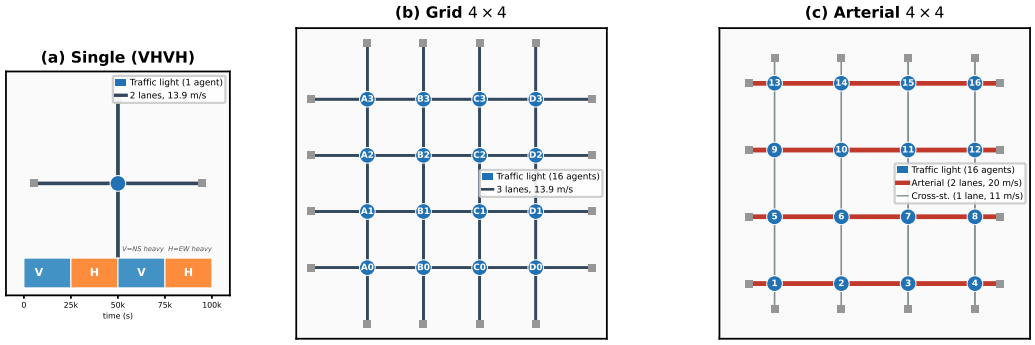


Figure 7: **SUMO Network Topologies.** Blue circles denote traffic-light intersections (agents); grey squares mark boundary entry/exit nodes. **(a)** Single 4-way intersection (1 agent, Discrete(4)) with VHVH non-stationary demand. **(b)** Grid 4×4: symmetric 16-intersection grid, uniform 3-lane roads, $\sim$1 500 vehicles, Discrete(8). **(c)** Arterial 4×4: asymmetric 16-intersection grid, EW arterial 2 lanes at 20 m/s vs. NS cross-streets 1 lane at 11 m/s, $\sim$2 500 vehicles, Discrete(5).

B.2.1 Bus Priority — Label and Cost

Observation extension. Three features per lane ($3 \times n_{\text{lanes}}$): binary bus presence $b_i \in \{0, 1\}$, bus count (normalized), and bus waiting time normalized by a threshold T_c.

Cost. Binary:

$$c_t = \begin{cases} 1 & \exists\, i : b_i > 0 \ \wedge \ \hat{w}_i^{\text{bus}} \geq 1, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where $\hat{w}_i^{\text{bus}} = w_i^{\text{bus}} / T_c$ is the normalized bus wait. Cost is phase-independent: it remains active while any bus is overdue regardless of whether its lane is currently served.

Label.

$$y(s) = \begin{cases} \text{cost} & \exists\, i : \ b_i > 0 \ \wedge \ \hat{w}_i^{\text{bus}} \geq T_w / T_c, \\ \text{reward} & \text{otherwise}. \end{cases} \tag{6}$$

The label fires when any bus has waited beyond the warning threshold, regardless of which lanes are currently served.

B.2.2 Convoy Must-Not-Split — Label and Cost

Observation extension. Three features per lane ($3 \times n_{\text{lanes}}$): binary convoy presence v_i, fraction of convoy vehicles observed at this intersection (convoy seen fraction $\hat{n}_i$), and convoy progress $p_i =$ (vehicles past stop line)/(total platoon).

Cost. Binary active-split detection:

$$c_t = \begin{cases} 1 & \exists\, i \notin \mathcal{S} : \ v_i > 0 \ \wedge \ 0 < p_i < 1, \\ 0 & \text{otherwise}. \end{cases} \tag{7}$$

Label. Presence-based, anticipatory:

$$y(s) = \begin{cases} \text{cost} & \exists\, i:\ v_i > 0\ \wedge\ \hat{n}_i > 0, \\ \text{reward} & \text{otherwise.} \end{cases} \tag{8}$$

The label fires as soon as any convoy vehicle is visible at the intersection ($\hat{n}_i > 0$), regardless of phase or progress. This is deliberately anticipatory: the label activates *before* cost (which requires $0 < p_i < 1$), providing the cost-sensitive update signal at the moment the phase-hold decision matters most.

The conflict in this scenario is rare and anticipatory. The label fires when a convoy becomes visible, before any cost has been incurred, and the window in which a violation can still be prevented spans only a few steps. A single multiplier shared across all states must therefore be either large enough to influence these rare states, which suppresses throughput across the ordinary traffic filling the rest of the episode, or small enough to preserve throughput, which leaves the cost gradients diluted by the majority.

B.2.3 Premium Priority — Label and Cost

Observation extension. Two features per lane ($2 \times n_{\text{lanes}}$): binary premium presence ρ_i and premium wait $\hat{w}_i^{\text{prem}}$ normalized by T_c. Premium wait is the *local* delay accumulated at this intersection, computed as route delay minus the delay snapshot at entry.

Cost. Binary:

$$c_t = \begin{cases} 1 & \exists\, i : \rho_i > 0\ \wedge\ \hat{w}_i^{\text{prem}} \geq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Label. Two-path decision frontier:

$$y(s) = \begin{cases} \text{cost} & \big(\exists\, i:\ \rho_i > 0\ \wedge\ \hat{w}_i^{\text{prem}} \geq T_w/T_c\big)\ \vee\ \big(\exists\, i \notin \mathcal{S}:\ \rho_i > 0\ \wedge\ \text{can_switch}(s)\big), \\ \text{reward} & \text{otherwise.} \end{cases} \tag{10}$$

The first condition fires when any premium vehicle's delay approaches the cost threshold (urgency). The second fires when a premium vehicle is on an unserved lane and the controller is past its minimum green time (actionable frontier).

Table 5: **SUMO scenario injection and threshold parameters.** Injection intervals are tuned per network so that each intersection sees ∼10–30% label rate under a random policy.

Parameter	Description	Bus	Convoy	Premium
Injection interval (s)	Single	90	300	60
	Grid 4×4	15	80	8
	Arterial 4×4	20	80	20
Vehicles per injection	Count per event	1	10–12	1
Vehicle length (m)	Injected vehicle size	12	5	5
Vehicle speed (m/s)	Max speed	11.1	13.9	13.9
Headway (s)	Spacing (convoy only)	—	2.0	—
T_c (cost threshold)	Normalization for wait	15 s	—	8 s
T_w (warn threshold)	Label urgency gate	6 s	—	3 s
Clean episode prob.	Prob. of no injection	0.5 (all scenarios)		

B.3 CityLearn Scenario Details

CityLearn [Nweye et al., 2024] is a building energy management benchmark where a central agent controls battery storage for a district of three buildings over a one-month episode (720 hourly steps).

At each step the agent observes electricity prices, solar generation, building loads, and battery state-of-charge (SOC), and decides how much to charge or discharge each battery (continuous action space). The reward is negative electricity cost; the constraint enforces a minimum reserve level whose semantics differ per scenario, creating a temporal conflict: the agent must trade off short-term cost savings against maintaining sufficient reserves for upcoming high-demand or low-generation periods.

All three CityLearn scenarios share the same reward $r_t = -\frac{1}{N}\sum_i \text{consumption}_i(t) \times \text{price}_i(t)$ and differ only in cost, label, and observation augmentation. The evaluation cost limit is $c \leq 5$ for all CityLearn scenarios (see Table 11). Each scenario appends one scalar feature to the observation. All normalized countdown features use $T_{\text{ep}} = 720$, the episode length. Event detection and thresholds are computed once from the exogenous schedules at environment initialization. Parameters are listed in Table 6. We denote mean battery SOC as $\bar{s}$, district import as $I_t = \max(\sum_i \text{NEC}_i(t),\, 0)$, and per-timestep max electricity price as π_t.

B.3.1 Arbitrage vs. Buffer — Label and Cost

Event detection. Peak-price periods: contiguous runs where π_t exceeds the $Q(0.92)$ percentile. A high-price sub-threshold π_{high} is the $Q(0.75)$ of *peak-hour* prices (not global).

Obs extension. Normalized steps until next peak (steps_until$/T_{\text{ep}}$).

Cost.

$$c_t = \min\Big(1,\ \frac{\max(s_{\text{safe}} - \bar{s},\, 0)}{s_{\text{safe}}}\Big), \qquad s_{\text{safe}} = 0.30. \tag{11}$$

Label.

$$y(s) = \begin{cases} \text{cost} & t \in \mathcal{P}_{\text{peak}} \ \wedge\ \big(\pi_t > \pi_{\text{high}} \ \vee\ \bar{s} < s_{\text{emerg}}\big), \\ \text{reward} & \text{otherwise}, \end{cases} \qquad s_{\text{emerg}} = 0.20. \tag{12}$$

B.3.2 Demand Response — Label and Cost

Event detection. Deterministic grid events at fixed timesteps (hours 168, 360, 528), each lasting 12 hours.

Obs extension. Normalized steps until next event.

Cost.

$$c_t = \begin{cases} \min\Big(1,\ \frac{\max(s_{\text{tgt}} - \bar{s},\, 0)}{s_{\text{tgt}}}\Big) & t \in \mathcal{P}_{\text{event}}, \\ 0 & \text{otherwise}, \end{cases} \qquad s_{\text{tgt}} = 0.80. \tag{13}$$

Label.

$$y(s) = \begin{cases} \text{cost} & t \in \mathcal{P}_{\text{event}}, \\ \text{reward} & \text{otherwise}. \end{cases} \tag{14}$$

The label is purely event-window membership: it fires throughout each 12-hour grid event, giving cost-sensitive updates control during the entire reserve-critical period.

B.3.3 Solar Ramp Reserve — Label and Cost

Event detection. Sunset transitions detected from the solar schedule: hour $t+1$ where $\text{solar}(t) > 0$ and $\text{solar}(t+1) = 0$. Pre-sunset prep: 2 hours before each sunset. Post-sunset buffer: 4 hours after each sunset.

Obs extension. Normalized steps until next buffer event.

Cost.

$$c_t = \begin{cases} \min\Big(1,\ \frac{\max(s_{\text{res}} - \bar{s},\, 0)}{s_{\text{res}}}\Big) & t \in \mathcal{P}_{\text{buffer}}, \\ 0 & \text{otherwise}, \end{cases} \qquad s_{\text{res}} = 0.50. \tag{15}$$

Label.

$$y(s) = \begin{cases} \text{cost} & t \in \mathcal{P}_{\text{prep}} \ \vee \ t \in \mathcal{P}_{\text{buffer}}, \\ \text{reward} & \text{otherwise.} \end{cases} \tag{16}$$

The label fires during both the 2-hour pre-sunset preparation window and the 4-hour post-sunset buffer, regardless of current electricity prices or battery state-of-charge (SOC). This gives cost-sensitive updates control over dispatch decisions throughout the entire sunset transition.

Table 6: **CityLearn scenario parameters.** All scenarios use the CityLearn Challenge 2023 Phase 2 dataset (3 buildings, 720 hourly steps per episode).

Parameter	Arb. vs. Buffer	Demand Response	Solar Ramp
SOC safety / target / reserve	0.30	0.80	0.50
Emergency SOC	0.20	0.40	0.25
Event quantile	Q_{92} (price)	fixed hours	solar schedule
High price quantile π_{high}	Q_{75} (peak-local)	—	—
Prep / buffer window (h)	—	— / 12	2 / 4

B.4 Atari RAM Dataset Details

Environment. We use the Arcade Learning Environment [Bellemare et al., 2013] with RAM observations, which are 128-dimensional byte vectors cast to float32. All games use the `NoFrameskip-v4` variant with sticky actions disabled, matching the standard Atari evaluation protocol. The eleven games evaluated are Qbert, Seaquest, Pong, SpaceInvaders, MsPacman, Frostbite, Alien, Amidar, BankHeist, KungFuMaster, and Gravitar.

