RLfOLD: Reinforcement Learning from Online Demonstrations in Urban Autonomous Driving

Daniel Coelho [1, 2], **Miguel Oliveira** [1, 2], **Vítor Santos** [1, 2]

[1] Department of Mechanical Engineering, University of Aveiro, 3810-193 Aveiro, Portugal
[2] Intelligent System Associate Laboratory (LASI), Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal
{danielsilveiracoelho, mriem, vitor}@ua.pt

Abstract

Reinforcement Learning from Demonstrations (RLfD) has emerged as an effective method by fusing expert demonstrations into Reinforcement Learning (RL) training, harnessing the strengths of both Imitation Learning (IL) and RL. However, existing algorithms rely on offline demonstrations, which can introduce a distribution gap between the demonstrations and the actual training environment, limiting their performance. In this paper, we propose a novel approach, Reinforcement Learning from Online Demonstrations (RLfOLD), that leverages online demonstrations to address this limitation, ensuring the agent learns from relevant and up-to-date scenarios, thus effectively bridging the distribution gap. Unlike conventional policy networks used in typical actor-critic algorithms, RLfOLD introduces a policy network that outputs two standard deviations: one for exploration and the other for IL training. This novel design allows the agent to adapt to varying levels of uncertainty inherent in both RL and IL. Furthermore, we introduce an exploration process guided by an online expert, incorporating an uncertainty-based technique. Our experiments on the CARLA NoCrash benchmark demonstrate the effectiveness and efficiency of RLfOLD. Notably, even with a significantly smaller encoder and a single-camera setup, RLfOLD surpasses state-of-the-art methods in this evaluation. These results, achieved with limited resources, highlight RLfOLD as a highly promising solution for real-world applications.

1 Introduction

Urban Autonomous Driving (AD) is considered a challenging and critical task. In order to navigate effectively, agents must analyze a highly intricate environment and continuously make real-time decisions to adhere to driving regulations, while also interacting with other dynamic agents like drivers and pedestrians (Coelho and Oliveira 2022). Consequently, researchers have been redirecting their efforts from rule-based methods to end-to-end learning approaches.

End-to-end learning methods can be divided into two categories: Imitation Learning (IL) and Reinforcement Learning (RL). In IL, an agent learns a task by imitating an expert's behavior, leveraging expert demonstrations as ground truth. The main advantage of IL is the ability to rapidly learn from expert knowledge, accelerating the learning process and acquiring safe and efficient behaviors (Liu et al. 2022). However, agents trained with IL may face challenges in generalizing to unseen scenarios, as they tend to be biased towards the demonstrated behavior (Chekroun et al. 2021). On the other hand, RL involves learning through trial and error, where an agent explores the environment, receives feedback in the form of rewards, and gradually improves its policy. A key advantage of RL is its ability to handle unknown situations. However, RL often requires extensive exploration and can be sample-inefficient, requiring significant time and resources for learning (Van Hasselt, Guez, and Silver 2016).

Reinforcement Learning from Demonstrations (RLfD) seeks to harness the benefits of both IL and RL by integrating expert demonstrations into the RL training. Thus offering a significant boost in sample efficiency compared to standalone RL (Liu et al. 2022). This enhancement stems from the ability of expert demonstrations to minimize the required interactions with the environment for learning the desired behavior. Moreover, the expert demonstrations provide valuable insights that enable the agent to explore the state-action space more effectively (Goecks et al. 2019).

Despite the recent advancements of RLfD (Nair et al. 2018; Hansen et al. 2022), the conventional approach of collecting a demonstration dataset has inherent limitations. One major drawback is the requirement of pre-collecting a dataset, which can be a laborious and time-consuming process. In complex domains, like urban AD, it can be particularly arduous to ensure the dataset's diversity and coverage of various scenarios and edge cases. Moreover, the reliance on an offline dataset introduces a potential distribution gap between the demonstrations and the training environment, hindering the agent's ability to generalize effectively (Chekroun et al. 2021). One known strategy to mitigate the distribution mismatch between offline datasets and the training environment is the DAGGER algorithm (Ross, Gordon, and Bagnell 2011), which iteratively refines policies by aggregating training data across a mixture of expert and learner-induced distributions. However, while DAGGER reduces the distribution gap, it does not inherently account for the uncertainty in decision-making, which can be critical in dynamic and unpredictable urban driving scenarios.

