

Vehicle Lane-Change Trajectory Pattern Analysis and Policy Learning in Highway Driving Scenarios

Hyeonsu Kim*, Yiwen Shen*

* Department of Software and Computer Engineering,
Ajou University,
Suwon, Republic of Korea
hskim5851@ajou.ac.kr, chrisshen@ajou.ac.kr

Abstract—This study presents an analysis of lane change patterns based on real-world driving data, combined with behavioral comparison using reinforcement learning policies. Specifically, we implement an Offline Reinforcement Learning (Offline RL) approach using Implicit Q-Learning (IQL) to safely derive an optimized lane change policy from large-scale driving datasets. The proposed velocity-proportional reward function is tested within the multi-quality mixed dataset environment of the AD4RL benchmark. Our autonomous driving agent achieves an average normalized score of 89.03 points (on a 0-100 scale), demonstrating expert-level performance while managing the trade-off between safety and high-speed driving efficiency, even in sudden cut-in situations. The results validate the effectiveness of our approach for trajectory prediction and decision-making in complex highway driving scenarios.

Index Terms—Lane change, trajectory prediction, deep reinforcement learning, autonomous vehicles.

I. INTRODUCTION

Lane changing is a key driving maneuver that requires accurate trajectory prediction for both safety and efficiency [1]. While deep reinforcement learning has improved autonomous driving decisions [2], offline reinforcement learning still suffers from overestimation when the policy evaluates out-of-distribution actions not supported by logged data. To address this issue, we adopt Implicit Q-Learning (IQL) [3], which focuses learning on in-dataset actions to improve stability under noisy and mixed-quality data. In this study, we aim to learn lane-change utility for policy decisions, evaluate a velocity-proportional reward in the AD4RL mixed-quality benchmark [4], and develop an autonomous agent that balances safety with high-speed driving, including sudden cut-in scenarios.

To achieve the objectives, we introduce an offline RL-based approach, which safely derives an optimal policy by utilizing only large-scale driving data preprocessed to enable collection and simulation benchmarking. We use the highway environment [5] based on the Gym DRL framework for training our model. This environment simulates realistic highway driving scenarios, allowing us to test our

trajectory prediction model in a controlled setting. The main contributions of this work are summarized as follows:

- We propose a velocity-proportional reward function that encourages efficient high-speed overtaking while maintaining safety constraints, and validate its effectiveness in the AD4RL benchmark.
- We implement an autonomous driving agent using IQL to learn optimal lane change policies from mixed-quality offline datasets, demonstrating expert-level performance in trajectory prediction and decision-making.
- We conduct extensive experiments to evaluate the performance of our approach, achieving an average normalized score of 89.03 points, confirming the ability to establish an efficient high-speed overtaking policy surpassing human expert levels.

The rest of this paper is organized as follows. Section II reviews related studies on lane-change prediction and trajectory forecasting. Section III introduces reinforcement learning and Implicit Q-Learning. Section IV presents our method, Section V describes the experimental setup and reports the results, and Section VI concludes the paper.

II. RELATED WORK

Lane change prediction and trajectory forecasting have been extensively studied in the context of autonomous driving. Traditional methods rely on kinematic models and hand-crafted features to predict vehicle trajectories. However, these approaches often struggle to capture the complex and interactive nature of highway driving.

Recent work most directly related to our offline RL setting for lane-change/highway driving includes RLfOLD, which combines RL with online demonstrations to narrow the distribution gap and improve urban driving performance in CARLA NoCrash [6], and world-model approaches such as Copilot4D [7] and Trajenglish [8], which improve trajectory forecasting quality via tokenized/discrete generative modeling on large-scale datasets.

In the context of offline reinforcement learning, several algorithms address learning from fixed datasets without further environment interaction. RIQL improves resilience to state/action/reward/dynamics corruption via Huberized critic

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the National Program for Excellence in SW (2022-0-01077) supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation) in 2026. Multiple LLM models were used to polish up the manuscript. (*Corresponding author: Yiwen Shen.*)

targets and quantile-based estimation [9]. PNLSVI introduces variance-weighted regression and pessimistic planning for strong instance-dependent guarantees under nonlinear function approximation [10]. MOREC enforces reward-consistent transition selection in model-based offline RL, reporting substantial gains on D4RL and NeoRL [11]. IQL is particularly attractive because it implicitly excludes out-of-distribution actions by only evaluating actions in the training dataset, mitigating overestimation bias under noisy or mixed-quality data. Rather than simply applying IQL with the original AD4RL reward, our contribution is the velocity-proportional reward design itself and its empirical convergence/failure-mode analysis on the Cut-in scenario.

III. PRELIMINARIES

In this section, we provide a brief overview of the reinforcement learning framework and IQL algorithm, which form the foundation of our approach to lane change trajectory prediction.