Expert policies. Expert demonstrations were collected by rolling out pre-trained pixel-based agents while simultaneously recording the corresponding 128-byte RAM vectors. All expert models were downloaded from the Hugging Face Hub:

- **Qbert, Seaquest, Pong, SpaceInvaders** — PPO (CnnPolicy) from `ThomasSimonini/ppo-{Game}NoFrameskip-v4`,[2] trained with Stable-Baselines3 [Raffin et al., 2021].
- **MsPacman, Frostbite, Alien, Amidar, BankHeist, KungFuMaster, Gravitar** — PPO with IMPALA-Net from `cleanrl/{Game}-v5-ppo_atari_envpool_async_jax_scan_impalanet_machado-seed1`,[3] trained with CleanRL [Huang et al., 2022].

All models are publicly available on the Hugging Face Hub without access restrictions. Stable-Baselines3 is released under the MIT License; CleanRL is released under the MIT License.

Non-expert data. Non-expert episodes are collected by sampling actions uniformly at random from the environment's action space in the same RAM-observation setting.

Mixed datasets. For each game and dataset size N, we construct a mixed dataset with a 1:1 ratio of expert and non-expert transitions. Each episode carries a guidance label indicating its provenance, $y(s) = \text{expert}$ for expert episodes and $y(s) = \text{improve}$ for non-expert episodes. Split-RL uses these labels to route gradient updates; baseline methods ignore them. Six dataset sizes are evaluated: 10K, 20K, 50K, 100K, 200K, and Full, where Full uses the complete available mixed dataset. We report normalized values in the main paper calculated as:

$$R_{\text{norm}} = \frac{R_\pi - R_{\text{non-expert}}}{R_{\text{expert}} - R_{\text{non-expert}}}.$$

[2] `https://huggingface.co/ThomasSimonini`

[3] `https://huggingface.co/cleanrl`

C Training Hyperparameters

All constrained RL experiments (MiniGrid, SUMO-RL, and CityLearn) are trained for 1M environment steps. Hyperparameters were selected via Bayesian search on a single seed and then held fixed across all evaluation seeds and scenarios within each domain. Split-RL and GBT baselines use oblivious trees and the cosine similarity split-score criterion throughout. These were selected as part of the hyperparameter search and we did not run dedicated experiments comparing which tree structure or split criterion is superior. Neural network baselines use two-layer MLPs (64–64) with tanh activations unless otherwise noted. We use 15 seeds per configuration for all constrained RL domains (MiniGrid, SUMO-RL, and CityLearn), including ablations. During training, episodic reward and cost are averaged over the last 100 episodes. Tables use the final averages from each seed and report their mean and sample standard deviation. The 95% confidence intervals use percentile bootstrapping across seeds with 2,000 resamples. Cost limits are listed in Table 11. Each limit is fixed before training and used as CPO's cost constraint, CUP's projection target, IPO's barrier, and the Lagrangian penalty threshold. The offline-RL evaluation protocol is described below alongside Table 10.

Constrained-RL implementation. All constrained-RL methods use the same environment wrappers, rollout collection, evaluation code, and seed protocol. Neural CPO, CUP, IPO, and PPO-Lagrangian are implemented in a shared Stable-Baselines3 [Raffin et al., 2021] codebase, following the corresponding OmniSafe [Ji et al., 2023] implementations. Split-RL and PPO-Lagrangian (GBT) use the same training interface as GBRL tree policies.

C.1 MiniGrid

Table 7 lists the hyperparameters for all MiniGrid methods. All methods share $\gamma = 0.99$ and $\lambda_{\text{GAE}} = 0.95$.

Table 7: **MiniGrid hyperparameters.** All methods share γ=0.99 and λ_{GAE}=0.95. NN baselines normalize observations and rewards; GBT methods use no normalization. "—" = not applicable.

Parameter	CPO	CUP	IPO	PPOLag	PPOLag (GBT)	Split-RL
Num. envs	1	16	8	16	16	16
n_{steps}	4096	1024	256	256	256	256
Batch size	256	128	256	128	512	512
Epochs	30	10	10	20	20	20
Learning rate	4.3×10^{-4}	7.4×10^{-5}	7.6×10^{-4}	2.1×10^{-4}	—	—
Policy LR	—	—	—	—	0.17	0.17
Value LR	—	—	—	—	0.01	0.01
Entropy coef.	—	—	0.02	—	0.0	0.0
Clip range	—	—	—	0.25	0.2	0.2
Tree depth	—	—	—	—	4	4
λ_0	—	—	—	—	—	1.0

C.2 SUMO-RL

Table 8 lists the hyperparameters for the SUMO-RL traffic signal control experiments. Training uses 1M environment steps per network–scenario pair with a conflict probability of 0.5 (special vehicles are injected in 50% of episodes; the remaining episodes are clean).

C.3 CityLearn

Table 9 lists the hyperparameters for the CityLearn building energy management experiments. Training uses 1M environment steps per scenario. Actions are continuous (battery charge/discharge); Split-RL and GBRL use a fixed log-std initialization.

Table 8: **SUMO-RL hyperparameters.** All methods use reward normalization. GBRL methods use oblivious trees.

Parameter	CPO	CUP	IPO	PPOLag	PPOLag (GBT)	Split-RL
n_{steps}	256	256	512	256	512	512
Batch size	128	256	256	256	128	128
Epochs	10	20	10	20	8	8
γ	0.95	0.95	0.90	0.95	0.90	0.90
λ_{GAE}	0.98	0.95	0.90	0.98	0.95	0.95
Clip range	0.2	0.2	0.3	0.1	0.2	0.2
Entropy coef.	0.01	4.2×10^{-3}	2.2×10^{-5}	1.1×10^{-5}	3.1×10^{-5}	3.1×10^{-5}
Learning rate	2.6×10^{-3}	7.7×10^{-5}	1.1×10^{-4}	2.3×10^{-4}	—	—
Policy LR	—	—	—	—	0.060	0.060
Value LR	—	—	—	—	0.036	0.036
Cost LR	—	—	—	—	0.008	0.008
Max grad norm	0.5	1.19	1.24	0.88	1.0	1.0
Target KL	0.01	—	—	—	0.02	0.02
Tree depth	—	—	—	—	5	5
λ_0	—	—	—	—	—	1.0

Table 9: **CityLearn hyperparameters.** All GBRL methods use oblivious depth-4 trees. NN baselines use reward normalization; GBT methods use no normalization.

Parameter	CPO	CUP	IPO	PPOLag	PPOLag (GBT)	Split-RL
n_{steps}	512	256	512	256	256	256
Batch size	128	256	64	64	64	128
Epochs	10	20	20	4	10	20
γ	0.90	0.95	0.90	0.90	0.95	0.95
λ_{GAE}	0.90	0.95	0.95	0.95	0.95	0.95
Clip range	0.2	0.2	0.2	0.3	0.2	0.2
Entropy coef.	0.01	4.0×10^{-3}	4.2×10^{-3}	2.5×10^{-5}	5.9×10^{-4}	5.9×10^{-4}
Learning rate	5.3×10^{-5}	4.3×10^{-5}	2.8×10^{-5}	1.9×10^{-3}	—	—
Policy LR	—	—	—	—	4.7×10^{-5}	4.7×10^{-5}
Value LR	—	—	—	—	4.3×10^{-4}	4.3×10^{-4}
Cost LR	—	—	—	—	—	2.0×10^{-3}
Target KL	0.02	0.02	—	—	0.05	0.05
Fixed std	—	—	—	—	yes	yes
Log-std init	—	—	—	—	-3	-3
Tree depth	—	—	—	—	4	4
λ_0	—	—	—	—	—	1.0

C.4 Atari RAM (Offline)

Table 10 lists the hyperparameters for the offline Atari RAM experiments. Training budgets differ by method: CQL and BCQ train for 500K gradient steps, Decision Transformer for 250K steps, and AWR-based methods (NN and GBRL) including Split-RL train for 50K steps. All methods use the same fixed mixed dataset per game (expert + random transitions at each reported dataset size) and each method is trained once per game and dataset size. For evaluation, we use deterministic action selection on five environment seeds with 10 episodes each, for a total of 50 evaluation episodes.

Offline-RL implementation. CQL, BCQ, and Decision Transformer use d3rlpy implementations [Seno and Imai, 2022]. AWR and AWR+BC are implemented in our codebase for both neural and GBT policies. To keep the BC and AWR gradients on comparable scales, Split-RL divides each objective's gradients by their standard deviation before aggregation.

Table 10: **Atari RAM offline hyperparameters.** The GBT baselines use the shared hyperparameter values selected for Split-RL. The main difference is whether guidance labels (Split-RL) or BC regularization (AWR+BC) is active.

Parameter	CQL	DT	AWR (NN)	AWR+BC (NN)	BCQ	AWR (GBRL)	AWR+BC (GBRL)	Split-RL
Training steps	500K	250K	50K	50K	500K	50K	50K	50K
Batch size	128	128	256	256	128	1024	1024	1024
γ	0.99	—	1.0	1.0	0.99	1.0	1.0	1.0
β (AWR temp)	—	—	1.0	1.0	—	1.0	1.0	1.0
Learning rate	2.6×10^{-4}	2.6×10^{-4}	1.4×10^{-4}	1.4×10^{-4}	2.1×10^{-4}	—	—	—
Policy LR	—	—	—	—	—	0.025	0.025	0.025
Value LR	—	—	—	—	—	0.005	0.005	0.005
Entropy coef.	—	—	1.5×10^{-4}	1.5×10^{-4}	—	0.01	0.01	0.01
CQL α	4.14	—	—	—	—	—	—	—
BC coef.	—	—	—	1.0	—	—	1.0	—
Context length	—	30	—	—	—	—	—	—
Tree depth	—	—	—	—	—	5	5	5
Eval. seeds	5	5	5	5	5	5	5	5
λ_0	—	—	—	—	—	—	—	1.0

D Detailed Results

This appendix reports the raw values underlying the main summaries. For constrained RL, bold entries mark the feasible method with the highest reward in each scenario. For Atari RAM, we report raw returns per game and dataset size; the main text reports expert-normalized averages computed from these values.

Feasibility is determined using the cost limits in Table 11. The cost limit is part of the problem specification rather than a post-hoc threshold: a single limit per scenario is set before training, and every constrained baseline receives it and optimizes against it, entering CPO's cost constraint, CUP's projection, IPO's barrier, and the Lagrangian penalty. Each method therefore targets exactly the limit it is scored against. Most mean-cost violations are large rather than marginal, while exceptions include PPO-Lagrangian on SUMO-RL Single/Premium (15.06 against 15) and CUP on CityLearn Arbitrage vs. Buffer (5.06 against 5).

Table 11: **Evaluation cost limits.** A method is *feasible* if its mean episodic cost does not exceed the limit for its scenario. Bold entries in Tables 12–14 mark the feasible method with highest reward.

Domain	Scenario	Cost limit
MiniGrid	DynamicCrossing	0.1
MiniGrid	FragileCrossing	0.1
MiniGrid	Corner	0.1
SUMO-RL	Bus Priority	7.0
SUMO-RL	Convoy Priority	6.0
SUMO-RL	Premium Priority	15.0
CityLearn	Arb. vs Buffer	5.0
CityLearn	Demand Response	5.0
CityLearn	Solar Ramp Reserve	5.0

Atari RAM. Table 15 reports the raw evaluation return for every game and dataset size, together with the expert and non-expert reference returns used for normalization. The expert-normalized averages in Table 3 are computed from these values.

Table 16 gives the corresponding per-game returns for the BC target ablation summarized in Table 4, at the full dataset size.

E Interpretability: Routing Density Under Policy Rollouts

What is measured. The guidance label is used only during training, to score candidate splits; at evaluation it is never an input. To ask whether the ensemble has internalized the conflict structure, we read it back out of the leaves. At tree construction each leaf records the fraction of its training

Table 12: **MiniGrid full results.** Mean ± std, with 95% bootstrap CI, over seeds. Episodic reward (↑) and cost (↓). **Bold** marks the feasible operating point with highest reward; underline marks the feasible operating point with the second-highest reward, using the limits in Table 11.

