Hi-CoLA: High Confidence Lower Bound Approximation Based Reinforcement Learning for Flex-Route Transit Operation Control

***Abstract*—Reinforcement Learning (RL) has demonstrated impressive empirical success, yet its adoption in safety- and cost-critical domains remains limited by a fundamental gap: trained policies lack statistically certified performance guarantees prior to deployment. Consequently, safety concerns arise when deploying RL in real-world environments, leading to a growing demand for safe RL algorithms. In this work, we propose the High Confidence Lower Bound Approximation (Hi-CoLA) framework, a confidence-integrated learning framework for decision-making in safety and cost critical environments. Specifically, we leverage behavioral cloning to transform rule-based decision-making processes into a parameterized policy network, and further employ Hi-CoLA to robustly improve the confidence lower bound and the overall performance of baseline policy toward optimality with real-world deployment performance guarantee. As a result of this robust training process, the framework is well-suited for real-world deployment. We compare the performance of Hi-CoLA with state-of-the-art safe RL and offline RL approaches in the context of Flex-route Transit (FRT), an intelligent demand-responsive transit system who requires real-time dynamic routing. Our approach enhances real-time control of FRT with guaranteed performance and is broadly applicable to other safety critical decision-making scenarios.**

I. Introduction

REINFORCEMENT learning (RL) has achieved notable success in domains such as robotics, games, and control by learning policies through interaction with complex environments [1]. After training, RL policies make decisions with minimal computation, making them well-suited for real-time control in dynamic systems. However, its deployment in real-world decision-making systems remains limited [2], particularly in scenarios involving financial or human safety risks since the learned policies lack statistically certified performance guarantees prior to deployment. This issue is particularly severe in safety and cost critical settings, where policy failures are unacceptable and online exploration is dangerous.

A common paradigm in such settings is to initialize policies via imitation learning from behavioral data. While these policies are safe, they are typically suboptimal, motivating the problem of safe policy improvement from offline data [3]. However, improving a behavioral policy without environment interaction imposes challenges due to distributional shift and unreliable off-policy evaluation. Existing approaches in offline and safe RL address this through pessimism, constraints, or risk-sensitive objectives, but they rarely provide explicit, high-confidence performance guarantees suitable for deployment. High-confidence off-policy evaluation (HCOPE) [4] offers a principled approach to estimating performance lower bounds with probabilistic guarantees, but it is computationally expensive and not directly amenable to gradient-based policy optimization.

In this work, we propose the High Confidence Lower Bound Approximation (Hi-CoLA) framework, which integrates confidence level with reinforcement learning training for safe policy improvement from behavioral data. Starting from a policy obtained via imitation learning, Hi-CoLA leverages importance sampling and concentration inequalities to estimate a high-confidence lower bound on policy performance. To enable efficient policy optimization, we introduce a differentiable surrogate model that approximates this lower bound as a function of policy parameters. This allows us to directly optimize policies with respect to their confidence-certified performance using gradient-based methods, resulting in monotonic improvement with performance guarantees.

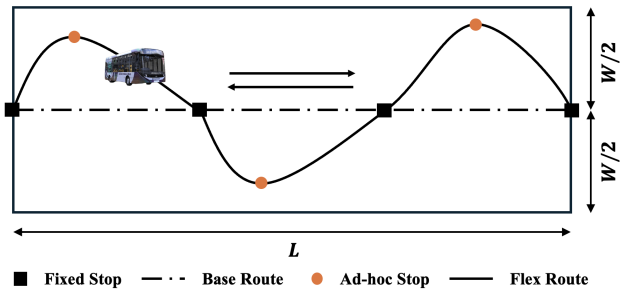


Fig. 1: Illustration of Flex-Route Transit System

To validate its effectiveness, we apply Hi-CoLA to a dynamic and stochastic decision-making problem inspired by flexible-route transit (FRT) systems [5], as illustrated in Fig. 1. This setting represents a realistic operational environment characterized by stochastic demand and limited tolerance for failure. The adoption of such systems is constrained by the inflexibility of rule-based vehicle routing methods [6] [7]. Experimental results demonstrate that Hi-CoLA achieves consistent performance improvements while maintaining high-confidence guarantees, outperforming state-of-the-art safe and offline RL baselines.

The contributions of this paper are summarized as follows: (1) We propose a reinforcement learning framework for real-time operation control of safety and cost critical applications, which encodes a quantified confidence level into the training pipeline. This enables performance guarantees for the trained policy and supports deployment in real-world settings. (2) The Hi-CoLA network can be used to estimate the performance lower bound of any given policy with a user-defined confi-

dence level. (3) The proposed framework reduces the reliance on constructing a detailed simulation environment and shows strong potential in partially observable and high-dimensional environments.

II. Related Work

In this section, we briefly review related work on the FRT system control and introduce reinforcement learning approaches, which concern the safety of the policy. For comprehensive literature reviews, readers may refer to [5] for FRT systems and [3], [8] for safe and offline reinforcement learning, respectively.

