Imperfect Dataset is Enough: Reinforcement Learning with Selective Imitation from Prior Data

Chanin Eom, Dongsu Lee, Minhae Kwon

***Abstract*—Deep reinforcement learning (RL) has served as a promising solution in tasks requiring sequential decision-making, akin to autonomous driving. However, online RL still suffers from the exploration of active data collection through environmental interactions for policy improvement. A randomly initialized policy could take a long time to accrue valuable experience, impeding learning efficiency. Fortunately, an *off-policy* method can leverage prior data (e.g., offline or demonstration data) for updating the policy. This work delves into how efficiently off-policy methods can function when using prior datasets. We use a *behavioral cloning* (BC) regularizer for the policy update. This can guide the agent to learn actions contained in prior datasets, but it could have a low resistance to suboptimal datasets. To address this drawback, we propose *reward-adaptive prior data RL* (RAPID-RL). The RAPID-RL distinguishes itself by selectively imitating prior data based on the reward information, thereby enhancing the efficiency and applicability of the learning process. To demonstrate the superiority of RAPID-RL, we consider autonomous driving tasks with various scenarios and datasets. Simulation results confirm that the proposed solution rapidly converges with high performance across scenarios.**

***Index Terms*—Deep Reinforcement Learning, Off-policy Reinforcement Learning, Prior Data, Behavioral Cloning, Imitation Learning**

I. Introduction

Deep RL has made remarkable achievements across diverse domains requiring sequential decision-making and human preferences, e.g., autonomous driving, robotics, and large language models [1], [2]. Nevertheless, sample efficiency remains a formidable hurdle in online RL settings [3], [4]. This challenge arises from the need for many interactions involving exploration and trial-and-error with the environment to iteratively refine policies. While this approach is suitable for tasks amenable to simulation, it faces substantial obstacles when applied to real-world problems due to the high cost associated with collecting data [5], [6].

Fortunately, the off-policy method offers a promising avenue for leveraging offline data stored within a replay buffer to improve policies. RL with prior data (RLPD) exploits this advantage, thereby optimizing policy using the prior (i.e., offline) dataset [3], [5]. This approach can enable a rapid escape from a suboptimal policy because the agent can utilize a high-quality trajectory, which it cannot generate through its own learning policy. However, relying solely on prior data is insufficient for building a policy that consistently delivers high performance. This is because the fundamental optimization challenges of RL cannot be overcome merely by leveraging additional prior datasets.

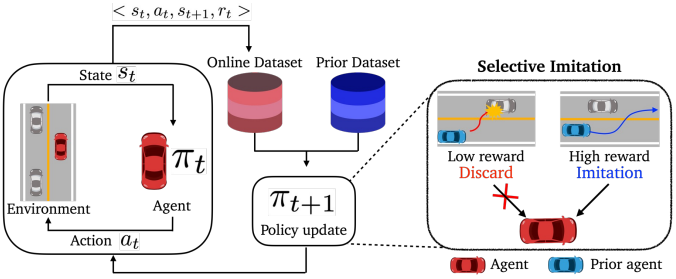


Fig. 1: Overview of the proposed solution. The RAPID method incorporates the prior dataset into the online RL framework to enhance sample efficiency. Additionally, selective imitation allows the learning policy to be guided only by high-quality sample data from both online and prior datasets.

Incorporating BC regularizers can mitigate this issue by effectively guiding the learning policy to stay close to the prior dataset through imitation. Due to its simplicity and effectiveness, these regularizers are widely adopted in studies where policies need to be derived from an offline dataset (e.g., offline RL) [7], [8]. However, applying a BC regularizer within an online RL framework requires careful consideration, because the quality of the dataset can set an upper bound for the policy due to the imitation property. Although utilizing online interaction data can mitigate this issue by broadening the dataset range, it also introduces another challenge. This stems from the imitation of suboptimal data generated by a randomly initialized policy. Consequently, simply using online data can exacerbate limitations imposed by the initial policy.

To address it, this work studies RLPD methods to accelerate learning using prior data more efficiently. We dub the proposed solution as RAPID-RL. The RAPID-RL adopts a BC regularizer to imitate the action of the prior and to use online data efficiently. We add reward information as an important weight into the BC regularizer to discern the good and bad data. This method is simple, has a low computational cost, and does not require the additional hyperparameter for training, compared to vanilla online RL. Namely, the proposed solution can take

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00278812).