		Neural Network				GBT	
Scenario	**Metric**	**CPO**	**CUP**	**IPO**	**PPOLag**	**PPOLag**	**Split-RL**
DynamicCrossing	Rew.	0.85 ± 0.52 $[0.59, 1.11]$	0.12 ± 0.13 $[0.05, 0.18]$	0.91 ± 0.69 $[0.57, 1.23]$	0.53 ± 0.51 $[0.29, 0.79]$	$\underline{1.48 \pm 0.13}$ $\underline{[1.40, 1.53]}$	$\mathbf{1.53 \pm 0.01}$ $[\mathbf{1.52}, \mathbf{1.53}]$
	Cost	0.34 ± 0.84 $[0.07, 0.80]$	5.98 ± 22.74 $[0.05, 17.76]$	2.72 ± 7.51 $[0.40, 6.85]$	23.38 ± 61.86 $[2.55, 55.79]$	$\underline{0.01 \pm 0.01}$ $\underline{[0.00, 0.01]}$	$\mathbf{0.00 \pm 0.00}$ $[\mathbf{0.00}, \mathbf{0.00}]$
FragileCrossing	Rew.	0.48 ± 0.30 $[0.34, 0.62]$	0.13 ± 0.20 $[0.03, 0.23]$	0.47 ± 0.37 $[0.29, 0.65]$	0.36 ± 0.37 $[0.17, 0.55]$	$\underline{0.93 \pm 0.21}$ $\underline{[0.82, 1.03]}$	$\mathbf{1.07 \pm 0.11}$ $[\mathbf{1.01}, \mathbf{1.10}]$
	Cost	0.34 ± 0.61 $[0.09, 0.66]$	0.00 ± 0.00 $[0.00, 0.00]$	0.24 ± 0.21 $[0.14, 0.34]$	0.02 ± 0.04 $[0.01, 0.04]$	$\underline{0.04 \pm 0.04}$ $\underline{[0.02, 0.06]}$	$\mathbf{0.01 \pm 0.01}$ $[\mathbf{0.00}, \mathbf{0.01}]$
Corner	Rew.	0.93 ± 0.11 $[0.87, 0.98]$	0.00 ± 0.00 $[0.00, 0.00]$	0.51 ± 0.48 $[0.27, 0.76]$	0.48 ± 0.47 $[0.24, 0.72]$	$\mathbf{0.97 \pm 0.09}$ $[\mathbf{0.92}, \mathbf{1.00}]$	$\underline{0.94 \pm 0.07}$ $\underline{[0.91, 0.97]}$
	Cost	0.22 ± 0.50 $[0.01, 0.49]$	0.02 ± 0.07 $[0.00, 0.06]$	0.69 ± 0.84 $[0.36, 1.13]$	0.23 ± 0.68 $[0.03, 0.59]$	$\mathbf{0.03 \pm 0.12}$ $[\mathbf{0.00}, \mathbf{0.09}]$	$\underline{0.00 \pm 0.01}$ $\underline{[0.00, 0.01]}$

Table 13: **SUMO-RL full results.** Mean ± std, with 95% bootstrap CI, over seeds. Episodic reward (↑) and cost (↓), evaluated with conflict probability 0.5 (i.e. special vehicles are injected in 50% of episodes). Reward is in units of 10^3 throughout; cost is unscaled. **Bold** marks the feasible operating point with highest reward; underline marks the feasible operating point with the second-highest reward, ties included, using the limits in Table 11.

		Neural Network				GBT	
Scenario	**Metric**	**CPO**	**CUP**	**IPO**	**PPOLag**	**PPOLag**	**Split-RL**
Arterial / Bus	Rew.	-1.407 ± 0.324 $[-1.570, -1.267]$	-1.332 ± 0.228 $[-1.447, -1.229]$	$\underline{-1.267 \pm 0.231}$ $\underline{[-1.389, -1.164]}$	-1.299 ± 0.256 $[-1.430, -1.185]$	-1.210 ± 0.137 $[-1.278, -1.147]$	$\mathbf{-1.189 \pm 0.181}$ $[\mathbf{-1.279}, \mathbf{-1.107}]$
	Cost	3.44 ± 5.85 $[1.33, 6.73]$	3.44 ± 3.15 $[2.12, 5.14]$	$\underline{1.85 \pm 1.04}$ $\underline{[1.38, 2.37]}$	6.60 ± 5.01 $[4.21, 9.15]$	20.96 ± 10.01 $[16.18, 26.02]$	$\mathbf{3.92 \pm 2.46}$ $[\mathbf{2.73}, \mathbf{5.11}]$
Arterial / Convoy	Rew.	-1.429 ± 0.273 $[-1.571, -1.303]$	-1.470 ± 0.220 $[-1.589, -1.369]$	$\underline{-1.385 \pm 0.258}$ $\underline{[-1.511, -1.271]}$	-1.232 ± 0.250 $[-1.362, -1.118]$	-1.229 ± 0.149 $[-1.303, -1.159]$	$\mathbf{-1.296 \pm 0.147}$ $[\mathbf{-1.377}, \mathbf{-1.230}]$
	Cost	2.79 ± 1.73 $[1.98, 3.72]$	1.88 ± 0.85 $[1.47, 2.29]$	$\underline{2.42 \pm 2.65}$ $\underline{[1.30, 3.88]}$	12.35 ± 6.26 $[9.58, 15.60]$	12.58 ± 3.12 $[11.17, 14.21]$	$\mathbf{2.16 \pm 0.86}$ $[\mathbf{1.80}, \mathbf{2.64}]$
Arterial / Premium	Rew.	-1.378 ± 0.315 $[-1.530, -1.231]$	$\underline{-1.366 \pm 0.242}$ $\underline{[-1.495, -1.268]}$	-1.394 ± 0.250 $[-1.514, -1.275]$	-1.170 ± 0.150 $[-1.243, -1.099]$	-1.259 ± 0.147 $[-1.335, -1.192]$	$\mathbf{-1.254 \pm 0.140}$ $[\mathbf{-1.322}, \mathbf{-1.185}]$
	Cost	7.18 ± 3.23 $[5.69, 8.84]$	$\underline{5.95 \pm 3.53}$ $\underline{[4.38, 7.81]}$	6.43 ± 2.49 $[5.24, 7.61]$	16.22 ± 7.27 $[13.37, 20.47]$	37.41 ± 10.62 $[32.30, 42.31]$	$\mathbf{13.91 \pm 6.96}$ $[\mathbf{10.77}, \mathbf{17.92}]$
Grid / Bus	Rew.	-0.216 ± 0.028 $[-0.230, -0.203]$	-0.156 ± 0.025 $[-0.169, -0.145]$	-0.163 ± 0.024 $[-0.176, -0.152]$	$\mathbf{-0.134 \pm 0.025}$ $[\mathbf{-0.147}, \mathbf{-0.123}]$	-0.153 ± 0.026 $[-0.167, -0.142]$	$\underline{-0.145 \pm 0.024}$ $\underline{[-0.158, -0.134]}$
	Cost	0.54 ± 0.43 $[0.35, 0.76]$	0.05 ± 0.06 $[0.02, 0.08]$	0.02 ± 0.02 $[0.01, 0.03]$	$\mathbf{0.09 \pm 0.07}$ $[\mathbf{0.06}, \mathbf{0.13}]$	0.31 ± 0.11 $[0.26, 0.36]$	$\underline{0.05 \pm 0.03}$ $\underline{[0.04, 0.07]}$
Grid / Convoy	Rew.	-0.267 ± 0.024 $[-0.279, -0.255]$	$\underline{-0.181 \pm 0.024}$ $\underline{[-0.193, -0.170]}$	-0.191 ± 0.023 $[-0.202, -0.180]$	-0.157 ± 0.025 $[-0.170, -0.146]$	-0.178 ± 0.029 $[-0.193, -0.165]$	$\mathbf{-0.171 \pm 0.027}$ $[\mathbf{-0.185}, \mathbf{-0.158}]$
	Cost	2.34 ± 0.69 $[1.99, 2.65]$	$\underline{1.35 \pm 0.42}$ $\underline{[1.15, 1.56]}$	3.39 ± 0.98 $[2.93, 3.89]$	7.94 ± 2.37 $[6.79, 9.14]$	8.95 ± 1.97 $[7.97, 9.87]$	$\mathbf{2.41 \pm 0.50}$ $[\mathbf{2.17}, \mathbf{2.65}]$
Grid / Premium	Rew.	-0.225 ± 0.027 $[-0.238, -0.212]$	-0.162 ± 0.025 $[-0.174, -0.150]$	-0.169 ± 0.025 $[-0.182, -0.158]$	$\mathbf{-0.138 \pm 0.025}$ $[\mathbf{-0.151}, \mathbf{-0.127}]$	$\underline{-0.153 \pm 0.022}$ $\underline{[-0.164, -0.143]}$	$\underline{-0.153 \pm 0.030}$ $\underline{[-0.169, -0.140]}$
	Cost	5.09 ± 2.28 $[3.99, 6.21]$	2.01 ± 0.63 $[1.68, 2.32]$	2.80 ± 0.80 $[2.40, 3.18]$	$\mathbf{3.02 \pm 1.18}$ $[\mathbf{2.44}, \mathbf{3.59}]$	$\underline{4.22 \pm 1.44}$ $\underline{[3.51, 4.86]}$	$\underline{1.84 \pm 0.43}$ $\underline{[1.62, 2.04]}$
Single / Bus	Rew.	-8.767 ± 0.187 $[-8.857, -8.674]$	-8.839 ± 0.626 $[-9.183, -8.610]$	-8.746 ± 0.182 $[-8.841, -8.663]$	$\mathbf{-8.421 \pm 0.051}$ $[\mathbf{-8.445}, \mathbf{-8.395}]$	-8.519 ± 0.082 $[-8.558, -8.478]$	$\underline{-8.433 \pm 0.125}$ $\underline{[-8.494, -8.377]}$
	Cost	2.31 ± 1.15 $[1.81, 2.94]$	2.96 ± 1.09 $[2.37, 3.47]$	3.76 ± 0.82 $[3.38, 4.19]$	$\mathbf{2.98 \pm 0.64}$ $[\mathbf{2.67}, \mathbf{3.27}]$	4.06 ± 0.82 $[3.65, 4.45]$	$\underline{0.87 \pm 0.14}$ $\underline{[0.79, 0.93]}$
Single / Convoy	Rew.	-75.609 ± 12.567 $[-80.388, -68.190]$	-33.296 ± 8.846 $[-38.061, -29.716]$	-14.121 ± 0.720 $[-14.494, -13.774]$	-17.406 ± 3.642 $[-19.187, -15.673]$	-14.949 ± 1.292 $[-15.578, -14.315]$	$\mathbf{-26.971 \pm 1.655}$ $[\mathbf{-27.818}, \mathbf{-26.196}]$
	Cost	19.81 ± 9.91 $[14.48, 24.48]$	13.81 ± 7.58 $[10.35, 17.54]$	25.95 ± 4.56 $[23.80, 28.08]$	23.51 ± 7.79 $[19.67, 27.26]$	78.83 ± 13.27 $[72.52, 85.53]$	$\mathbf{5.66 \pm 2.50}$ $[\mathbf{4.46}, \mathbf{6.96}]$
Single / Premium	Rew.	-23.398 ± 25.250 $[-36.364, -11.951]$	-9.093 ± 0.283 $[-9.232, -8.954]$	$\underline{-8.955 \pm 0.195}$ $\underline{[-9.049, -8.863]}$	-8.601 ± 0.160 $[-8.691, -8.536]$	-8.541 ± 0.072 $[-8.576, -8.508]$	$\mathbf{-8.679 \pm 0.183}$ $[\mathbf{-8.775}, \mathbf{-8.598}]$
	Cost	38.26 ± 84.51 $[6.37, 87.90]$	8.63 ± 3.81 $[6.80, 10.63]$	$\underline{13.70 \pm 1.72}$ $\underline{[12.90, 14.58]}$	15.06 ± 1.75 $[14.22, 15.92]$	24.84 ± 2.14 $[23.77, 25.86]$	$\mathbf{6.95 \pm 1.09}$ $[\mathbf{6.41}, \mathbf{7.47}]$

samples that carried the cost label. We pass a visited state through the ensemble, read that fraction from the leaf it reaches in each tree, and average over trees. We call the result the *routing density* of

Table 14: **CityLearn full results.** Mean ± std, with 95% bootstrap CI, over seeds. Episodic reward (↑) and cost (↓). **Bold** marks the feasible operating point with highest reward; no method besides Split-RL is feasible in any CityLearn scenario, so no entry is underlined here (using the limits in Table 11).