A. Reinforcement Learning Framework

We model the RL task as a Markov decision process (MDP), $(\mathcal{S}, \mathcal{A}, p_0(s), p(s'|s, a), r(s, a), \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $p_0(s)$ is the initial-state distribution, $p(s'|s, a)$ defines the transition dynamics, $r(s, a)$ is the reward function, and γ is the discount factor. The agent follows a policy $\pi(a|s)$, and the objective is to learn a policy that maximizes cumulative discounted return:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim p_0(\cdot), \right. \\ \left. a_t \sim \pi(\cdot | s_t), s_{t+1} \sim p(\cdot | s_t, a_t) \right]. \quad (1)$$

Off-policy RL methods based on approximate dynamic programming typically use a state-action value function (Q -function), denoted by $Q(s, a)$, which represents the expected discounted return from taking action a in state s and then following policy π .

Unlike online RL methods (on-policy or off-policy), offline RL learns solely from previously collected datasets without further environment interaction. As in many modern offline RL approaches, we rely on approximate dynamic programming and minimizes the temporal difference error using the following objective:

$$L_{TD}(\theta) = \mathcal{E}_{(s,a,s') \sim \mathcal{D}} [(r(s, a) + \gamma \max_{a'} Q_{\hat{\theta}}(s', a') - Q_{\theta}(s, a))^2], \quad (2)$$

where $\mathcal{D}$ is the dataset, $Q_{\theta}(s, a)$ is a parameterized Q -function, $Q_{\hat{\theta}}(s, a)$ is a target network (e.g., with soft parameters updates defined via Polyak averaging), and the policy is defined as $\pi(s) = \arg \max_a Q_{\theta}(s, a)$. Most recent offline RL methods modify either the value function loss (above) to regularize the value function in a way that keeps the resulting policy close to the data, or constrain the $\arg \max$

policy directly. This is important because out-of-distribution actions a' can produce erroneous values for $Q_{\hat{\theta}}(s', a')$ in the above objective, often leading to *overestimation* as the policy is defined to maximize the (estimated) Q -value.

B. Implicit Q-Learning

Implicit Q-Learning (IQL) is an offline RL method that avoids explicit maximization over unseen actions and therefore reduces overestimation bias in fixed datasets [3]. Instead of directly computing a policy target with $\max_{a'} Q(s', a')$, IQL decomposes learning into value, critic, and actor updates constrained by dataset support. First, the value network is trained with expectile regression:

$$L_V(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}} [L_2^{\tau}(Q_{\theta}(s, a) - V_{\psi}(s))], \quad (3)$$

where $L_2^{\tau}(u) = |\tau - \mathbf{1}(u < 0)|u^2$ is the asymmetric expectile loss. Next, the critic is updated using a bootstrapped target based on the learned value:

$$L_Q(\theta) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}} [(r + \gamma V_{\psi}(s') - Q_{\theta}(s, a))^2]. \quad (4)$$

Finally, the actor is optimized by advantage-weighted behavior cloning:

$$L_{\pi}(\phi) = -\mathbb{E}_{(s,a) \sim \mathcal{D}} \left[\exp(\beta(Q_{\theta}(s, a) - V_{\psi}(s))) \log \pi_{\phi}(a|s) \right], \quad (5)$$

which prioritizes high-advantage actions while keeping the policy close to the data distribution. This structure makes IQL particularly suitable for mixed-quality autonomous driving datasets where stability and safety are critical.

IV. DESIGN AND METHODOLOGY

A fundamental challenge in offline reinforcement learning is the overestimation bias phenomenon, where an agent overly optimistically evaluates hypothetical actions not present in the dataset. IQL, adopted in this study, addresses this issue by implicitly excluding actions outside the dataset from the evaluation process. This enables stable convergence and the extraction of only optimal trajectories, even within data mixed with noise or failure cases. In this section, we detail the design of the state and action spaces, loss functions for neural networks, and the reward function used in our lane change trajectory prediction model. Fig. 1 illustrates the functional diagram of the proposed IQL-based lane-change policy learning pipeline.

A. State and Action Space Design

The data model is designed to match the specifications of the value-based architecture of the AD4RL benchmark:

- **Input Data (State):** A one-dimensional numerical vector (1D vector) containing compressed surrounding perception information, including the ego-vehicle's velocity, relative distances, and relative speeds to vehicles in the front, rear, and adjacent lanes.

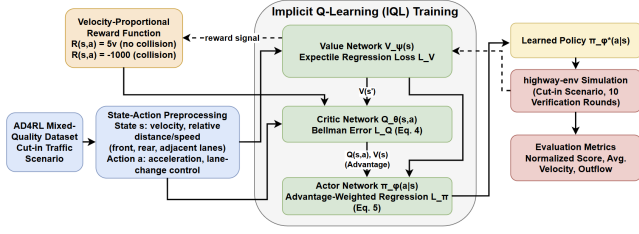


Fig. 1. Functional diagram of the proposed IQL-based lane-change policy learning pipeline.