A. Real-time operations control for FRT

FRT extends the accessibility of traditional fixed route transit by allowing route deviations for dynamic passenger requests, and has emerged as a promising hybrid solution [5]. To date, researchers have developed heuristic-based [6] and optimization-based [9] approaches for FRT real-time operation control. [10] first examined dynamic FRT using a schedule-insertion heuristic under simplified settings, with later extensions to multi-vehicle coordination with dynamic demand [11] and re-routing with real-time traffic via Google Maps [12]. [13] formulates a Mixed Integer Linear Programming (MILP) for pre-booked demand and then inserts on-the-fly requests, subject to capacity and tolerance on schedule adherence. Furthermore, [14] extends this approach to complex settings with random request cancellations. While these approaches have laid the groundwork, they present notable trade-offs between accuracy and computation efficiency, limiting their applicability for dynamic environments [15]. In contrast, reinforcement learning (RL) learns reactive deviation policies from real-time state and shows consistent gains over heuristic baselines in flex-route operations [16].

B. Safe Reinforcement Learning

The ultimate goal of safe reinforcement learning (SRL) is to make RL dependable enough for deployment in safety-critical settings; however, because "safety" has multiple formal meanings in the literature, research has diverged into distinct paths. One widely adopted definition, summarized in the comprehensive survey [8], frames SRL as the process of learning policies that maximize the expected return while ensuring reasonable system performance and/or respecting safety constraints during both the learning and deployment phases. This view often uses constrained MDP formulations, with methods such as Constrained Policy Optimization (CPO) [17], Lyapunov-based RL [18], shielding [19], and reachability-based safe exploration [20]. Risk-sensitive RL, which optimizes tail-aware criteria [21] (e.g., CVaR and coherent risks), and robust MDPs [22], which optimize worst-case value over ambiguity sets for the dynamics, also fall in this category. Another perspective defines safety as robustness to arbitrary off-policyness. For instance, Retrace(λ) [23] remain stable and low-variance even when the behavior and target policies differ substantially, making them suitable for heterogeneous or logged datasets. Closely related, offline reinforcement learning such as IQL [24] and CQL [25] focuses on learning from fixed datasets without interaction risks, but distributional shift and unreliable off-policy evaluation pose central challenges.

The definition of safety adopted by this work is from [26]. Here, safety is treated as a guarantee against regression: given a baseline ρ and tolerance δ, the algorithm should return a policy whose expected performance falls below ρ with probability of at most δ. Methods such as SPIBB [27] and High-Confidence Off-Policy Evaluation (HCOPE) [4] enforce this by combining lower confidence bounds with conservative updates, ensuring no performance regression at deployment. Beyond these, related progress in offline RL strengthens such guarantees via conservatism. Notably, [28] provides lower-bound value guarantees and monotonic improvement tendencies from static datasets; theoretical work shows that pessimistic value iteration [29] can be provably efficient under realistic coverage assumptions.

Prior safe RL work either enforces constraints without certifying performance at deployment or targets off-policy stability without guarantees. Even the few methods that do provide explicit statistical guarantees can be too conservative to realize improvements. We introduce Hi-CoLA, an offline framework that learns a high-confidence lower bound on policy performance from real-world operations data and directly maximizes that performance lower bound. This closes the deployment gap while avoiding simulator dependence and enables a simple and automatic cycle of log, improve, and deploy for FRT system.

III. Problem Formulation

In this section, we present the mathematical problem formulation of the Flex-Route Transit real-time operation control problem. We first introduce the system description of the Flex-Route Transit system and then formulate it under the Partially Observed Markov Decision Process (POMDP) framework so that we can solve it from a reinforcement learning perspective. Table I presents a list of major notations used throughout the paper. In this paper, we adopt the convention of using bold symbols to represent a Matrix or a vector and non-bold symbols to represent a scalar or an element out of a vector.

A. Flex-Route Transit Operation System Description

The Flex-Route Transit Operation is formulated as a Vehicle Routing Problem with Pick-up and Drop-off (VRPPD). The objective is to serve passenger requests at both fixed and flexible stops within a designated service area by dynamically dispatching and coordinating a fleet of vehicles. The service area is modeled as a rectangle of dimensions $W \times L$, bounded by two terminal checkpoints. The routing network is represented by a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of all the potential stops, and $\mathcal{E}$ is the set of edges connecting stops. Fixed stops along the base route, which are mandatory to be visited, are denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ with terminals c_1 and c_n; flexible stops located between consecutive fixed stops are denoted as $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$.