		Neural Network				GBT	
Scenario	**Metric**	**CPO**	**CUP**	**IPO**	**PPOLag**	**PPOLag**	**Split-RL**
Arb. vs Buffer	Rew.	-7.96 ± 0.41 $[-8.16, -7.78]$	-8.48 ± 0.14 $[-8.54, -8.41]$	-8.62 ± 0.05 $[-8.64, -8.59]$	-8.13 ± 0.28 $[-8.25, -7.97]$	-8.81 ± 0.10 $[-8.86, -8.76]$	$\mathbf{-8.90 \pm 0.13}$ $[\mathbf{-8.97}, \mathbf{-8.84}]$
	Cost	5.66 ± 3.77 $[3.99, 7.62]$	5.06 ± 11.55 $[1.02, 11.54]$	8.90 ± 8.48 $[4.92, 13.10]$	15.02 ± 9.36 $[11.34, 20.09]$	9.97 ± 2.53 $[8.63, 11.17]$	$\mathbf{2.32 \pm 1.47}$ $[\mathbf{1.67}, \mathbf{3.11}]$
Demand Response	Rew.	-7.94 ± 0.59 $[-8.24, -7.66]$	-8.07 ± 0.57 $[-8.33, -7.78]$	-8.49 ± 0.16 $[-8.56, -8.41]$	-8.47 ± 0.09 $[-8.50, -8.42]$	-8.90 ± 0.10 $[-8.95, -8.85]$	$\mathbf{-8.83 \pm 0.05}$ $[\mathbf{-8.85}, \mathbf{-8.81}]$
	Cost	8.06 ± 3.84 $[6.16, 9.94]$	18.30 ± 10.78 $[12.63, 23.42]$	6.25 ± 2.54 $[5.10, 7.55]$	28.34 ± 0.12 $[28.28, 28.40]$	10.46 ± 2.92 $[9.06, 11.90]$	$\mathbf{4.64 \pm 0.72}$ $[\mathbf{4.33}, \mathbf{5.00}]$
Solar Ramp	Rew.	-7.84 ± 0.51 $[-8.11, -7.59]$	-8.13 ± 0.49 $[-8.37, -7.89]$	-8.55 ± 0.37 $[-8.72, -8.37]$	-8.53 ± 0.09 $[-8.57, -8.48]$	-8.94 ± 0.13 $[-9.00, -8.87]$	$\mathbf{-9.11 \pm 0.11}$ $[\mathbf{-9.17}, \mathbf{-9.05}]$
	Cost	7.33 ± 3.82 $[5.43, 9.16]$	6.52 ± 3.92 $[4.41, 8.33]$	7.32 ± 3.62 $[5.44, 9.01]$	43.57 ± 24.01 $[31.78, 55.64]$	10.84 ± 2.57 $[9.56, 11.99]$	$\mathbf{3.10 \pm 0.78}$ $[\mathbf{2.72}, \mathbf{3.50}]$

that state: the share of cost-labelled training data sharing a region with it. A model that routes by position alone would assign cost- and reward-labelled states at the same location the same density; a model that routes by conflict structure separates them.

Protocol. We run 60 evaluation episodes per environment with actions sampled from the trained policy and record every visited state. Figures 5 and 8 aggregate these visits per cell: panel (a) shows the environment, panel (b) the fraction of visits to that cell at which the ground-truth guidance label fired, and panel (c) the mean routing density of the leaves those same visits reached. Both panels describe the same set of events and are therefore directly comparable. Cells that were never visited are left grey. The recorded rollouts cover 8632 steps over 95 cells in Corner, 686 steps over 11 cells in DynamicCrossing, and 5329 steps over 16 cells in FragileCrossing.

Result. In FragileCrossing, panels (b) and (c) coincide: the cells where the cost label fires during rollouts are the cells whose visits reach cost-heavy leaves. DynamicCrossing shows the same pattern. No label is supplied at evaluation, so the ensemble recovers the conflict structure from the observation alone. The partition is therefore readable after training: high routing density indicates a higher proportion of cost-labelled samples.

Corner. Corner's guidance label is a property of the cell and heading rather than of the trajectory, and its trained policy avoids the lava entirely, so cost-labelled states are almost never visited under rollouts: only 31 of its 468 reachable state configurations carry the cost label, and the policy reaches few of them. Its panels are therefore close to uniform here. We show it under rollouts anyway, for consistency with the other two environments; enumerating all reachable states instead of sampling trajectories recovers the separation.

Scope. This probe reports what the leaves contain, not which features the splits use or how much each one contributes. It shows that routing correlates with the conflict structure under the visited state distribution; it does not attribute that behaviour to particular split variables, and it does not measure how much of the chosen action is explained by the routed region. Split-level attribution and feature importance are natural extensions that we leave to future work.

F Ablations

F.1 Label-Purity Regularization

In Figure 3 we ablate λ_0 in CityLearn Solar Ramp Reserve. Removing the purity penalty weakens cost control, while nonzero λ_0 lowers cost by discouraging label-mixing splits in conflict windows.

Table 15: **Atari RAM — Full Results (All Dataset Sizes).** At the final training step, we average raw return over 10 episodes per evaluation seed and report the mean ± std of the five seed averages. Size column shows total mixed transitions from a 1:1 expert/non-expert dataset (*Full* ≈ 500K). Expert and non-expert columns show the mean return of expert and non-expert policies. **Bold** = best per row.