- **Output Data (Action):** A continuous vector for simulator control, featuring a two-dimensional (2D) structure that includes longitudinal acceleration and lateral lane-change control values.

B. Loss Functions for Neural Networks

1) **Critic Loss (Value Evaluation Neural Network Error):** To enhance the accuracy of the predicted future cumulative reward (Q) for the current state s and action a , Mean Squared Bellman Error (MSBE) based on the Bellman Equation is used as shown in Eq. (4), where r is the current reward, γ is the discount factor, and $V(s')$ is the safe, top-tier expected value derived through Expectile Regression. This enables recursive learning (bootstrapping) within offline single-step data.

2) **Actor Loss (Policy Control Neural Network Error):** To optimize the agent’s actual steering policy, an Advantage Weighted Regression method as shown in Eq. (5) is used. This method avoids simple imitation learning and filters only high-quality driving data by assigning weight only when a specific driving action value (Q) is in an “advantage” state, meaning it outperforms the average value (V).

C. Reward Function Design

The redesigned reward function encourages active high-speed overtaking while maintaining safety:

$$R(s, a) = \begin{cases} v \times 5.0 & \text{if no collision,} \\ -1000 & \text{if collision occurs,} \end{cases} \quad (6)$$

where v is the ego-vehicle’s velocity. The velocity-proportional bonus encourages efficient high-speed driving, while the heavy penalty protects safety constraints.

V. EVALUATION AND RESULTS

In this section, we describe the experimental setup, including the hardware and software environment, dataset construction, custom reward function, hyperparameter settings, and evaluation metrics used to assess the performance of our lane change trajectory prediction model.

A. Evaluation Environment

- **OS/Hardware:** MPS (Metal Performance Shaders) based on Apple Silicon GPU in a macOS environment
- **Simulator/Framework:** Built an autonomous driving simulation environment based on Eclipse SUMO and

the Flow framework. Utilized PyTorch and d3rlpy (v2.x), a library dedicated to offline RL

The dataset and reward function used in the study are described below:

- **Target Dataset:** Constructed a mixed dataset of 100,000 total steps by consolidating four quality data levels (Final, Medium, Random, Human-like) from the Cut-in Traffic scenario within the AD4RL benchmark (split into 80% Train, 20% Validation)
- **Reward Function:** The velocity-proportional reward function was customized to encourage high-speed overtaking while maintaining collision avoidance

B. Hyperparameter Settings

The hyperparameter settings used in the study are presented in Table I:

TABLE I
HYPERPARAMETER CONFIGURATION

Parameter	Value
Batch Size	512
Actor/Critic Learning Rate	3e-4 / 3e-4
Weight Temp (beta)	4.0
Max Weight	100.0
Total Training Steps (Epochs)	100,000 (50 Epochs)

C. Evaluation Metrics

We adopt the standard normalization formula (Normalized Score) of the AD4RL benchmark to objectively evaluate the agent’s cumulative reward performance:

$$\text{Normalized Score} = \frac{\text{Return} - \text{Return}_{\text{random}}}{\text{Return}_{\text{final}} - \text{Return}_{\text{random}}} \times 100 \quad (7)$$

The performance of a random policy (Random) that has learned nothing is defined as 0 points, and the performance of the optimal expert policy (Final), which serves as the upper bound, is defined as 100 points.

Objective driving performance according to the normalized score (0–100) is categorized into five grades as shown in Table II.

TABLE II
PERFORMANCE EVALUATION SCALE

Score Range	Baseline Mapping	Objective Performance
0–20	Random Policy	Inoperable driving state
21–40	Human-like (Lower)	Conservative patterns
41–60	Medium Policy	Basic lane keeping
61–80	Medium-Expert	Safety-first overtaking
81–100	Final (Expert)	Optimal expert driving

The normalized return values and driving metrics during the model’s 10 verification rounds are presented in Table III. The performance summary of the proposed model is as follows:

- **Average normalized return:** 89.03 ± 9.78 points
- **Average velocity:** 27.70 ± 2.21 m/s
- **Average outflow:** 1845.20 ± 125.41 vehicles/hour

TABLE III
VERIFICATION RESULTS (10 ROUNDS)

Round	Normalized Score	Round	Normalized Score
0	92.71	5	91.10
1	95.50	6	94.98
2	89.94	7	80.60
3	62.50	8	95.25
4	94.41	9	93.35

D. Key Findings

1) *Optimal Expert Level Achievement*: The average performance of the proposed model landed within the highest tier, the Final (Expert) level. This is evaluated to be because the Advantage Weighted Regression (AWR) mechanism of IQL successfully filtered and restored only high-performance overtaking policies with high advantages. The average velocity also demonstrated efficient high-speed driving capability at 27.7 m/s.