All authors are with the Department of Intelligent Semiconductors and Minhae Kwon is with School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea (e-mail: eci0623@soongsil.ac.kr, movement-water@soongsil.ac.kr, minhae@ssu.ac.kr) (*Corresponding author: Minhae Kwon*).

IEEE ICRA (International Conference on Robotics and Automation) Workshop on Human-aligned Reinforcement Learning for Autonomous Agents and Robots, Yokohama, Kanagawa, Japan. 2024.

robustness regarding dataset quality at a low cost. We run the simulation to demonstrate the efficiency of these minimal changes. The overall concept illustrated in this study can be seen in Figure 1.

II. Background

A. Off-policy Reinforcement Learning with Prior Data

RL tasks can be modeled with a Markov decision process. It can be defined as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma >$, which includes a state $s_t \in \mathcal{S}$, an action $a_t \in \mathcal{A}$, a state transition probability $\mathcal{T}$, a reward function $\mathcal{R}$ and a temporal discounted factor γ. The agent aims to maximize the cumulative reward $\mathbb{E}\big[\sum_{t=0}^{H-1} \gamma^t r_t\big]$, where $r_t = \mathcal{R}\big(s_t, a_t, s_{t+1}\big)$ is a reward, and H represents the finite time horizon. This objective can be modified as the maximize state-value function $\mathbb{E}\big[Q(s_t, a_t)\big]$ by Bellman equation.

In off-policy RL, the agent can leverage transition data (s_t, a_t, r_t, s_{t+1}) from various policies. Conventionally, many studies store online transition data in an online buffer (i.e., a replay buffer) denoted as $\mathcal{D}_{on}$ and reuse it by sampling to train policy [9]. To exploit these advantages more efficiently, recent works leverage both prior buffer $\mathcal{D}_{prior}$ and the online buffer $\mathcal{D}_{on}$ [3], [10]. In this framework, the objective of policy π can be expressed as follows.

$$\pi = \arg\max_{\pi} \mathbb{E}_{s_t \sim \mathcal{D}_{on} \cup \mathcal{D}_{prior}} \Big[Q\Big(s_t, \pi(s_t)\Big) \Big]$$

In this study, we follow the off-policy RL framework with prior data. Unlike previous works, we focus on a method that can more efficiently utilize the prior dataset with a BC regularizer.

B. Behavioral Cloning

BC is a conventional learning method that imitates the behaviors included in the prior buffer $\mathcal{D}_{prior}$. The objective of BC is to minimize the error between the agent's action $\pi(s_t)$, and sampled prior action a_t as follows.

$$\pi = \arg\min_{\pi} \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}_{prior}} \Big(\pi(s_t) - a_t \Big)^2$$

Unlike RL optimization, BC employs a supervised learning method and thus heavily depends on the quality of the dataset. From this perspective, most studies utilize expert demonstration as the prior datasets, which consistently provide high-quality samples [10]. Although some studies address this issue by incorporating the BC term into the RL objective as a regularizer, they still face challenges in balancing the RL and BC objectives [7], [8].

To address the ongoing challenges, we proposed the selective BC regularizer, which dynamically adjusts the degree of the BC term based on the quality of the sampled data.

III. Proposed Solution: Reward Adaptive Prior Data Reinforcement Learning

A. Selective Behavioral Cloning Regularizer

This work aims to selectively leverage high-quality prior behaviors, irrespective of dataset quality. To do this, we assign reward information of prior data transition as a weighting factor into the BC term. This weight automatically determines the importance between BC and Q terms depending on the quality of a given data.

The objective of the proposed solution can be defined as follows.

$$\pi = \arg\max_{\pi} \mathbb{E}_{(s_t, a_t, r_t) \sim \mathcal{D}_{on} \cup \mathcal{D}_{prior}} \Big[Q\Big(s_t, \pi(s_t)\Big) - \sigma\Big(r_t - \epsilon\Big) \Big(\pi(s_t) - a_t\Big)^2 \Big] \tag{1}$$

In (1), $\sigma(\cdot)$ denotes the rectified linear unit function, i.e., $\sigma(r_t) = \max(0, r_t)$. This function activates the BC term if a given data is valuable ($r_t > 0$), and conversely, deactivates the BC term when $r_t \leq 0$. Suppose there is a specific threshold ϵ for distinguishing the value of the reward desired by the user. In that case, the importance weight can be set as the difference between the reward and the threshold $(r_t - \epsilon)$. This revised objective enables the agent to be guided by only high-quality prior data.