		Reference		NN				GBT	
Game	**Size**	Non-Expert	Expert	CQL	DT	AWR	BCQ	AWR	**Split-RL**
Qbert	10K	155.3	15704.6	282.0 ± 29.4	12930.5 ± 1393.7	10135.0 ± 1048.2	5729.0 ± 2120.7	10351.0 ± 987.0	**15167.0 ± 421.7**
	20K			5670.0 ± 1788.5	10998.5 ± 1699.9	13983.0 ± 1308.7	6606.0 ± 1368.7	9899.0 ± 927.6	**15723.0 ± 36.1**
	50K			11366.0 ± 1167.2	7970.5 ± 1261.7	15691.0 ± 45.1	459.0 ± 28.8	7807.0 ± 635.3	**15711.0 ± 38.0**
	100K			6009.0 ± 2301.2	5911.5 ± 334.2	**15711.0 ± 38.0**	688.0 ± 27.3	11105.0 ± 439.3	**15711.0 ± 38.0**
	200K			10455.0 ± 1378.7	4670.0 ± 1242.6	10446.0 ± 1372.0	440.0 ± 45.0	8140.0 ± 501.4	**15711.0 ± 38.0**
	Full			8158.0 ± 1312.0	4574.0 ± 28.7	**15711.0 ± 38.0**	402.0 ± 14.9	10561.0 ± 536.1	**15711.0 ± 38.0**
Seaquest	10K	80.9	1822.2	547.6 ± 266.7	1168.0 ± 254.8	712.0 ± 248.4	510.8 ± 269.0	1175.6 ± 66.3	**1596.4 ± 137.1**
	20K			128.0 ± 28.6	1022.0 ± 187.8	598.4 ± 193.3	639.2 ± 259.8	1249.6 ± 59.1	**1725.6 ± 61.5**
	50K			29.2 ± 4.8	1496.4 ± 14.8	656.0 ± 68.2	55.2 ± 13.2	1226.8 ± 67.5	**1821.6 ± 4.8**
	100K			361.6 ± 50.4	1500.4 ± 7.6	548.8 ± 107.6	212.4 ± 51.2	1164.4 ± 61.3	**1821.6 ± 4.8**
	200K			854.0 ± 91.5	1476.8 ± 9.4	1354.4 ± 119.3	84.4 ± 1.5	755.2 ± 62.1	**1821.6 ± 4.8**
	Full			364.0 ± 27.9	1573.2 ± 15.7	1491.6 ± 119.9	536.0 ± 136.6	943.2 ± 72.4	**1821.6 ± 4.8**
Pong	10K	-20.3	21.0	-21.0 ± 0.0	**6.4 ± 6.5**	-20.1 ± 0.4	-1.3 ± 6.7	-1.8 ± 4.2	4.9 ± 3.8
	20K			-5.4 ± 5.5	5.5 ± 6.1	4.0 ± 7.4	-8.5 ± 6.7	-1.6 ± 4.2	**14.3 ± 5.0**
	50K			-20.5 ± 0.2	**21.0 ± 0.0**	14.6 ± 4.8	11.4 ± 3.2	-15.1 ± 1.7	14.8 ± 4.7
	100K			-6.3 ± 6.2	19.4 ± 1.4	-5.0 ± 5.5	-11.9 ± 4.7	-16.9 ± 1.8	**21.0 ± 0.0**
	200K			-19.3 ± 0.2	0.7 ± 6.7	-7.1 ± 3.5	11.4 ± 5.4	-18.3 ± 0.6	**21.0 ± 0.0**
	Full			-19.8 ± 0.2	18.0 ± 2.9	-4.5 ± 5.2	-13.2 ± 3.2	-14.4 ± 3.9	**21.0 ± 0.0**
SpaceInvaders	10K	145.2	1229.1	924.8 ± 188.1	854.6 ± 139.6	1138.4 ± 98.2	385.8 ± 70.2	517.6 ± 38.3	**1263.6 ± 43.1**
	20K			157.8 ± 23.9	1037.8 ± 114.7	1008.8 ± 81.2	511.6 ± 93.3	469.8 ± 40.3	**1156.4 ± 93.4**
	50K			417.6 ± 33.6	1229.0 ± 25.7	833.2 ± 53.6	257.8 ± 46.3	688.6 ± 75.3	**1251.6 ± 44.5**
	100K			262.8 ± 30.3	1229.0 ± 25.7	821.4 ± 77.9	201.0 ± 11.0	441.6 ± 63.3	**1251.6 ± 44.5**
	200K			486.6 ± 62.1	1229.0 ± 25.7	997.8 ± 54.3	236.2 ± 49.0	451.6 ± 51.8	**1251.6 ± 44.5**
	Full			505.0 ± 33.5	1229.0 ± 25.7	915.8 ± 78.5	370.8 ± 46.3	429.0 ± 28.5	**1251.6 ± 44.5**
Alien	10K	206.0	5102.5	675.8 ± 202.9	898.2 ± 231.7	761.6 ± 327.9	763.2 ± 321.9	**1065.0 ± 207.6**	1042.2 ± 386.6
	20K			924.6 ± 322.1	1094.6 ± 129.0	1470.2 ± 609.5	**1603.2 ± 590.8**	945.8 ± 226.0	996.4 ± 73.1
	50K			921.8 ± 185.1	1763.6 ± 154.5	2148.6 ± 272.9	371.2 ± 30.0	1711.8 ± 248.0	**4031.2 ± 643.8**
	100K			1526.6 ± 336.9	1783.6 ± 168.5	2088.6 ± 397.8	429.6 ± 38.5	1752.2 ± 280.9	**4723.4 ± 657.4**
	200K			997.4 ± 147.5	1734.4 ± 61.8	3497.6 ± 486.1	559.2 ± 76.7	2125.4 ± 424.5	**5037.8 ± 733.4**
	Full			1257.8 ± 153.5	1769.0 ± 82.0	3094.6 ± 534.1	664.2 ± 104.0	2232.2 ± 378.0	**5156.2 ± 677.9**
MsPacman	10K	252.2	3597.6	2064.0 ± 615.1	1186.8 ± 280.3	3560.8 ± 839.6	3551.2 ± 841.2	3228.8 ± 266.0	**4031.2 ± 614.9**
	20K			2462.8 ± 29.0	1003.4 ± 256.5	1910.8 ± 296.6	460.8 ± 43.2	3310.4 ± 297.1	**4064.8 ± 600.1**
	50K			3313.6 ± 364.7	1091.2 ± 426.6	1854.0 ± 26.1	767.6 ± 82.7	3512.0 ± 370.2	**4064.8 ± 600.1**
	100K			4020.0 ± 615.6	1284.2 ± 220.3	4013.6 ± 622.1	2103.6 ± 38.3	3665.6 ± 393.4	**4064.8 ± 600.1**
	200K			2187.6 ± 77.1	1252.8 ± 190.0	**4064.8 ± 600.1**	1370.0 ± 279.8	3755.6 ± 410.8	**4064.8 ± 600.1**
	Full			2441.2 ± 24.9	448.0 ± 72.1	1854.0 ± 26.1	878.8 ± 260.6	3034.4 ± 195.8	**4064.8 ± 600.1**
Gravitar	10K	171.5	456.9	344.0 ± 53.0	**390.0 ± 35.4**	320.0 ± 43.2	276.0 ± 44.2	325.0 ± 41.8	383.0 ± 35.4
	20K			230.0 ± 52.2	314.0 ± 64.5	329.0 ± 32.9	241.0 ± 38.0	289.0 ± 102.0	**368.0 ± 65.2**
	50K			232.0 ± 33.7	370.0 ± 70.5	383.0 ± 74.4	293.0 ± 46.1	**401.0 ± 59.4**	371.0 ± 47.3
	100K			267.0 ± 62.3	409.0 ± 81.0	378.0 ± 86.5	311.0 ± 49.6	293.0 ± 48.9	**464.0 ± 57.7**
	200K			357.0 ± 18.6	353.0 ± 78.0	285.0 ± 57.4	332.0 ± 79.0	454.0 ± 38.4	**472.0 ± 60.5**
	Full			350.0 ± 73.7	**431.0 ± 26.3**	261.0 ± 33.4	391.0 ± 88.3	307.0 ± 52.4	393.0 ± 49.5
KungFuMaster	10K	331.5	34312.9	2476.0 ± 660.6	1050.0 ± 719.1	1294.0 ± 1324.4	1098.0 ± 1149.2	1636.0 ± 179.6	**7772.0 ± 2009.9**
	20K			2194.0 ± 737.0	738.0 ± 127.5	1280.0 ± 2150.3	588.0 ± 207.3	3912.0 ± 754.3	**24080.0 ± 3484.8**
	50K			914.0 ± 432.9	2570.0 ± 1713.7	740.0 ± 1180.5	932.0 ± 333.4	3656.0 ± 448.4	**27488.0 ± 2743.8**
	100K			2252.0 ± 805.2	5706.0 ± 3005.5	2926.0 ± 1527.8	1040.0 ± 57.6	5088.0 ± 1413.4	**34992.0 ± 1815.6**
	200K			17692.0 ± 1517.0	3910.0 ± 1297.2	14236.0 ± 3561.0	1388.0 ± 588.8	5840.0 ± 1427.3	**34992.0 ± 1815.6**
	Full			12720.0 ± 3149.9	2904.0 ± 1215.4	18258.0 ± 2892.9	2716.0 ± 678.9	5312.0 ± 2157.8	**34992.0 ± 1815.6**
Frostbite	10K	74.0	931.2	549.6 ± 239.0	858.8 ± 324.0	875.2 ± 262.6	**1009.0 ± 264.5**	144.0 ± 9.0	966.6 ± 276.9
	20K			636.2 ± 205.3	**992.0 ± 253.0**	986.8 ± 273.5	943.6 ± 237.5	920.0 ± 257.9	971.2 ± 229.4
	50K			67.2 ± 17.0	**1060.2 ± 330.4**	855.6 ± 276.4	111.6 ± 5.4	163.0 ± 7.2	921.4 ± 247.7
	100K			111.2 ± 9.3	**1116.0 ± 269.2**	886.0 ± 257.3	107.4 ± 4.2	158.2 ± 10.3	970.4 ± 247.3
	200K			134.6 ± 17.3	**1071.8 ± 282.8**	473.6 ± 173.5	54.4 ± 10.4	139.8 ± 7.9	924.8 ± 245.6
	Full			127.2 ± 9.6	**1103.6 ± 261.8**	1009.0 ± 263.0	853.8 ± 367.3	150.4 ± 6.0	907.8 ± 251.2
Amidar	10K	1.2	1049.8	66.2 ± 19.0	381.6 ± 117.0	252.0 ± 84.0	0.0 ± 0.0	**513.2 ± 59.6**	472.7 ± 55.6
	20K			2.8 ± 0.4	916.5 ± 103.0	882.0 ± 126.0	811.7 ± 126.1	699.8 ± 45.3	**1050.0 ± 0.0**
	50K			300.2 ± 21.8	**1049.9 ± 0.1**	881.8 ± 55.9	1049.8 ± 0.1	629.7 ± 47.1	1049.8 ± 0.1
	100K			166.1 ± 26.3	**1049.9 ± 0.1**	1049.8 ± 0.1	888.8 ± 46.6	569.6 ± 54.7	1049.8 ± 0.1
	200K			265.2 ± 68.8	**1049.9 ± 0.1**	943.6 ± 40.0	17.0 ± 0.9	572.8 ± 37.1	1049.8 ± 0.1
	Full			754.4 ± 91.0	**1049.9 ± 0.1**	1049.8 ± 0.1	927.7 ± 51.6	568.4 ± 54.5	1049.8 ± 0.1
BankHeist	10K	14.1	1361.2	67.8 ± 47.8	647.4 ± 85.2	43.2 ± 40.6	23.2 ± 4.9	**1373.8 ± 1.6**	1354.2 ± 16.1
	20K			169.6 ± 161.7	1164.2 ± 47.3	98.4 ± 101.9	58.4 ± 51.4	**1377.2 ± 1.5**	1368.0 ± 5.8
	50K			688.8 ± 121.8	1331.6 ± 18.0	272.0 ± 160.6	64.2 ± 103.9	**1375.4 ± 1.5**	1355.4 ± 16.0
	100K			1131.4 ± 16.3	1371.6 ± 10.1	127.2 ± 101.7	64.2 ± 23.7	**1372.0 ± 3.0**	1369.2 ± 8.9
	200K			1362.0 ± 1.3	**1374.4 ± 4.8**	14.0 ± 2.5	152.0 ± 104.9	1374.2 ± 1.8	1365.8 ± 7.1
	Full			1362.0 ± 6.7	1370.0 ± 10.8	641.0 ± 161.5	28.8 ± 1.8	**1372.8 ± 3.8**	1366.2 ± 5.6

Table 16: **BC target ablation — per-game raw returns.** At the final training step, we average raw return over 10 episodes per evaluation seed and report the mean ± std of the five seed averages. *Non-Expert* and *Expert* columns show reference returns used for normalization. See Table 4 for method descriptions. **Bold** = best per row.

			NN			GBT			
			AWR + BC			AWR + BC			Split-RL
Environment	Non-Expert	Expert	all	expert	masked	all	expert	masked	
Qbert	155.3	15704.6	**15711.0 ± 38.0**	**15711.0 ± 38.0**	15594.0 ± 29.8	9878.0 ± 308.8	**15711.0 ± 38.0**	10322.0 ± 579.0	**15711.0 ± 38.0**
Seaquest	80.9	1822.2	1238.4 ± 114.4	1465.2 ± 157.9	712.8 ± 161.7	953.2 ± 35.6	1258.0 ± 60.2	1005.6 ± 105.8	**1821.6 ± 4.8**
Pong	-20.3	21.0	-5.0 ± 5.5	-2.0 ± 4.3	-10.9 ± 3.4	0.7 ± 5.6	**21.0 ± 0.0**	11.4 ± 5.4	**21.0 ± 0.0**
SpaceInvaders	145.2	1229.1	551.6 ± 46.8	1245.6 ± 43.5	1220.0 ± 46.0	542.4 ± 20.3	801.4 ± 77.2	471.6 ± 34.2	**1251.6 ± 44.5**
Alien	206.0	5102.5	4196.0 ± 461.9	3510.0 ± 565.4	3276.8 ± 232.2	2886.8 ± 268.1	4094.0 ± 333.9	1254.0 ± 76.2	**5156.2 ± 677.9**
MsPacman	252.2	3597.6	4020.8 ± 619.0	4064.8 ± 600.1	3775.2 ± 482.7	4293.2 ± 562.4	**4612.0 ± 668.4**	4298.8 ± 553.6	4064.8 ± 600.1
Gravitar	171.5	456.9	355.0 ± 98.5	347.0 ± 52.0	254.0 ± 78.9	**448.0 ± 44.6**	436.0 ± 13.2	266.0 ± 67.6	393.0 ± 49.5
KungFuMaster	331.5	34312.9	4642.0 ± 2638.3	6480.0 ± 3962.9	0.0 ± 0.0	8128.0 ± 982.5	11524.0 ± 2301.4	1976.0 ± 363.2	**34992.0 ± 1815.6**
Frostbite	74.0	931.2	853.6 ± 221.7	959.8 ± 229.3	**1056.6 ± 243.6**	959.0 ± 264.2	944.8 ± 238.1	152.0 ± 9.8	907.8 ± 251.2
Amidar	1.2	1049.8	836.2 ± 63.4	**1049.8 ± 0.1**	896.0 ± 45.2	649.0 ± 72.9	**1049.8 ± 0.1**	733.1 ± 54.9	**1049.8 ± 0.1**
BankHeist	14.1	1361.2	915.4 ± 180.0	1365.6 ± 12.7	16.8 ± 2.3	1128.0 ± 89.9	**1376.6 ± 1.6**	1260.0 ± 101.5	1366.2 ± 5.6

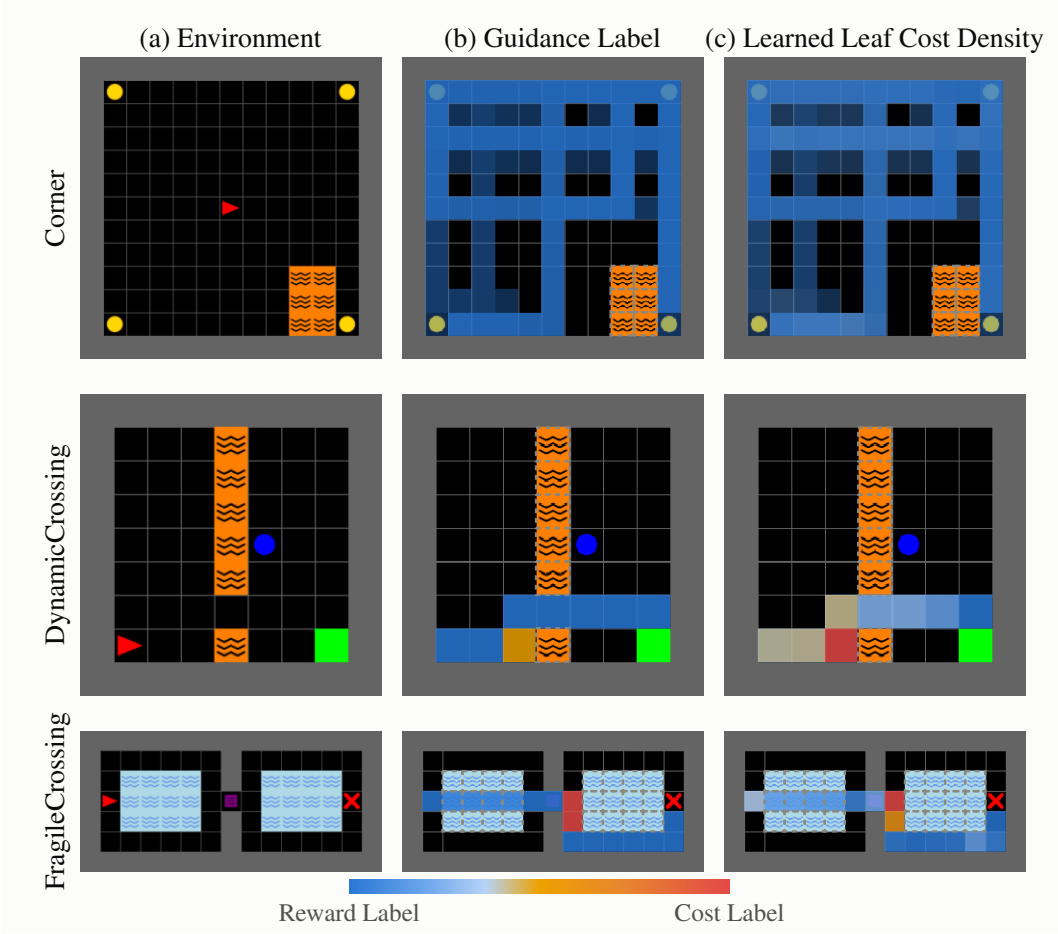


Figure 8: **Guided routing density under policy rollouts, all environments.** Same measurement as Figure 5, for Corner, DynamicCrossing, and FragileCrossing.