TABLE I: List of Major Notations Used in the Paper

Notations	Descriptions
$W \times L$	dimensions of service area
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	the routing network (represented as a graph)
$\mathcal{V}$	set of total stops, $\lvert\mathcal{V}\rvert = N$
$\mathcal{C}$	set of fixed stops, $\lvert\mathcal{C}\rvert = n$
$\mathcal{F}$	set of flexible stops, $\lvert\mathcal{F}\rvert = m$
$\mathcal{E}$	set of directed edges connecting stops
i, j	node index, $v_i, v_j \in \mathcal{V},\ i, j \in \{1, \ldots, N\}$
$e_{i,j}$	edge fromv_i pointing to v_j, $i \neq j$
$t_{i,j}$	travel time on $e_{i,j}$
μ_t, σ_t	travel time sampling parameters
$\boldsymbol{T}$	travel time matrix
K	# of vehicles
Q_c	vehicle capacity
h	departure interval
$W_{\text{fixed}}, W_{\text{flex}},$	waiting time threshold at fixed and flexible stops
β_1	passenger no-show rate
Q_t^p	passenger arrival matrix
Q_t^o	vehicle location matrix
π	FRT operation control policy
$\boldsymbol{\theta}$	policy parameters
H	trajectory length
c_p	size of perturbation space
β_{kl}	KL-divergence threshold
J	objective of 's module
α	learning rate of 's module

Then $\mathcal{V}$ can be expressed by $\mathcal{V} = \mathcal{C} \cup \mathcal{F}$ and $v_i = c_i$ and $v_{i+n} = f_i$. The directed edge from node v_i pointing to node v_j is denoted as $\{e_{i,j} = (v_i, v_j) |\ \forall v_i, v_j \in \mathcal{V},\ i \neq j\}$ and the edges set $\mathcal{E}$ can be partitioned as $\mathcal{E} = \mathcal{E}_{\text{B}} \cup \mathcal{E}_{\text{D}}$, where $\mathcal{E}_{\text{B}}$ includes all base edges connecting fixed stops and $\mathcal{E}_{\text{D}}$ includes all deviate edges that connect flexible stops and fixed stops. Due to the deviation nature of the system, deviate edges do not geometrically align with base edges.An illustration graph of overall setting is given in Fig. 2

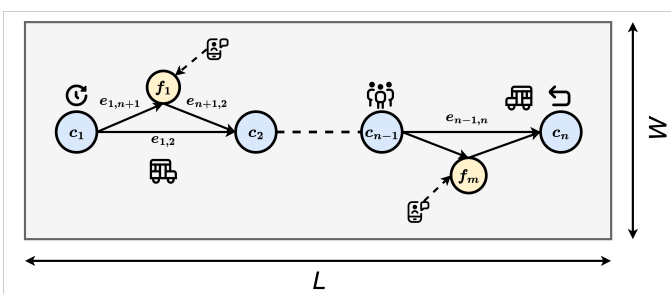


Fig. 2: Flex-Route Transit Operation System Description

Each edge $e_{i,j} \in \mathcal{E}$ is assigned a travel time denoted by $t_{u,w}$. For base edges $e_{i,j} \in \mathcal{E}_{\text{B}}$, the travel time is a random variable with a norm distribution to model the dynamic and stochastic traffic conditions, i.e., $t_{i,j} \sim \mathcal{N}(\mu_t, \sigma_t^2)$, where μ_t and σ_t are chosen to represent different congestion levels. For deviate edges $e_{i,j} \in \mathcal{E}_{\text{D}}$, travel time $t_{i,j}$ is set to be a constant t_{D}. Then, the travel times are defined as $T = \{t_{i,j} |\ (i, j) \in \mathcal{E}\}$.

Our Flex-Route Transit service operates K homogeneous vehicles, each with a capacity of Q_c. Vehicles are dispatched sequentially from stop c_1 at intervals of h minutes. If all vehicles have been dispatched and none have returned, the system must wait until a vehicle reaches terminal c_n and returns to the origin stop c_1 to initiate the next departure. Passengers comprise (i) regular riders arriving at fixed stops $\mathcal{C}$, and (ii) on-demand riders requesting pick-up at flexible stops $\mathcal{F}$. On-demand passengers make a request in advance or just walk to an on-demand stop. Requests are not proactively rejected, but expire if waiting exceeds a threshold of W_{fixed} at $\mathcal{C}$ or W_{flex} at $\mathcal{F}$. The pre-booked deviation requests may not show up with probability β_1. The number of passengers waiting at all stops at time t is represented by the set $Q_t^p = \{q_m(t) |\ \forall m \in \mathcal{V}\}$.

At any time t, vehicles are either dwelling at a stop or traversing a directed edge. A binary *occupancy matrix* $Q_t^{\text{o}} \in \{0, 1\}^{N \times N}$ is defined to track vehicles' locations at each time t. For all stops $v_i \in \mathcal{V}$:

$$Q_t^{\text{o}}(i, i) = \begin{cases} 1 & \text{if stop } v_i \text{ is occupied,} \\ 0 & \text{otherwise,} \end{cases}$$

And for all $e_{i,j} \in \mathcal{E}$

$$Q_t^{\text{o}}(i, j) = \begin{cases} 1 & \text{if edge } e_{i,j} \text{ is occupied,} \\ 0 & \text{otherwise} \end{cases}$$

Vehicles make routing decisions only upon arrival at control stops, which is a subset of $\mathcal{C}$, to decide whether to proceed along the *base* edge or detour via the corresponding *deviation* stop. The objective is to select deviations to serve more requests.