Algorithm 1 Reward Adaptive PrIor Data RL (RAPID-RL)

Require: the number of episodes E, episode length T, batch size B, soft-update rate τ

Initialization: prior buffer $\mathcal{D}_{prior}$, online buffer $\mathcal{D}_{on} = \emptyset$, critic network Q_θ, actor-network π_ϕ, target networks: $Q_{\theta'} \leftarrow Q_\theta$, $\pi_{\phi'} \leftarrow \pi_\phi$

```
for episode e = 1 to E do
    Reset state s_1
    for t = 1 to T do
        Take action a_t ~ π_φ(s_t)
        Get next state s_{t+1} and reward r_t ← R(s_t, a_t, s_{t+1})
        Store transition (s_t, a_t, r_t, s_{t+1}) in online buffer D_on
        Sample B/2 transition from D_on
        Sample B/2 transition from D_prior
        Update actor φ based on (2)
        Update critic θ based on (3)
        Update target networks:
        θ' ← τθ + (1 − τ)θ'    φ' ← τφ + (1 − τ)φ'
    end for
end for
```

B. Online Reinforcement Learning with Selective Behavioral Cloning Regularizer

We use the three techniques to leverage the prior data in an online RL scheme efficiently.

1) Partition of Prior and Online Buffers: We use the prior buffer $\mathcal{D}_{prior}$ along with the online buffer $\mathcal{D}_{on}$ to enhance sample efficiency. Both of these buffers are incorporated in the training phase by symmetric sampling. When the batch size is B, we sample the same amount of data $B/2$ from each buffer and use them for training [11], [12].

2) Layer Normalization: Given that the proposed solution enables the collection of an additional dataset, we opt not to use the explicit regularizer in the Q network optimization. Motivated by the previous works, we consider the introduction of layer normalization, which can mitigate the effect of incorrect behavioral estimation of the Q value [3].

3) Actor-critic Algorithm: We adopt an actor-critic approach, off-policy RL methods, to approximate the policy and Q function over continuous state and action spaces. The actor network π_ϕ can be regarded as a policy because it determines the action from the given state. The loss function of the actor network π_ϕ can be expressed as follows.

$$\mathcal{L}(\phi) = \mathbb{E}_{(s_t,a_t,r_t)\sim\mathcal{D}_{on}\cup\mathcal{D}_{prior}}\Bigg[-Q_\theta\Big(s_t,\pi_\phi(s_t)\Big) + \sigma\Big(r_t-\epsilon\Big)\Big(\pi(s_t)-a_t\Big)^2\Bigg] \tag{2}$$

The critic network Q_θ aims to accurately evaluate the value of the state-action pair (s, a). The loss function of the critic network can be defined as follows.

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t,a_t,s_{t+1},r_t)\sim\mathcal{D}_{on}\cup\mathcal{D}_{prior}}\Bigg[Q_\theta\Big(s_t,a_t\Big) - \Big(r_t+\gamma Q_\theta\Big(s_{t+1},\pi_\phi(s_{t+1})\Big)\Big)\Bigg]^2 \tag{3}$$

Detailed pseudocode for training policy is provided in Algorithm 1.

IV. Simulation Results

In this section, we experimentally demonstrate our proposed solution within an autonomous driving task. First, we outline the simulation setup, including dataset details and baselines. Next, we provide experimental validations of the proposed solution across different scenarios and datasets.

A. Simulation Setups

To demonstrate the efficiency of the proposed solution, we select an autonomous driving task based on the FLOW simulator [13]. The simulation includes a single autonomous driving vehicle and non-autonomous driving vehicles.

1) Autonomous Driving Tasks: In this study, we employ the autonomous driving task to validate our proposed solution. In this task, the autonomous vehicle aims to drive with the desired velocity while preventing unsafe behaviors akin to car accidents. We take into account the following three driving scenarios.

- Lane-drop: This scenario features a lane-drop section that reduces road capacity. Traffic bottlenecks are frequently observed since the road carries a larger number of vehicles than it can accommodate. Therefore, the agent should learn to navigate through congested traffic by negotiating with other vehicles.
- Cut-in: This scenario is characterized by obstacle vehicles that drive at a constant speed below the desired velocity of the agent. The agent should overtake these obstacle vehicles to achieve its desired velocity.
- Merge: In this scenario, the agent always enters the on-ramp lane, which merges with the main lane. This condition requires the agent to develop a merging strategy that does not violate the safety distance of vehicles in the main lane.