F.2 Robustness to Label Noise

We corrupt the guidance label with probability $p \in \{0, 0.1, 0.25, 0.5, 1.0\}$: each label is retained with probability $1 - p$ and independently redrawn uniformly at random with probability p before it reaches the split-selection procedure. Figure 4 shows reward and cost across all three MiniGrid environments as p increases. Performance is stable up to $p = 0.25$: cost stays near its $p = 0$ value in both DynamicCrossing and FragileCrossing, and reward is essentially unchanged across the whole sweep, including at $p = 1.0$, where the label carries no information. Cost rises sharply once $p \geq 0.5$,

which is expected: Split-RL acts on the label rather than consulting it, so an uninformative label produces arbitrary routing rather than a graceful fallback to unguided training. The reward/cost split in this result matches the failure mode the paper is built around: unsafe states are a minority of the distribution, so cost is the rare-but-critical signal that guidance protects against dilution; reward, driven by the majority distribution, is far less sensitive to losing that guidance.

F.3 Privileged-Information Ablation

Section 6 compares all five constrained-RL baselines under three training conditions. No-label results are already reported in Tables 12–14. Tables 17–22 below report the same metrics for Label-in-Obs and Masked, one table per domain per condition, in the same five-algorithm-column format as Table 13. Split-RL is not included in either condition, since neither is a Split-RL variant. Mean, std, and the 95% bootstrap CI are shown for each entry, matching Tables 12–14.

Four of the five baselines change by at most one scenario under either condition. CPO under Masked is the exception, gaining four, but the gains are not uniform relative to its own no-label results: in FragileCrossing feasibility comes with reward falling from 0.48 to 0.26; in Corner its cost of 0.10 ± 0.39 lies on the 0.1 limit with a bootstrap CI reaching 0.31; in Single/Premium masking stabilizes the run, with cost falling from 38.26 ± 84.51 to 11.36 ± 3.49; and in Solar Ramp Reserve improves on both axes, from reward -7.84 at cost 7.33 to -7.65 at 2.85. Where CPO was already infeasible it can also degrade: in DynamicCrossing its cost rises from 0.34 to 17.87, and in Arbitrage vs. Buffer its reward falls from -7.96 to -29.75. Across all five baselines, Masked changes $+4$ of 75 scenario verdicts and Label-in-Obs -1.

F.3.1 Label-in-Obs

Tables 17–19 report SUMO-RL, CityLearn, and MiniGrid with the guidance label appended to the observation as a one-hot feature.

Table 17: **Label-in-Obs, SUMO-RL.** Mean $\pm$ std, with 95% bootstrap CI, over seeds. The guidance label is concatenated into the observation; otherwise standard baselines. Reward is in units of 10^3; cost is unscaled. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Environment	Scenario	Metric	CPO	CUP	IPO	PPOLag	PPOLag (GBT)
Arterial 4x4	Bus	Rew.	$\mathbf{-1.403 \pm 0.341}$ $\mathbf{[-1.576, -1.248]}$	$\mathbf{-1.284 \pm 0.186}$ $\mathbf{[-1.373, -1.197]}$	$\mathbf{-1.295 \pm 0.195}$ $\mathbf{[-1.395, -1.199]}$	-1.182 ± 0.185 $[-1.275, -1.091]$	-1.250 ± 0.175 $[-1.351, -1.179]$
		Cost	$\mathbf{3.76 \pm 2.49}$ $\mathbf{[2.56, 4.96]}$	$\mathbf{1.96 \pm 2.04}$ $\mathbf{[1.08, 3.04]}$	$\mathbf{4.63 \pm 4.81}$ $\mathbf{[2.68, 7.31]}$	8.11 ± 7.48 $[4.94, 12.19]$	24.82 ± 14.42 $[18.09, 31.92]$
	Convoy	Rew.	$\mathbf{-1.483 \pm 0.354}$ $\mathbf{[-1.648, -1.317]}$	$\mathbf{-1.476 \pm 0.262}$ $\mathbf{[-1.607, -1.352]}$	$\mathbf{-1.264 \pm 0.160}$ $\mathbf{[-1.349, -1.190]}$	-1.245 ± 0.193 $[-1.340, -1.154]$	-1.249 ± 0.144 $[-1.321, -1.183]$
		Cost	$\mathbf{3.89 \pm 3.22}$ $\mathbf{[2.42, 5.62]}$	$\mathbf{1.77 \pm 1.71}$ $\mathbf{[1.05, 2.67]}$	$\mathbf{1.57 \pm 1.48}$ $\mathbf{[1.04, 2.48]}$	11.70 ± 4.04 $[9.77, 13.59]$	14.77 ± 7.14 $[11.78, 18.78]$
	Premium	Rew.	$\mathbf{-1.397 \pm 0.281}$ $\mathbf{[-1.541, -1.264]}$	$\mathbf{-1.485 \pm 0.327}$ $\mathbf{[-1.653, -1.335]}$	$\mathbf{-1.362 \pm 0.327}$ $\mathbf{[-1.534, -1.212]}$	-1.194 ± 0.216 $[-1.302, -1.096]$	-1.174 ± 0.136 $[-1.243, -1.114]$
		Cost	$\mathbf{6.75 \pm 4.11}$ $\mathbf{[4.96, 8.91]}$	$\mathbf{6.33 \pm 3.83}$ $\mathbf{[4.59, 8.10]}$	$\mathbf{6.99 \pm 5.58}$ $\mathbf{[4.75, 9.93]}$	15.35 ± 8.14 $[12.14, 19.89]$	34.91 ± 13.59 $[28.49, 41.39]$
Grid 4x4	Bus	Rew.	$\mathbf{-0.198 \pm 0.011}$ $\mathbf{[-0.203, -0.192]}$	$\mathbf{-0.138 \pm 0.005}$ $\mathbf{[-0.140, -0.135]}$	$\mathbf{-0.150 \pm 0.004}$ $\mathbf{[-0.152, -0.148]}$	$\mathbf{-0.119 \pm 0.004}$ $\mathbf{[-0.121, -0.117]}$	$\mathbf{-0.135 \pm 0.004}$ $\mathbf{[-0.138, -0.133]}$
		Cost	$\mathbf{0.24 \pm 0.18}$ $\mathbf{[0.16, 0.33]}$	$\mathbf{0.06 \pm 0.07}$ $\mathbf{[0.03, 0.10]}$	$\mathbf{0.01 \pm 0.01}$ $\mathbf{[0.01, 0.02]}$	$\mathbf{0.13 \pm 0.08}$ $\mathbf{[0.10, 0.18]}$	$\mathbf{0.29 \pm 0.12}$ $\mathbf{[0.23, 0.34]}$
	Convoy	Rew.	$\mathbf{-0.341 \pm 0.333}$ $\mathbf{[-0.515, -0.251]}$	$\mathbf{-0.163 \pm 0.012}$ $\mathbf{[-0.168, -0.157]}$	$\mathbf{-0.173 \pm 0.010}$ $\mathbf{[-0.178, -0.168]}$	-0.140 ± 0.009 $[-0.144, -0.136]$	-0.165 ± 0.005 $[-0.168, -0.163]$
		Cost	$\mathbf{2.73 \pm 1.93}$ $\mathbf{[2.05, 3.76]}$	$\mathbf{1.36 \pm 0.55}$ $\mathbf{[1.11, 1.63]}$	$\mathbf{2.48 \pm 0.81}$ $\mathbf{[2.09, 2.85]}$	6.87 ± 2.06 $[5.89, 7.88]$	10.54 ± 0.62 $[10.25, 10.86]$
	Premium	Rew.	$\mathbf{-0.211 \pm 0.012}$ $\mathbf{[-0.217, -0.205]}$	$\mathbf{-0.144 \pm 0.006}$ $\mathbf{[-0.147, -0.141]}$	$\mathbf{-0.154 \pm 0.005}$ $\mathbf{[-0.157, -0.152]}$	$\mathbf{-0.124 \pm 0.006}$ $\mathbf{[-0.127, -0.121]}$	$\mathbf{-0.141 \pm 0.004}$ $\mathbf{[-0.144, -0.139]}$
		Cost	$\mathbf{4.88 \pm 2.12}$ $\mathbf{[3.87, 5.92]}$	$\mathbf{2.04 \pm 0.68}$ $\mathbf{[1.70, 2.38]}$	$\mathbf{2.74 \pm 1.04}$ $\mathbf{[2.23, 3.22]}$	$\mathbf{3.93 \pm 1.20}$ $\mathbf{[3.36, 4.51]}$	$\mathbf{4.83 \pm 1.32}$ $\mathbf{[4.13, 5.43]}$
Single VHVH	Bus	Rew.	$\mathbf{-8.847 \pm 0.255}$ $\mathbf{[-8.980, -8.727]}$	$\mathbf{-8.679 \pm 0.168}$ $\mathbf{[-8.761, -8.594]}$	$\mathbf{-8.717 \pm 0.141}$ $\mathbf{[-8.789, -8.651]}$	$\mathbf{-8.428 \pm 0.071}$ $\mathbf{[-8.462, -8.394]}$	$\mathbf{-8.542 \pm 0.106}$ $\mathbf{[-8.595, -8.488]}$
		Cost	$\mathbf{2.95 \pm 1.48}$ $\mathbf{[2.26, 3.70]}$	$\mathbf{3.11 \pm 1.07}$ $\mathbf{[2.58, 3.64]}$	$\mathbf{3.85 \pm 0.80}$ $\mathbf{[3.48, 4.26]}$	$\mathbf{3.35 \pm 0.67}$ $\mathbf{[3.03, 3.67]}$	$\mathbf{3.54 \pm 0.78}$ $\mathbf{[3.17, 3.92]}$
	Convoy	Rew.	-73.796 ± 9.963 $[-78.314, -68.208]$	-30.291 ± 3.850 $[-32.178, -28.389]$	-13.762 ± 0.581 $[-14.038, -13.482]$	-17.361 ± 2.338 $[-18.535, -16.265]$	-15.015 ± 1.140 $[-15.610, -14.514]$
		Cost	28.09 ± 22.24 $[18.14, 40.15]$	8.08 ± 9.85 $[3.96, 13.36]$	21.92 ± 4.47 $[19.90, 24.09]$	20.78 ± 4.49 $[18.65, 22.96]$	72.71 ± 11.90 $[66.88, 78.77]$
	Premium	Rew.	-14.990 ± 17.560 $[-25.189, -9.147]$	$\mathbf{-9.032 \pm 0.200}$ $\mathbf{[-9.127, -8.933]}$	$\mathbf{-8.918 \pm 0.254}$ $\mathbf{[-9.058, -8.810]}$	$\mathbf{-8.613 \pm 0.082}$ $\mathbf{[-8.648, -8.569]}$	-8.570 ± 0.073 $[-8.604, -8.532]$
		Cost	21.71 ± 44.48 $[8.76, 45.25]$	$\mathbf{7.98 \pm 3.29}$ $\mathbf{[6.48, 9.70]}$	$\mathbf{14.33 \pm 1.34}$ $\mathbf{[13.62, 14.98]}$	$\mathbf{14.51 \pm 2.10}$ $\mathbf{[13.49, 15.50]}$	24.27 ± 1.71 $[23.42, 25.12]$