B. Real-time Operation Control

Since we treat the Flex-Route Transit operation control from a reinforcement learning (RL) perspective, we establish it within the Partially Observed Markov Decision Process (POMDP) framework. A POMDP can be described as $\langle S, A, \mathsf{P}, R, \Omega, O, \gamma \rangle$. S indicates the set of states, A refers to the action set for agent, R is the reward, Ω is the set of observations for agent, P and O are state transition probabilities and observation probabilities, respectively, γ is the discount factor. In the context of FRT real-time operation control, we treat the whole system as the control agent, and adaptively control the deviation decision at each control point as well as the dispatching decision for the fleet. The travel time matrix T, Passenger distribution Q_t^p, vehicle location Q_t^o, and the number of idle vehicles at the starting point c_1, compose the agent observation Ω. Action A consists of deviation decisions at each control point and a binary departure decision. The action is updated based on a fixed resolution and vehicle arrival at the fixed stops. The discount factor $\gamma = 1$; the instantiated reward r_t is set to be the served passengers at each timestep. Let $\pi(a|s)$ be the probability (density) of taking action a in state s when using policy π. It is denoted as $\pi(\boldsymbol{\theta})$ if it can be parametrized by vector $\boldsymbol{\theta}$. The objective is to find the policy $\pi(\boldsymbol{\theta})$, which maximizes the cumulative reward.

IV. Methodology

This section presents our High Confidence Lower Bound Approximation (Hi-CoLA) based reinforcement learning

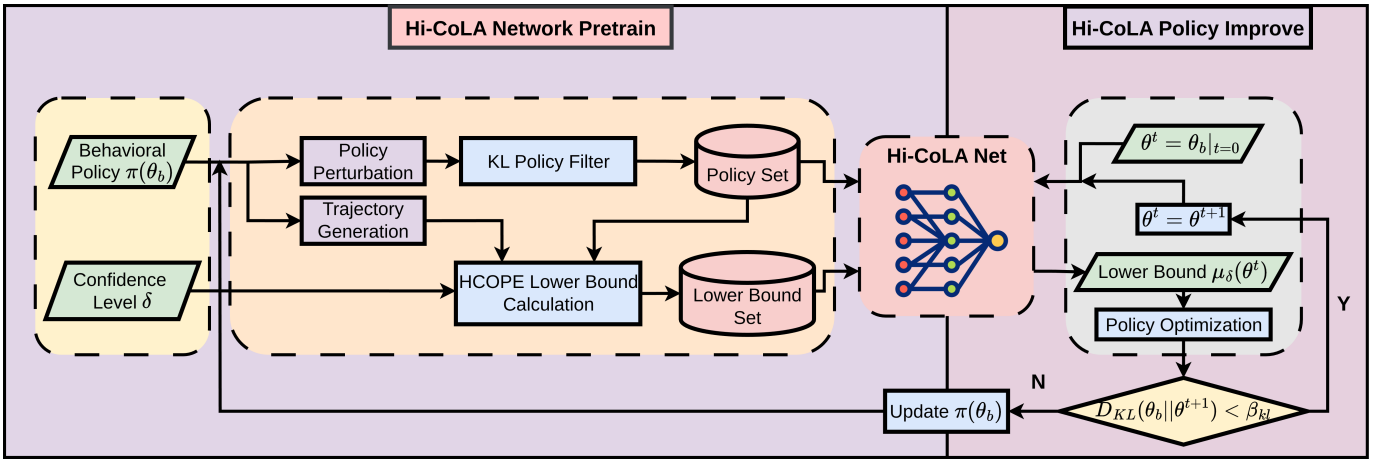


Fig. 3: Framework and the abstract algorithm flow process

framework for reliable and efficient policy improvement for the real-time control of the FRT system. As background, we first introduce high-confidence off-policy evaluation (HCOPE) [4]. It employs importance sampling and concentration inequalities to estimate the lower-bound performance of a given policy at a quantified confidence level. Building on this foundation, we propose the Hi-CoLA framework to enable systematic policy refinement while ensuring that operational performance improves under a user-defined confidence guarantee.

A. Confidence Guaranteed Policy Performance Evaluation

We assume that the FRT real-time operation control problem can be modeled as a partially observable Markov decision process (POMDP). The operating process can be represented by a trajectory of length H, which is an ordered set of states, actions, and rewards as we defined previously:

$$\tau = \{s_1^\tau, a_1^\tau, r_1^\tau, s_2^\tau, a_2^\tau, r_2^\tau, \ldots, s_H^\tau, a_H^\tau, r_H^\tau\}.$$

The normalized and discounted return of a trajectory is defined as :

$$R(\tau) = \frac{\left(\sum_{t=1}^{H} \gamma^{t-1} r_t^\tau\right) - R_-}{R_+ - R_-} \in [0,1] \tag{1}$$

where R_- and R_+ are the theoretical lower and upper bounds on the cumulated return for each trajectory in the environment. And the bounded performance of the policy π parameterized by $\boldsymbol{\theta}$ is denoted by

$$\rho(\pi(\boldsymbol{\theta})) = \mathbb{E}[R(\tau)|\pi(\boldsymbol{\theta})]$$