2) Prior Datasets: We use three prior datasets collected from behavioral policies with different performances.

- `Initial`: This refers to the suboptimal prior dataset collected by an online RL agent with a pre-trained policy until $1/10$ of the total online RL training.
- `Medium`: It is generated using an online RL agent that has been trained up to half of the total training timesteps.
- `Final+Medium`: It is a blended dataset of `Medium` and `Final` datasets with the same ratio. The `Final` dataset is collected by a fully trained agent.

3) Baselines: We benchmark the following baseline algorithms for comparison.

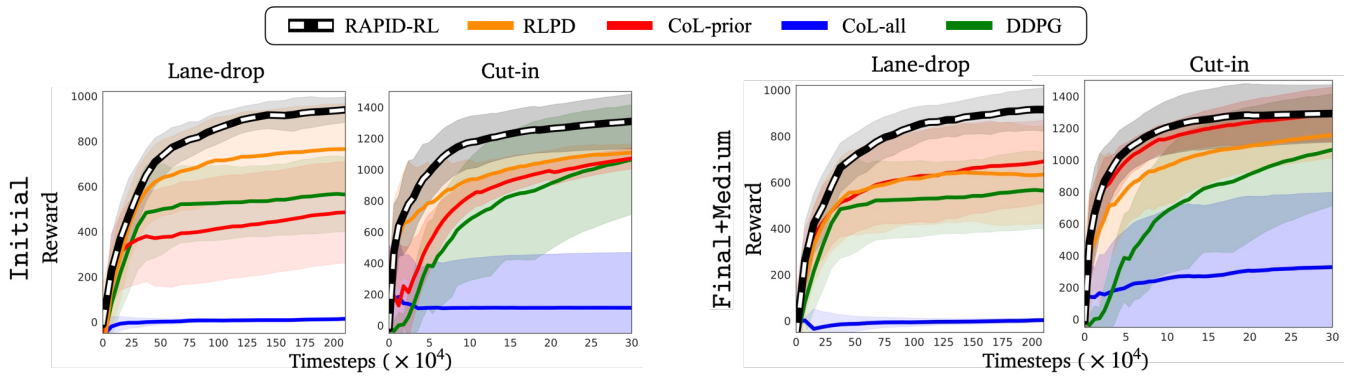
- **Reward-adaptive Prior Data Reinforcement Learning (RAPID-RL)**: It is the proposed solution that uses selective imitation. We consider deep deterministic policy gradient (DDPG) [9] as the backbone algorithm.
- **Reinforcement Learning with Prior Data (RLPD)** [3]: This method is an actor-critic algorithm that leverages a prior dataset. It can serve as a baseline, depending on whether a BC loss is applied.
- **Cycle-of-Learning (CoL)** [14]: This algorithm employs the BC regularized method. It can serve as a baseline for assessing the effects of introducing selective imitation. Given that a policy based on CoL can be limited by imitating suboptimal online data from a randomly initialized policy, we divide it into the following two approaches.
  - **CoL-prior**: This approach uses only the prior dataset to calculate the BC loss as; $\mathbb{E}_{(s_t,a_t)\sim D_{prior}}\Big(\pi(s_t)-a_t\Big)^2$.
  - **CoL-all**: This method calculates BC loss with the concatenated batches from both the online and prior buffers, i.e., $\mathbb{E}_{(s_t,a_t)\sim D_{prior}\cup D_{on}}\Big(\pi(s_t)-a_t\Big)^2$.
- **Deep Deterministic Policy Gradient (DDPG)** [9]: It is a popular online RL algorithm, serving as the baseline for the pure off-policy method.

B. Performance Comparison

This subsection assesses how each algorithm performs across different scenarios and datasets. We provide the reward comparison and convergence speed analysis across the baselines.

1) Reward Comparison: Figure 2 presents the reward curve across the driving scenarios. Each solid line and the shaded area represent the average reward and one standard deviation over the 10 random seeds.

Firstly, Figure 2(a) shows the reward performance using the `Initial` dataset, which is a low-quality prior dataset.



(a) Reward curves with `Initial` prior dataset.

(b) Reward curves with `Final+Medium` prior dataset.

Fig. 2: Reward curves in lane-drop and cut-in scenarios with `Initial` and `Final+Medium` prior datasets.