2) *Stabilization Mechanisms*: Three critical mechanisms enabled training stability:

- 1) **Weight Upper Limit (Max Weight = 100.0)**: Weight clipping limits large gradients in the policy network and helps keep training stable.
- 2) **Variance Control (Weight Temp = 4.0)**: In the high-variance mixed dataset, tuning β reduces noise across data qualities and emphasizes high-quality overtaking behaviors.
- 3) **Gradient Smoothing (Batch Size = 512)**: A larger batch size balances samples from multiple datasets in each update and stabilizes the critic's Bellman error.

3) *Notable Exception*: A temporary performance drop was observed in Round 3 (62.50 points). Manual inspection of the rollout suggests a late-triggering, aggressive cut-in forced a hard braking response, so the collision-avoidance term transiently dominated the velocity term, reflecting the trade-off between the reward's safety and velocity components; a full trajectory-level analysis is left for future work.

4) *Convergence Analysis*: The training loss graphs, as shown in Fig. 2, revealed:

- **Critic Loss**: The error temporarily increased in early learning stages while adapting to the redesigned reward scale, but stabilized downward in alignment with the Bellman Equation
- **Actor Loss**: The loss did not converge to 0 in the latter half (after Epoch 40) but showed minor oscillations, judged to be due to variance from weight calculation caused by the multi-quality distribution unique to the mixed dataset, leading to convergence toward a realistic compromise policy

VI. CONCLUSION

This study developed an autonomous driving agent using Implicit Q-Learning and a velocity-proportional reward in the AD4RL Cut-in environment. The agent achieved an

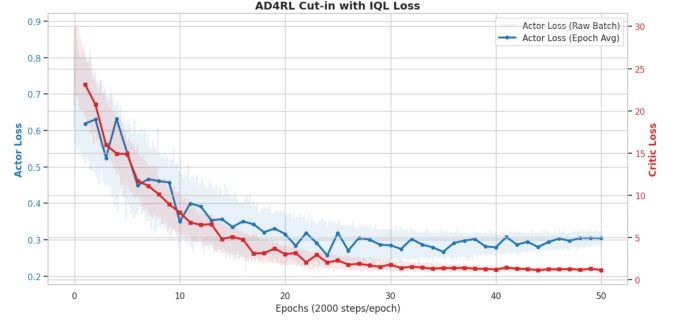


Fig. 2. Actor and Critic Loss Convergence

average normalized score of 89.03, indicating expert-level high-speed overtaking performance even with mixed-quality offline data. It also maintained safety constraints while reaching an average speed of 27.7 m/s, showing strong potential for real-world autonomous driving. Future work will benchmark against offline RL baselines (CQL, BCQ, TD3+BC) and the original AD4RL reward, report safety metrics (collision rate, minimum time-to-collision), expand evaluation to more scenarios with statistical testing, examine the sim-to-real gap, and add jerk/comfort constraints.

REFERENCES

- [1] H. H. Jeong, Y. C. Shen, J. P. Jeong, and T. T. Oh, "A comprehensive survey on vehicular networking for safe and efficient driving in smart transportation: A focus on systems, protocols, and applications," *Vehicular Communications*, vol. 31, p. 100349, 2021.
- [2] H. Chu, H. Wang, Y. Cheng, A. Li, W. Tian, B. Gao, and H. Chen, "Decision making for autonomous vehicles: A mixed curriculum reinforcement learning approach and a novel safety intervention method," *Transportation Research Part C: Emerging Technologies*, vol. 181, p. 105369, 2025.
- [3] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," 2021.
- [4] D. Lee, C. Eom, and M. Kwon, "Ad4rl: Autonomous driving benchmarks for offline reinforcement learning with value-based dataset," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [5] E. Leurent, "An environment for autonomous driving decision-making," <https://github.com/eleurent/highway-env>, 2018.
- [6] D. Coelho, M. Oliveira, and V. Santos, "Rlfold: Reinforcement learning from online demonstrations in urban autonomous driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 660–11 668.
- [7] L. Zhang, Y. Xiong, Z. Yang, S. Casas, R. Hu, and R. Urtasun, "Copilot4d: Learning unsupervised world models for autonomous driving via discrete diffusion," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [8] J. Philion, X. B. Peng, and S. Fidler, "Trajenglish: Traffic modeling as next-token prediction," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [9] R. Yang, H. Zhong, J. Xu, A. Zhang, C. Zhang, L. Han, and T. Zhang, "Towards robust offline reinforcement learning under diverse data corruption," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [10] Q. Di, H. Zhao, J. He, and Q. Gu, "Pessimistic nonlinear least-squares value iteration for offline reinforcement learning," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [11] F.-M. Luo, T. Xu, X. Cao, and Y. Yu, "Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.