We deploy a behavioral policy $\pi(\boldsymbol{\theta_b})$ to generate a set of n trajectories and store them into $\mathcal{D}$ i.e.,

$$\mathcal{D} := \{\tau_i | i \in \{1, \ldots, n\}, \text{ generated by } \pi(\boldsymbol{\theta_b})\}.$$

Then we can use *High-confidence off-policy evaluation* (HCOPE) method proposed by [4] to accurately approximate the confidence lower-bound of the performance of a given policy, $\pi(\boldsymbol{\theta_e})$, also called the evaluation policy, with the generated trajectory set $\mathcal{D}$ and behavioral policy $\pi(\boldsymbol{\theta_b})$. The HCOPE approach is based on *importance sampling*, which can be used to produce an unbiased estimator of $\rho(\pi(\boldsymbol{\theta_e}))$ from each trajectory $\tau_i \in \mathcal{D}$. The proof can be found in [4]. This estimator is denoted by $\hat{\rho}(\boldsymbol{\theta_e}, \tau_i, \boldsymbol{\theta_b})$, and is given by

$$\hat{\rho}(\boldsymbol{\theta_e}, \tau_i, \boldsymbol{\theta_b}) = R(\tau_i)\frac{Pr(\tau_i|\boldsymbol{\theta_e})}{Pr(\tau_i|\boldsymbol{\theta_b})} = R(\tau_i)\prod_{t=1}^{H}\frac{\pi(a_t|s_t, \boldsymbol{\theta_e})}{\pi(a_t|s_t, \boldsymbol{\theta_b})} \tag{2}$$

The estimators are then used to produce a high-confidence lower bound on the performance of an evaluation policy with a special form of concentration inequality theorem, which is independent of the range and distribution of the estimators. Let $\hat{\mu}_\delta(\theta_e)$ be a prediction, computed from $\{\hat{\rho}(\theta_e, \tau_i, \theta_b)| \ \forall \tau_i \in \mathcal{D}\}$, of the $(1-\delta)$ confidence lower bound on $\mathbb{E}[R(\tau)|\pi(\boldsymbol{\theta_e})]$. The predictor is

$$\hat{\mu}_\delta(\boldsymbol{\theta_e}) = \frac{1}{n}\sum_{i=1}^{n} M_i - \frac{7c\ln(2/\delta)}{3(n-1)} - \sqrt{\frac{2\ln(2/\delta)}{n^2(n-1)}\left[n\sum_{i=1}^{n} {M_i}^2 - \left(\sum_{i=1}^{n} M_i\right)^2\right]} \tag{3}$$

where M_i is shorthand for $\min\{\hat{\rho}(\boldsymbol{\theta_e}, \tau_i, \boldsymbol{\theta_b}), c\}$.

To obtain a tight predictor $\hat{\mu}_\delta(\boldsymbol{\theta_e})$ of the $(1-\delta)$ confidence lower bound, a small subset of the data can be used to choose the optimal $c^\star$ with bi-section selection.

$$c^\star \in \arg\max_{c \in [1,\infty)} \hat{\mu}_\delta(\boldsymbol{\theta_e}),$$

B. Hi-CoLA based Policy improvement

The previous subsection discusses the background of confidence-integrated policy performance evaluation. In this subsection, we extend the philosophy to policy improvement by introducing the High Confidence Lower Bound Approximation-based reinforcement learning framework (Hi-CoLA). It is designed to deliver performance-guaranteed policy improvement for real-time control of Flex-Route Transit

(FRT) operations. Hi-CoLA integrates HCOPE into the policy training process by using a differentiable surrogate, namely the Hi-CoLA network, which enables gradient-based policy optimization. The overall policy training scheme is illustrated in Figure 3 and includes two key components: Hi-CoLA network training and gradient-based policy improvement.

1) Hi-CoLA Network Training: Although HCOPE provides a tight confidence lower bound, it is computationally expensive and non-differentiable, making it impractical for iterative policy updates. In this work, we treat the modeling process from policy parameters θ to the corresponding performance confidence lower-bound$\hat{\mu}_\delta(\boldsymbol{\theta})$ as a machine learning problem, and establish the Hi-CoLA network, which takes the shape of multi-layer perceptron (MLP) i.e., $\hat{\mu}_\delta(\boldsymbol{\theta}) = mlp(\boldsymbol{\theta})$. This differentiable proxy amortizes the computational cost of HCOPE and enables backpropagation-based gradient optimization in a trusted region.