Table 18: **Label-in-Obs, CityLearn.** Mean ± std, with 95% bootstrap CI, over seeds. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Scenario	Metric	CPO	CUP	IPO	PPOLag	PPOLag (GBT)
Arb. vs Buffer	Rew.	-7.75 ± 0.37 $[-7.93, -7.58]$	-8.45 ± 0.33 $[-8.57, -8.29]$	-8.61 ± 0.06 $[-8.64, -8.58]$	-8.11 ± 0.25 $[-8.22, -7.98]$	-8.84 ± 0.13 $[-8.91, -8.78]$
	Cost	7.33 ± 3.28 $[5.76, 9.00]$	7.26 ± 9.64 $[2.94, 12.25]$	6.08 ± 5.77 $[3.37, 9.00]$	12.91 ± 5.69 $[10.26, 15.85]$	9.61 ± 3.60 $[7.75, 11.29]$
Demand Response	Rew.	-7.63 ± 0.38 $[-7.81, -7.45]$	-7.80 ± 0.86 $[-8.30, -7.44]$	-8.33 ± 0.33 $[-8.48, -8.15]$	-8.42 ± 0.25 $[-8.51, -8.28]$	-8.92 ± 0.10 $[-8.98, -8.87]$
	Cost	8.78 ± 3.63 $[6.88, 10.57]$	9.11 ± 8.40 $[5.44, 13.70]$	7.03 ± 5.04 $[4.71, 9.76]$	28.29 ± 0.15 $[28.22, 28.36]$	10.46 ± 3.05 $[8.82, 12.04]$
Solar Ramp	Rew.	-7.77 ± 0.47 $[-8.01, -7.55]$	-8.07 ± 0.41 $[-8.27, -7.87]$	-8.55 ± 0.38 $[-8.72, -8.35]$	-8.40 ± 0.19 $[-8.49, -8.30]$	-8.98 ± 0.10 $[-9.02, -8.93]$
	Cost	5.75 ± 3.22 $[4.23, 7.39]$	7.66 ± 3.97 $[5.70, 9.60]$	9.04 ± 3.77 $[7.11, 10.69]$	35.83 ± 24.31 $[24.68, 47.94]$	10.96 ± 2.76 $[9.59, 12.37]$

Table 19: **Label-in-Obs, Safety MiniGrid.** Mean ± std, with 95% bootstrap CI, final training steps. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Environment	Metric	CUP	CPO	IPO	PPOLag	PPOLag (GBT)
DynamicCrossing	Reward	0.15 ± 0.10 $[0.10, 0.20]$	0.14 ± 0.13 $[0.07, 0.20]$	0.29 ± 0.30 $[0.17, 0.46]$	0.16 ± 0.35 $[0.04, 0.35]$	$\mathbf{1.48 \pm 0.17}$ $[\mathbf{1.39}, \mathbf{1.53}]$
	Cost	6.39 ± 23.96 $[0.11, 18.80]$	0.12 ± 0.08 $[0.09, 0.17]$	4.61 ± 10.01 $[1.13, 10.16]$	26.80 ± 79.85 $[2.03, 71.43]$	$\mathbf{0.01 \pm 0.01}$ $[\mathbf{0.00}, \mathbf{0.01}]$
FragileCrossing	Reward	$\mathbf{-0.02 \pm 0.13}$ $[\mathbf{-0.08}, \mathbf{0.04}]$	0.27 ± 0.08 $[0.23, 0.31]$	$\mathbf{0.19 \pm 0.22}$ $[\mathbf{0.08}, \mathbf{0.30}]$	$\mathbf{0.10 \pm 0.18}$ $[\mathbf{0.02}, \mathbf{0.19}]$	0.90 ± 0.20 $[0.80, 0.99]$
	Cost	$\mathbf{0.00 \pm 0.00}$ $[\mathbf{0.00}, \mathbf{0.00}]$	0.15 ± 0.29 $[0.04, 0.32]$	$\mathbf{0.04 \pm 0.08}$ $[\mathbf{0.00}, \mathbf{0.08}]$	$\mathbf{0.02 \pm 0.07}$ $[\mathbf{0.00}, \mathbf{0.06}]$	0.20 ± 0.50 $[0.03, 0.46]$
Corner	Reward	0.06 ± 0.21 $[0.00, 0.17]$	0.73 ± 0.30 $[0.57, 0.87]$	0.00 ± 0.01 $[0.00, 0.01]$	0.00 ± 0.00 $[0.00, 0.00]$	$\mathbf{1.00 \pm 0.00}$ $[\mathbf{1.00}, \mathbf{1.00}]$
	Cost	0.37 ± 1.15 $[0.00, 1.04]$	0.86 ± 0.98 $[0.41, 1.39]$	25.23 ± 72.59 $[1.56, 64.45]$	3.91 ± 10.84 $[0.17, 10.42]$	$\mathbf{0.00 \pm 0.00}$ $[\mathbf{0.00}, \mathbf{0.00}]$

F.3.2 Masked

Tables 20–22 report the same three domains with the label used to mask the losses: reward-labelled samples enter only the reward update, cost-labelled samples only the cost update.

Table 20: **Masked, SUMO-RL.** Mean ± std, with 95% bootstrap CI, over seeds. Reward-labelled samples get only the reward update, cost-labelled samples get only the cost update, no conflict-aware score. Reward is in units of 10^3; cost is unscaled. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Environment	Scenario	Metric	CPO	CUP	IPO	PPOLag	PPOLag (GBT)
Arterial 4x4	Bus	Rew.	$\mathbf{-1.432 \pm 0.292}$	$\mathbf{-1.264 \pm 0.165}$	$\mathbf{-1.308 \pm 0.286}$	$\mathbf{-1.131 \pm 0.222}$	-1.196 ± 0.147
			$[\mathbf{-1.577}, \mathbf{-1.291}]$	$[\mathbf{-1.350}, \mathbf{-1.189}]$	$[\mathbf{-1.472}, \mathbf{-1.181}]$	$[\mathbf{-1.243}, \mathbf{-1.038}]$	$[-1.272, -1.128]$
		Cost	$\mathbf{3.69 \pm 3.77}$	$\mathbf{2.34 \pm 1.96}$	$\mathbf{2.18 \pm 2.07}$	$\mathbf{6.00 \pm 4.05}$	19.07 ± 8.05
			$[\mathbf{2.10}, \mathbf{5.81}]$	$[\mathbf{1.47}, \mathbf{3.41}]$	$[\mathbf{1.30}, \mathbf{3.34}]$	$[\mathbf{4.16}, \mathbf{8.04}]$	$[15.16, 22.86]$
	Convoy	Rew.	$\mathbf{-1.402 \pm 0.243}$	-1.411 ± 0.216	$\mathbf{-1.491 \pm 0.298}$	-1.288 ± 0.161	-1.219 ± 0.178
			$[\mathbf{-1.522}, \mathbf{-1.286}]$	$[-1.515, -1.312]$	$[\mathbf{-1.663}, \mathbf{-1.354}]$	$[-1.366, -1.212]$	$[-1.312, -1.138]$
		Cost	$\mathbf{4.63 \pm 3.93}$	6.71 ± 5.22	$\mathbf{4.47 \pm 4.50}$	8.00 ± 3.81	15.72 ± 3.00
			$[\mathbf{2.93}, \mathbf{6.80}]$	$[4.20, 9.38]$	$[\mathbf{2.37}, \mathbf{6.86}]$	$[6.11, 9.99]$	$[14.15, 17.14]$
	Premium	Rew.	$\mathbf{-1.431 \pm 0.237}$	$\mathbf{-1.356 \pm 0.245}$	$\mathbf{-1.223 \pm 0.155}$	-1.157 ± 0.140	-1.199 ± 0.141
			$[\mathbf{-1.550}, \mathbf{-1.310}]$	$[\mathbf{-1.483}, \mathbf{-1.245}]$	$[\mathbf{-1.302}, \mathbf{-1.150}]$	$[-1.227, -1.091]$	$[-1.272, -1.133]$
		Cost	$\mathbf{5.43 \pm 2.24}$	$\mathbf{7.26 \pm 5.44}$	$\mathbf{5.34 \pm 2.03}$	21.66 ± 14.76	42.71 ± 10.15
			$[\mathbf{4.37}, \mathbf{6.53}]$	$[\mathbf{4.80}, \mathbf{10.07}]$	$[\mathbf{4.34}, \mathbf{6.34}]$	$[15.16, 29.69]$	$[37.85, 47.78]$
Grid 4x4	Bus	Rew.	$\mathbf{-0.195 \pm 0.008}$	$\mathbf{-0.139 \pm 0.005}$	$\mathbf{-0.147 \pm 0.005}$	$\mathbf{-0.117 \pm 0.003}$	$\mathbf{-0.135 \pm 0.003}$
			$[\mathbf{-0.199}, \mathbf{-0.191}]$	$[\mathbf{-0.141}, \mathbf{-0.136}]$	$[\mathbf{-0.150}, \mathbf{-0.145}]$	$[\mathbf{-0.118}, \mathbf{-0.115}]$	$[\mathbf{-0.136}, \mathbf{-0.133}]$
		Cost	$\mathbf{0.20 \pm 0.11}$	$\mathbf{0.15 \pm 0.19}$	$\mathbf{0.24 \pm 0.33}$	$\mathbf{0.19 \pm 0.19}$	$\mathbf{0.35 \pm 0.16}$
			$[\mathbf{0.15}, \mathbf{0.26}]$	$[\mathbf{0.07}, \mathbf{0.26}]$	$[\mathbf{0.10}, \mathbf{0.42}]$	$[\mathbf{0.11}, \mathbf{0.31}]$	$[\mathbf{0.27}, \mathbf{0.43}]$
	Convoy	Rew.	$\mathbf{-0.457 \pm 0.529}$	$\mathbf{-0.172 \pm 0.010}$	$\mathbf{-0.179 \pm 0.010}$	$\mathbf{-0.149 \pm 0.007}$	-0.172 ± 0.004
			$[\mathbf{-0.756}, \mathbf{-0.252}]$	$[\mathbf{-0.177}, \mathbf{-0.167}]$	$[\mathbf{-0.183}, \mathbf{-0.174}]$	$[\mathbf{-0.152}, \mathbf{-0.146}]$	$[-0.174, -0.170]$
		Cost	$\mathbf{3.66 \pm 3.92}$	$\mathbf{1.16 \pm 0.52}$	$\mathbf{2.14 \pm 0.70}$	$\mathbf{3.78 \pm 0.77}$	13.25 ± 0.78
			$[\mathbf{2.27}, \mathbf{5.91}]$	$[\mathbf{0.91}, \mathbf{1.42}]$	$[\mathbf{1.79}, \mathbf{2.48}]$	$[\mathbf{3.41}, \mathbf{4.19}]$	$[12.90, 13.65]$
	Premium	Rew.	$\mathbf{-0.206 \pm 0.013}$	$\mathbf{-0.145 \pm 0.009}$	$\mathbf{-0.154 \pm 0.007}$	$\mathbf{-0.119 \pm 0.004}$	$\mathbf{-0.139 \pm 0.003}$
			$[\mathbf{-0.212}, \mathbf{-0.199}]$	$[\mathbf{-0.150}, \mathbf{-0.141}]$	$[\mathbf{-0.157}, \mathbf{-0.151}]$	$[\mathbf{-0.122}, \mathbf{-0.117}]$	$[\mathbf{-0.140}, \mathbf{-0.137}]$
		Cost	$\mathbf{6.01 \pm 1.78}$	$\mathbf{1.60 \pm 0.52}$	$\mathbf{3.35 \pm 1.14}$	$\mathbf{2.84 \pm 1.26}$	$\mathbf{5.86 \pm 1.56}$
			$[\mathbf{5.15}, \mathbf{6.84}]$	$[\mathbf{1.34}, \mathbf{1.84}]$	$[\mathbf{2.82}, \mathbf{3.92}]$	$[\mathbf{2.24}, \mathbf{3.48}]$	$[\mathbf{5.06}, \mathbf{6.56}]$
Single VHVH	Bus	Rew.	$\mathbf{-8.754 \pm 0.227}$	$\mathbf{-8.704 \pm 0.203}$	$\mathbf{-8.702 \pm 0.140}$	$\mathbf{-8.457 \pm 0.085}$	$\mathbf{-8.516 \pm 0.053}$
			$[\mathbf{-8.860}, \mathbf{-8.638}]$	$[\mathbf{-8.801}, \mathbf{-8.610}]$	$[\mathbf{-8.769}, \mathbf{-8.634}]$	$[\mathbf{-8.499}, \mathbf{-8.415}]$	$[\mathbf{-8.542}, \mathbf{-8.489}]$
		Cost	$\mathbf{3.03 \pm 1.12}$	$\mathbf{3.00 \pm 1.02}$	$\mathbf{4.62 \pm 0.96}$	$\mathbf{2.92 \pm 0.89}$	$\mathbf{4.20 \pm 0.79}$
			$[\mathbf{2.55}, \mathbf{3.62}]$	$[\mathbf{2.48}, \mathbf{3.45}]$	$[\mathbf{4.19}, \mathbf{5.12}]$	$[\mathbf{2.50}, \mathbf{3.37}]$	$[\mathbf{3.83}, \mathbf{4.59}]$
	Convoy	Rew.	-72.898 ± 13.303	-32.999 ± 3.678	-27.005 ± 2.706	-28.032 ± 4.621	-16.880 ± 2.612
			$[-78.773, -66.007]$	$[-34.795, -31.246]$	$[-28.338, -25.603]$	$[-30.199, -25.745]$	$[-18.187, -15.591]$
		Cost	21.92 ± 13.75	15.95 ± 16.10	10.23 ± 8.09	12.16 ± 3.75	37.11 ± 7.33
			$[15.19, 28.76]$	$[8.62, 24.34]$	$[7.44, 14.78]$	$[10.32, 14.08]$	$[33.64, 40.56]$
	Premium	Rew.	$\mathbf{-8.989 \pm 0.167}$	$\mathbf{-9.005 \pm 0.224}$	$\mathbf{-8.989 \pm 0.396}$	$\mathbf{-8.586 \pm 0.084}$	-8.609 ± 0.088
			$[\mathbf{-9.068}, \mathbf{-8.910}]$	$[\mathbf{-9.121}, \mathbf{-8.905}]$	$[\mathbf{-9.208}, \mathbf{-8.824}]$	$[\mathbf{-8.626}, \mathbf{-8.545}]$	$[-8.652, -8.565]$
		Cost	$\mathbf{11.36 \pm 3.49}$	$\mathbf{9.85 \pm 2.79}$	$\mathbf{14.28 \pm 1.00}$	$\mathbf{14.74 \pm 2.25}$	25.69 ± 2.26
			$[\mathbf{9.65}, \mathbf{13.13}]$	$[\mathbf{8.59}, \mathbf{11.15}]$	$[\mathbf{13.81}, \mathbf{14.76}]$	$[\mathbf{13.67}, \mathbf{15.92}]$	$[24.62, 26.82]$