TABLE I: The average final performance and one standard deviation of each algorithm across the driving scenarios and dataset. The cyan-shaded area represents the highest reward achieved in each experiment.

scenario	Dataset	RAPID-RL (Proposed)	RLPD	CoL-prior	CoL-all	DDPG
Lane-drop	`Initial`	939.7 ± 52.4	765.4 ± 190.2	485.8 ± 213.2	14.3 ± 3.2	
	`Medium`	946.5 ± 110.2	739.0 ± 174.0	611.2 ± 112.0	11.2 ± 3.3	566.3 ± 165.8
	`Final+Medium`	915.6 ± 88.2	634.6 ± 200.7	690.6 ± 167.9	3.2 ± 7.1	
Cut-in	`Initial`	1309.2 ± 167.0	1109.2 ± 52.2	1071.5 ± 64.1	113.7 ± 336.7	
	`Medium`	1265.8 ± 183.1	1157.6 ± 155.1	1127.7 ± 66.7	104.0 ± 322.5	1064.0 ± 314.6
	`Final+Medium`	1292.1 ± 172.0	1155.8 ± 137.9	1288.3 ± 160.9	327.5 ± 445.9	
Merge	`Initial`	1993.1 ± 104.6	1716.0 ± 237.6	1618.4 ± 171.8	962.0 ± 529.7	
	`Medium`	2064.1 ± 79.1	1709.5 ± 99.7	2131.15 ± 158.1	1170.3 ± 900.0	1618.4 ± 171.8
	`Final+Medium`	2333.5 ± 91.4	1588.5 ± 289.4	2186.9 ± 290.8	1128.8 ± 851.4	

Notably, the proposed RAPID-RL method consistently shows the fastest policy improvement and the highest rewards across all driving scenarios. Although the RLPD method also exhibits improved performance compared to the DDPG approach, it consistently underperforms relative to the RAPID-RL method. Furthermore, most CoL-based methods fail to achieve the performance level of DDPG because imitating suboptimal data can limit the potential for policy improvement. These results confirm that the proposed solution can provide high-performance policies even with suboptimal datasets.

The reward performances under a high-quality dataset (i.e., `Final+Medium`) are provided in Figure 2(b). Overall, each algorithm achieves better performance compared to those using the `Initial` datasets, due to the enhanced quality of the prior data. Interestingly, the CoL-prior approach achieved significant performance improvements and is slightly less than the proposed solution. This result confirms that the performance of the CoL-based method significantly depends on the quality of the prior dataset. However, the CoL-all approach still fails to improve the policy because it imitates suboptimal online data from a randomly initialized policy, thus restricting policy improvement.

Table I presents the final reward values across the datasets and scenarios. In this table, the cyan-shaded box highlights the highest mean reward value among the baselines. The numerical results demonstrate that the proposed solution achieves the highest reward in most cases. Additionally, the CoL-prior exhibits a consistent decline in performance as the quality of the dataset deteriorates. These findings confirm that the proposed solution maintains robustness against variations in dataset quality, ensuring high-performance policies even with suboptimal datasets.

2) Behavioral Cloning Loss Analysis: This subsection offers a detailed analysis to explain why the proposed solution remains superior, even when implemented with a suboptimal prior dataset. Specifically, we analyze the trends in the BC loss across merge scenarios using the `Initial` and `Final+Medium` datasets. We take into account the following two types of BC losses.

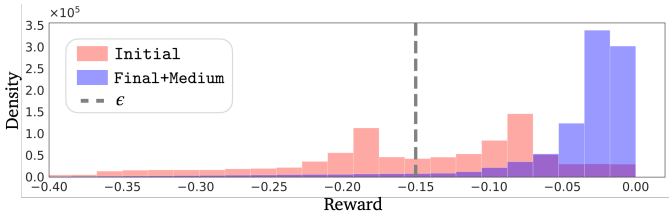
- Prior loss ($\mathcal{L}_{BC,prior}$): It denotes BC loss between the actions taken by agent $\pi(s_t)$ and the prior action data sampled from the prior buffer $a_t \sim \mathcal{D}_{prior}$.

$$\mathcal{L}_{BC,prior} = \mathbb{E}_{(s_t,a_t)\sim D_{prior}}\Big(\pi(s_t) - a_t\Big)^2$$