To generate training data for the Hi-CoLA network, we first execute a behavioral policy $\pi(\boldsymbol{\theta_b})$ that satisfies the basic service requirements of the system and collect its trajectories in the real world in a dataset $\mathcal{D}$ for confidence bound calculation. Around this behavioral policy, we create a set of candidate policies $\pi(\boldsymbol{\theta_p})$ by applying parameter-scaled perturbations with Eq. 4, where c_p is a constant to adjust the size of perturbation space, and $\theta^{(i)}$ refers to the indexed parameter in the respective $\boldsymbol{\theta}$.

$$\theta_p^{(i)} = \theta_b^{(i)} + \epsilon_i, \text{ where } \epsilon_i \sim \mathcal{N}(0, c_p \cdot |\theta_b^{(i)}|) \tag{4}$$

To ensure the importance sampling remains stable, we filter these candidates using a KL-divergence threshold β_{kl}, retaining only those that stay sufficiently close to the behavioral policy, where:

$$D_{\mathrm{KL}}(\pi_b \parallel \pi_p) = \frac{1}{N} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \pi_b(a,s) \log \left(\frac{\pi_b(a,s)}{\pi_p(a,s)} \right) \leq \beta_{kl}$$

With Eq. 3, we can calculate the tight confidence lower bound $\hat{\mu}_\delta(\theta_p)$ for each accepted policy $\pi(\boldsymbol{\theta_p})$, and these policy–lower bound pairs form the supervised training data for the Hi-CoLA network. The detailed training process is delivered in Algorithm 1.

2) Hi-CoLA Based Policy Improvement: Once trained, the Hi-CoLA network acts as a fast and differentiable surrogate for estimating confidence lower bounds near the behavioral policy. Using backpropagation, we compute the gradient of $\hat{\mu}_\delta(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, *i.e.*, $\partial\hat{\mu}_\delta(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$. Then we can update the policy parameters with the stochastic gradient descent method with the objective $J = (1-\mu)^2$ since the performance is bounded by 1 (Eq. 1):

$$\theta^{t+1} = \theta^t - \alpha \nabla_\theta (1 - \hat{\mu}(\theta))^2 \tag{5}$$

This iterative process continues until the updated policy drifts beyond the safe KL-divergence threshold β_{kl}. Upon convergence, the resulting policy achieves improved performance with a certified confidence guarantee. This policy can then be deployed in the field or used as a new baseline behavioral policy for further refinement. This approach preserves the safety and interpretability of HCOPE while substantially reducing computational overhead, enabling more responsive and reliable policy optimization.

Algorithm 1: Training of Hi-CoLA Network

Input: Behavioral Policy $\pi(\boldsymbol{\theta_b})$; Confidence level δ; # of Trajectory n; Perturbation Size N;
Output: Well-trained Hi-CoLA Network $\hat{\mu}_\delta(\boldsymbol{\theta}) = mlp(\boldsymbol{\theta})$
Init: Hi-CoLA Network: $\hat{\mu}_\delta(\boldsymbol{\theta}) = mlp(\boldsymbol{\theta})$; Trajectory Dataset $\mathcal{D}_t = \emptyset$; Policy Set $\mathcal{D}_\theta = \emptyset$; Lower-Bound Set $\mathcal{D}_\mu = \emptyset$; $i, j = 0$

while $i \leq n$ **do**
  Generate trajectory τ_i using $\pi(\theta_b)$;
  $\tau_i = \{s_1^{(i)}, a_1^{(i)}, r_1^{(i)}, \ldots, s_H^{(i)}, a_H^{(i)}, r_H^{(i)}\}$;
  Store τ_i in $\mathcal{D}$;
  $i \leftarrow i + 1$;
while $j \leq N$ **do**
  foreach $i \in \{1, 2, \ldots, |\boldsymbol{\theta_b}|\}$ **do**
    $\theta_j^{(i)} = \theta_b^{(i)} + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, c_1 \cdot |\theta_b^{(i)}|)$;
  if $D_{KL}(\boldsymbol{\theta_j} || \boldsymbol{\theta_b}) \leq \beta_{kl}$ **then**
    Store $\boldsymbol{\theta_j}$ in $\mathcal{D}_\theta$;
    Calculate $\hat{\mu}_\delta(\boldsymbol{\theta_j})$ using Eq. 3 with trajectories $\mathcal{D}$;
    Add $\hat{\mu}_\delta(\boldsymbol{\theta_j})$ to $\mathcal{D}_\mu$;
    $j \leftarrow j + 1$
Compile $\mathcal{D}_{\text{train}} = \{(\theta_1, \mu_1), (\theta_2, \mu_2), \ldots, (\theta_N, \mu_N)\}$;
Train $\hat{\mu}_\delta(\boldsymbol{\theta}) = mlp(\boldsymbol{\theta})$ with $\mathcal{D}_{\text{train}}$;
Final.

V. Experiments and Analysis

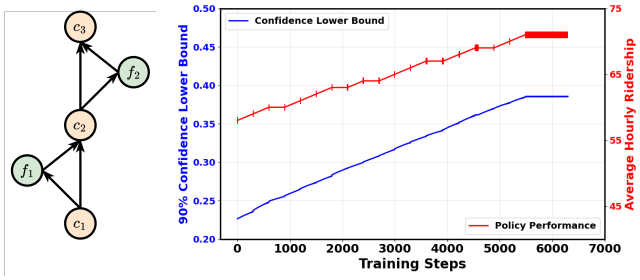
In this section, we benchmark the Hi-CoLA performance for FRT real-time operation control with state-of-the-art safe and offline RL algorithms. For the state-of-the-art, we choose (1) Implicit Q-learning (IQL) [24]; (2) Conservitive Q-learning(CQL) [25]; (3) TD3 with behaviour cloning [30]; (4) Daedalus [26]. Notably, IQL, CQL, TD3+BC are offline RL, and Daedalus is safe RL algorithm developed to improve policy with confidence The main hyperparameter for selected algorithms, as well as Hi-CoLA, are configured as follows: (1) Hi-CoLA: $\alpha = 5 \times 10^{-5}$, $\delta = 0.1$; (2) IQL: $\alpha = 1 \times 10^{-4}$; (3) CQL:: $\alpha = 2 \times 10^{-4}$; (4) TD3+BC: $\alpha = 2 \times 10^{-4}$; (5) Daedalus: $\delta = 0.1$. All algorithms are implemented in Python 3.9 with publicly available source code, and the experiments are conducted on a server equipped wth a 16-core, 32 threads CPU, 64GB RAM, NVIDIA GTX TiTan XP GPU and the 64-bit Ubuntu system.