Table 21: **Masked, CityLearn.** Mean ± std, with 95% bootstrap CI, over seeds. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Scenario	Metric	CPO	CUP	IPO	PPOLag	PPOLag (GBT)
Arb. vs Buffer	Rew.	-29.75 ± 7.97	-7.23 ± 0.29	-11.24 ± 0.13	-8.19 ± 0.43	-8.79 ± 0.04
		$[-33.72, -25.81]$	$[-7.39, -7.11]$	$[-11.30, -11.18]$	$[-8.40, -7.97]$	$[-8.80, -8.77]$
	Cost	9.92 ± 2.71	206.65 ± 16.22	61.00 ± 5.57	238.68 ± 18.58	211.04 ± 9.11
		$[8.58, 11.21]$	$[198.48, 214.25]$	$[58.47, 63.99]$	$[229.94, 248.23]$	$[206.01, 215.09]$
Demand Response	Rew.	-7.55 ± 0.35	-7.77 ± 1.00	-10.19 ± 0.80	-8.48 ± 0.11	-8.84 ± 0.09
		$[-7.73, -7.38]$	$[-8.30, -7.33]$	$[-10.58, -9.80]$	$[-8.51, -8.41]$	$[-8.88, -8.80]$
	Cost	6.07 ± 4.07	10.55 ± 8.60	5.31 ± 1.26	28.24 ± 0.16	12.50 ± 3.70
		$[4.41, 8.16]$	$[6.78, 15.04]$	$[4.74, 5.93]$	$[28.16, 28.32]$	$[10.84, 14.39]$
Solar Ramp	Rew.	$\mathbf{-7.65 \pm 0.54}$	$\mathbf{-8.09 \pm 0.67}$	$\mathbf{-8.57 \pm 0.54}$	-8.77 ± 0.44	-9.11 ± 0.07
		$[\mathbf{-7.96}, \mathbf{-7.42}]$	$[\mathbf{-8.43}, \mathbf{-7.78}]$	$[\mathbf{-8.82}, \mathbf{-8.29}]$	$[-9.02, -8.59]$	$[-9.14, -9.08]$
	Cost	$\mathbf{2.85 \pm 1.52}$	$\mathbf{4.62 \pm 3.13}$	$\mathbf{3.58 \pm 1.65}$	5.20 ± 3.90	10.76 ± 1.01
		$[\mathbf{2.09}, \mathbf{3.56}]$	$[\mathbf{3.16}, \mathbf{6.09}]$	$[\mathbf{2.77}, \mathbf{4.34}]$	$[3.39, 7.20]$	$[10.25, 11.25]$

Table 22: **Masked, Safety MiniGrid.** Mean $\pm$ std, with 95% bootstrap CI, final training steps. **Bold** marks entries where the method is feasible (cost within the limit in Table 11); unbolded entries exceed the limit.

Environment	Metric	CUP	CPO	IPO	PPOLag	PPOLag (GBT)
DynamicCrossing	Reward	0.20 ± 0.20 $[0.11, 0.31]$	0.19 ± 0.13 $[0.14, 0.26]$	0.39 ± 0.32 $[0.25, 0.56]$	0.07 ± 0.14 $[0.01, 0.14]$	1.20 ± 0.26 $[1.08, 1.32]$
	Cost	45.64 ± 109.96 $[2.19, 107.89]$	17.87 ± 20.52 $[8.68, 29.17]$	20.82 ± 35.60 $[6.25, 40.96]$	31.98 ± 73.13 $[2.70, 73.67]$	0.36 ± 0.31 $[0.22, 0.52]$
FragileCrossing	Reward	$\mathbf{0.12 \pm 0.19}$ $[\mathbf{0.03}, \mathbf{0.22}]$	$\mathbf{0.26 \pm 0.13}$ $[\mathbf{0.20}, \mathbf{0.32}]$	0.18 ± 0.23 $[0.07, 0.30]$	0.22 ± 0.20 $[0.12, 0.32]$	$\mathbf{0.64 \pm 0.01}$ $[\mathbf{0.64}, \mathbf{0.65}]$
	Cost	$\mathbf{0.05 \pm 0.18}$ $[\mathbf{0.00}, \mathbf{0.14}]$	$\mathbf{0.09 \pm 0.09}$ $[\mathbf{0.05}, \mathbf{0.14}]$	0.26 ± 0.63 $[0.01, 0.61]$	0.13 ± 0.38 $[0.00, 0.35]$	$\mathbf{0.01 \pm 0.02}$ $[\mathbf{0.00}, \mathbf{0.02}]$
Corner	Reward	0.04 ± 0.14 $[0.00, 0.11]$	$\mathbf{0.95 \pm 0.10}$ $[\mathbf{0.89}, \mathbf{0.99}]$	0.73 ± 0.28 $[0.58, 0.85]$	0.02 ± 0.08 $[0.00, 0.06]$	$\mathbf{1.00 \pm 0.00}$ $[\mathbf{1.00}, \mathbf{1.00}]$
	Cost	4.25 ± 16.48 $[0.00, 12.76]$	$\mathbf{0.10 \pm 0.39}$ $[\mathbf{0.00}, \mathbf{0.31}]$	4.05 ± 4.19 $[2.25, 6.23]$	5.19 ± 13.63 $[0.00, 12.22]$	$\mathbf{0.00 \pm 0.00}$ $[\mathbf{0.00}, \mathbf{0.00}]$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the scope as guided state splitting for localized signal conflict and summarize the constrained-RL and offline-RL evaluation settings. The claims are matched to the theoretical properties in Appendix A, the empirical results in Section 6, and the stated limitations in the Limitations section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section. It discusses the dependence on useful guidance labels, the reduced benefit when conflicts are global rather than localized, and the current restriction to GBT policies and structured observations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix A states each proposition with its assumptions and provides complete proofs. The results are algebraic properties of the Split-RL scoring and leaf aggregation rules and are cross-referenced from the main method section.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 5–6 describe the experimental domains, baselines, evaluation metrics, and aggregation over seeds. Appendices B.1–C provide environment definitions, label and cost rules, dataset construction details, training budgets, hyperparameters, and seed counts.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general.
- Releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The Split-RL code repository is found at `https://github.com/NVlabs/split_rl`. The code release provides the Split-RL implementation, experiment configurations, and instructions for reproducing the main constrained-RL and offline-RL experiments. For Atari offline RL, the release provides scripts that download the public expert-policy checkpoints from their original publicly available repositories and generate the rollout datasets locally; third-party checkpoint weights and generated Atari transition files are not redistributed.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the experiment design, baselines, metrics, and evaluation protocol. Appendix C reports training steps, batch sizes, discount and GAE parameters, learning rates, tree depths, seeds, and how hyperparameters were selected.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables report mean ± standard deviation across seeds, with the source of variability stated in the captions, and 95% percentile bootstrap confidence intervals computed over seeds with 2,000 resamples. Constrained-RL results are aggregated over 15 seeds. Atari results use one training run per game and dataset size, evaluated over 50 episodes (10 episodes on each of 5 evaluation seeds), and the main normalized summary reports variation across games after normalization.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it.
- The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or plots symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments were run on an H100-class GPU cluster. Individual constrained-RL runs used the training budgets and 15-seed protocol reported in Appendix C; MiniGrid and SUMO-RL runs took a few hours, while CityLearn runs took approximately 24 hours. Atari RAM offline runs used the same cluster with method-dependent budgets of 50K–500K gradient steps. Preliminary tuning used Bayesian search on a single seed before fixing hyperparameters across evaluation seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We reviewed the NeurIPS Code of Ethics. The work studies RL algorithms using simulated environments and offline benchmark data, and does not involve human subjects, private data, or deployment of high-risk models.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Split-RL is a methodological RL contribution evaluated in simulation and offline benchmark settings. The paper does not deploy trained controllers, release a user-facing model, use human-subject or private data, or make application-specific deployment claims; any societal impact would depend on a separate downstream deployment outside the scope of the paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation.
- On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk pretrained models, image generators, scraped datasets, or other assets with an obvious high risk for misuse. The evaluated environments and offline benchmarks are standard simulation or control benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The existing software, environment, and model assets are credited in the main text, appendix, and references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The submission does not introduce a standalone new dataset, pretrained model, or benchmark suite as a primary contribution. The custom scenarios and label/cost rules are experimental implementation components documented in Appendices B.1–B.3 and included with the code release; they are not presented as a separate asset release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as part of the core methodology or experiments. They were used to review the paper for accuracy and correctness, including checking mathematical derivations, helping formulate and edit paragraphs, checking spelling and grammar, and assisting with the illustration in Figure 1. All content and suggested changes were reviewed and verified by the authors.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to the LLM policy in the NeurIPS handbook for what should or should not be described.