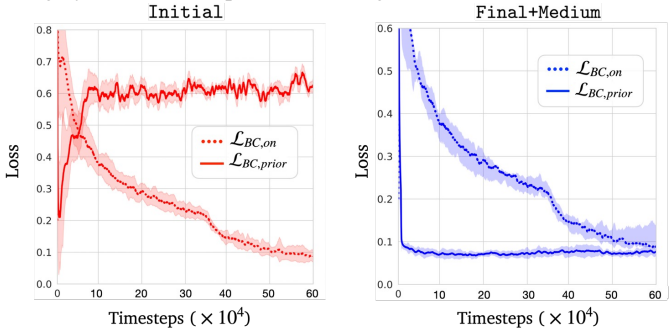
- Online loss ($\mathcal{L}_{BC,on}$): It quantifies BC loss between the actions taken by agent $\pi(s_t)$ and the replay action data sampled from the online buffer $a_t \sim \mathcal{D}_{on}$.

$$\mathcal{L}_{BC,on} = \mathbb{E}_{(s_t,a_t)\sim D_{on}}\Big(\pi(s_t) - a_t\Big)^2$$

Figure 3(a) illustrates the reward distribution of each dataset. In this figure, the gray dotted line represents the target reward $\epsilon = -0.15$. Most of the reward data for the `Final+Medium` dataset is distributed above ϵ, whereas the `Initial` dataset



(a) Reward distribution of `Initial` and `Final+Medium` dataset. The gray dotted line represents the target reward value.



(b) BC loss over timesteps. The solid line represents the prior loss, and the dotted line indicates the online loss across the 10 random seeds. The shaded area represents one standard deviation across the random seeds.

Fig. 3: The reward distribution and BC loss of the `Initial` and `Final+Medium` prior datasets.

exhibits a widely distributed reward centered around the value of ϵ. This property indicates that the agent should selectively imitate the `Initial` datasets, and imitate most of the `Final+Medium` datasets.

The results shown in Figure 3(b) effectively illustrate the operation of the proposed solution across datasets of varying quality. This figure details the BC loss results: the red line represents the results using the `Initial` dataset, and the blue line represents the BC loss using the `Final+Medium` dataset. In both cases, a solid line indicates the prior loss $\mathcal{L}_{BC,prior}$, and a dotted line indicates the online loss $\mathcal{L}_{BC,on}$ averaged over 10 random seeds. The shaded area around each line indicates one standard deviation.

In the `Initial` dataset, the prior loss initially starts lower than the online loss due to the suboptimality of the randomly initialized policy. However, as the policy improves, prior loss increases beyond the online loss. In contrast, for the `Final+Medium` dataset, prior loss remains consistently lower than online loss throughout training. These findings confirm that the proposed solution effectively adapts its imitation strategy based on the quality of the dataset. We believe that this selective imitation property leads to consistently high and rapid performance across various dataset qualities.

V. Conclusion

In this study, we propose RAPID-RL, a method that selectively imitates only high-quality prior data. To implement it, we modify the conventional BC regularizer to the selective BC regularizer. The simulation results confirm that our proposed method exhibits a significantly rapid convergence speed compared to vanilla online approaches and demonstrates robustness regardless of the quality of prior datasets.

References

[1] L. Garaffa, M. Basso, A. Konzen, and E. Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[2] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

[3] P. Ball, L. Smith, I. Kostrikov, and S. Levine, "Efficient online reinforcement learning with offline data," in *ICML*, 2023.

[4] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[5] H. Walke, J. Yang, A. Yu, A. Kumar, J. Orbik, A. Singh, and S. Levine, "Don't start from scratch: Leveraging prior data to automate robotic reinforcement learning," in *CORL*, 2023.

[6] J. Lyu, X. Ma, X. Li, and Z. Lu, "Mildly conservative Q-learning for offline reinforcement learning," in *NeurIPS*, 2022.

[7] D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov, "Revisiting the minimalist approach to offline reinforcement learning," in *NeurIPS*, 2023.

[8] S. Fujimoto and S. Gu, "A minimalist approach to offline reinforcement learning," in *NeurIPS*, 2021.

[9] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR*, 2016.

[10] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[11] M. Nakamoto, Y. Zhai, A. Singh, M. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, "Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning," in *NeurIPS*, 2023.

[12] S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin, "Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble," in *CORL*, 2022.

[13] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. Bayen, "FLOW: A modular learning framework for mixed autonomy traffic," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1270–1286, 2021.

[14] V. Goecks, G. Gremillion, V. Lawhern, J. Valasek, and N. Waytowich, "Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments," in *AAMAS*, 2020.