In the following subsections, we first present a simple yet illustrative scenario to showcase Hi-CoLA's capability of guaranteeing monotonic policy improvement, and then benchmark

the performance of the Hi-CoLA algorithm with the state of the art in terms of performance distribution and CVaR.

A. Performance Improvement Guarantee Illustration

This subsection demonstrates the performance improvement guarantee of the Hi-CoLA training framework in a simplified FRT environment with 3 fixed stops and 2 flexible stops in between. The topology of the route and service direction is shown in Fig.4(a). For each trip, the vehicle starts at node



(a) Route Map (b) Training Process and Policy Improvement

Fig. 4: Training process

c_1 and makes the deviation decisions when it arrives fixed stops c_1 and c_2. Passenger arrivals are sampled from a Poisson distribution with an average arrival rate of 18 pax/min for fixed stops and 10 pax/min for flexible stops. The maximized waiting time for each passenger is 30 minutes and vehicles are dispatched every 12 minutes. Initially, the vehicles are controlled by a rule-based behavioral policy, which has 0.6 probablity to deviate if the passenger requests at f_i are more than half of the passengers waiting at c_{i+1}; otherwise, the deviation probability is 0.2. Then, we collect 6000 pairs of observations and corresponding actions to train the policy network imitating this rule-based policy for parameterization purposes. The parameterized policy is used as the initial behavioral policy. Fig. 4(b) shows the evolution of the 90% confidence lower bound at each step and policy performance in terms of the average hourly served passengers after each training cycle. From the plot, we can see that as training progresses, the performance of the trained policy monotonically improves, and the updated behavioral policies. consistently outperform their predecessors.

B. Benchmark with State of the Arts

This subsection compares the Hi-CoLA-based FRT real-time operation policy with state-of-the-art methods in terms of average hourly ridership distribution and Conditional Value at Risk (CVaR). The evaluation environment includes five fixed stops and four flexible stops. To model the dynamics of a real transportation system, the travel time between fixed stops is randomly drawn from a uniform distribution and updated at each timestamp. Passengers waiting at flexible stops will have 0.2 probablity to cancel the ride and the maximized waiting time is 15 minutes. The passenger arrivals are sampled with a Poisson distribution with an average arrival rate of 9.6 pax/hour for fixed stops and 7.5 pax/hour for flexible stops.

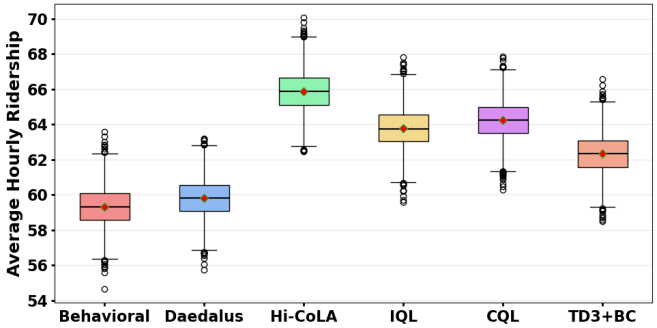


Fig. 5: Performance Distribution Comparison

TABLE II: Hourly Ridership CVaR Comparison

Method	$\text{CVaR}_{5\%}$	$\text{CVaR}_{10\%}$	$\text{CVaR}_{5\%}$ Improve	$\text{CVaR}_{10\%}$ Improve
Behavior	57.07	57.41	0.00%	0.00%
IQL	61.39	61.75	7.57%	7.56%
CQL	61.95	62.28	8.55%	8.48%
TD3+BC	59.97	60.35	5.08%	5.12%
Hi-CoLA	**63.52**	**63.88**	**11.30%**	**11.27%**
Daedalus	57.51	57.86	0.77%	0.78%

The results in Fig. 5 and Table II demonstrate that Hi-CoLA consistently outperforms all baselines in both average performance and tail-risk metrics. From the distributional perspective, Hi-CoLA achieves the highest median and mean hourly ridership, while also exhibiting a tighter spread compared to other methods, indicating improved stability. More importantly, in terms of risk-sensitive evaluation, Hi-CoLA attains the best CVaR performance, achieving 63.52 and 63.88 for $\text{CVaR}_{5\%}$ and $\text{CVaR}_{10\%}$, respectively. The corresponding improvement over the behavioural policy outperforms offline RL baselines such as IQL, CQL, and TD3+BC. These results highlight that Hi-CoLA not only improves average performance but also substantially enhances worst-case outcomes.

VI. CONCLUSION

This paper propose Hi-CoLA, a High-Confidence Lower Bound Approximation-based reinforcement learning framework that enables policy improvement with quantifiable confidence guarantees, addressing the key challenges of stochasticity and sim-to-real transfer that hinder the deployment of conventional reinforcement learning in cost or safety-critical scenarios. The resukts show that Hi-CoLA consistently improves policy performance while maintaining safety, outperforming state-of-the-art offline and safe RL algorithms.

Overall, Hi-CoLA offers a reliable and practical framework for deploying reinforcement learning in cost- and safety-critical systems. Future work may explore improving the sample efficiency and performance variance confidence estimation problems.

REFERENCES

[1] Y. Tong, J. Wang, Q. Du, X. Zhang, and J. Yu, "A survey on reinforcement learning methods in bionic underwater robots," *Biomimetics*, vol. 8, no. 1, p. 33, 2023.

[2] G. Dulac-Arnold, D. Mankowitz, T. Hester, *et al.*, "Challenges of real-world reinforcement learning," in *NeurIPS 2021 Workshop on Real-World Reinforcement Learning*, 2021.

[3] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10 237–10 257, 2024.

[4] P. S. Thomas, G. Theocharous, and M. Ghavamzadeh, "High-confidence off-policy evaluation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*. AAAI Press, 2015, pp. 3000–3006.

[5] R. Shahin, P. Hosteins, P. Pellegrini, P.-O. Vandanjon, and L. Quadrifoglio, "A survey of flex-route transit problem and its link with vehicle routing problem," *Transportation Research Part C: Emerging Technologies*, vol. 158, p. 104437, 2024.

[6] W. Lu, L. Lu, and L. Quadrifoglio, "Scheduling multiple vehicle mobility allowance shuttle transit (m-mast) services," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 125–132.

[7] R. G. Farwell and E. Marx, "Planning, implementation, and evaluation of omniride demand-driven transit operations: Feeder and flex-route services," *Transportation Research Record*, vol. 1557, no. 1, pp. 1–9, 1996.

[8] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015.

[9] J. Zhang, W. Li, Y. Zheng, and R. Guo, "Dynamic clustering meeting points strategy to improve operational service capability of flex-route transit," *Journal of Transportation Engineering, Part A: Systems*, vol. 149, no. 6, p. 04023038, 2023.

[10] L. Quadrifoglio, M. M. Dessouky, and K. Palmer, "An insertion heuristic for scheduling mobility allowance shuttle transit (MAST) services," *Journal of Scheduling*, vol. 10, no. 1, pp. 25–40, Feb. 2007.

[11] W. Lu, L. Lu, and L. Quadrifoglio, "Scheduling multiple vehicle mobility allowance shuttle transit (m-mast) services," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 125–132.

[12] F. Qiu, W. Li, and C. An, "A google maps-based flex-route transit scheduling system," in *CICTP 2014: Safe, Smart, and Sustainable Multimodal Transportation Systems*. Reston, VA: American Society of Civil Engineers, June 2014, pp. 247–257.

[13] Y. Zheng and W. Li, "Flex-route transit service with different degree of dynamism," in *Proceedings of the 19th COTA International Conference of Transportation Professionals (CICTP 2019): Transportation in China—Connecting the World*. Reston, VA: American Society of Civil Engineers, July 2019, pp. 4369–4380.

[14] Y. Zheng, L. Gao, and W. Li, "Vehicle routing and scheduling of flex-route transit under a dynamic operating environment," *Discrete Dynamics in Nature and Society*, vol. 2021, pp. 1–10, 2021.

[15] E. Cipriani, S. Gori, and M. Petrelli, "Combining its and optimization in public transportation planning: state of the art and future research paths," *EURO Journal on Transportation and Logistics*, vol. 8, no. 1, pp. 7–34, 2019.

[16] J. Rodriguez, H. N. Koutsopoulos, and J. Zhao, "Flex-route transit for smart cities: A reinforcement learning approach to balance ridership and performance," *Smart Cities*, vol. 8, no. 5, 2025.

[17] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 22–31.

[18] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[19] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[20] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 37, 2015, pp. 997–1005.

[21] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-sensitive and robust decision-making: A cvar optimization approach," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[22] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[23] R. Munos, T. Stepleton, A. Harutyunyan, and M. G. Bellemare, "Safe and efficient off-policy reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[24] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2022.

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

[26] P. S. Thomas, G. Theocharous, and M. Ghavamzadeh, "High confidence policy improvement," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, July 2015, pp. 2380–2388.

[27] R. Laroche, P. Trichelair, and R. Tachet des Combes, "Safe policy improvement with baseline bootstrapping," in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 3652–3661.

[28] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[29] T. Xie, C.-A. Cheng, N. Jiang, P. Mineiro, and A. Agarwal, "Bellman-consistent pessimism for offline reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[30] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 20 132–20 145.