To tackle the limitations of traditional RLfD, we introduce RLfOLD. RLfOLD utilizes online demonstrations, collected using privileged information from the simulator, which cir-

cumvents the need for a pre-collected dataset. These demonstrations are seamlessly integrated into the agent's replay buffer, ensuring that the agent learns from up-to-date and pertinent scenarios, thereby effectively bridging the distribution gap. Furthermore, RLfOLD innovates by merging IL with RL through a dual standard deviation policy network. By employing different standard deviations, the algorithm can adapt to the varying levels of uncertainty inherent in RL and IL. Inspired by recent works (Menda, Driggs-Campbell, and Kochenderfer 2019; Peng et al. 2022; Kelly et al. 2019; Li, Peng, and Zhou 2022; Dey et al. 2021), our approach further refines the exploration process. It empowers the agent with the ability to selectively invoke expert guidance when faced with high uncertainty, enhancing the decision-making process and potentially leading to more effective learning.

Overall, we summarize our main contributions as follows:

- Introduce RLfOLD, a novel approach that seamlessly integrates IL and RL by leveraging online demonstrations, effectively bridging the distribution gap between demonstration and training environments;
- Propose a policy network that outputs two standard deviations, enabling adaptive control for exploration and IL training while considering uncertainty in both domains;
- Incorporate an uncertainty-based technique guided by an online expert to enhance the exploration process;
- Conduct extensive experiments on the NoCrash benchmark, which demonstrate the superior effectiveness and efficiency of RLfOLD, surpassing state-of-the-art methods even with reduced resources.

The source code of RLfOLD is available at `https://github.com/DanielCoelho112/rlfold`.

2 Related Work

While RLfOLD is applicable to various tasks, our focus is on testing RLfOLD in the context of urban AD using the CARLA simulator (Dosovitskiy et al. 2017). As such, this section is focused on the application of IL, RL, and RLfD methods within the CARLA environment.

2.1 Imitation Learning

IL methods aim to learn from an expert using offline demonstrations. In the domain of AD, various IL approaches have been proposed, and they have demonstrated significant success. Notably, IL-based methods have consistently achieved top performance in the CARLA Leaderboard, showcasing their effectiveness in tackling complex driving tasks. Early works include CIL (Codevilla et al. 2017) and CILRS (Codevilla et al. 2019) that employ a conditional architecture to activate different policies based on the navigation command received. LBC (Chen et al. 2020) and Roach (Zhang et al. 2021) use privilege experts to provide knowledge to student models. Transfuser (Chitta et al. 2022; Prakash, Chitta, and Geiger 2021) designs a multimodal transformer that fuses the front camera image and LiDAR data, and then a simple GRU to auto-regress navigation waypoints. LAV (Chen and Krähenbühl 2022) trains on data from experiences collected not just from the ego-vehicle, but also from all surrounding vehicles. This is accomplished by learning a viewpoint-invariant spatial intermediate representation. TCP (Wu et al. 2022) proposes two branches that generate the planned trajectory and the multi-step control commands, respectively. Then the outputs of both branches are fused to achieve complementary advantages. Finally, InterFuser (Shao et al. 2023) uses a transformer to fuse multi-view sensors to encourage global contextual perception. Despite the remarkable achievements of IL approaches, a significant challenge in their deployment lies in addressing the distribution gap between the demonstration dataset and the environment in which the agent interacts.

2.2 Reinforcement Learning

RL has been used in AD to overcome the shortcomings of IL, however, vision-based RL presents several challenges. One such challenge is the training of a convolution encoder alongside a policy network, which often leads to catastrophic self-overfitting (Coelho, Oliveira, and Santos 2023). To address this issue, RLAD (Coelho, Oliveira, and Santos 2023) proposes an image encoder that leverages both Adaptive Local Signal Mixing (A-LIX) (Cetin et al. 2022) layers and image augmentations. While RLAD represents a significant advancement in vision-based RL for urban AD, its performance still falls short of the current state-of-the-art methods. The most successful RL algorithms applied in urban AD disentangle the perception network from the policy network by performing two-stage training (Coelho, Oliveira, and Santos 2023). The first stage consists of encoding the sensor data in a latent representation by pretraining a large encoder on visual tasks, such as classification and segmentation (Chekroun et al. 2021). Then, the latent representation is processed by an RL algorithm to train the policy network. Following this line, IAs (Toromanoff, Wirbel, and Moutarde 2019) proposes an algorithm composed of two subsystems. First, the encoder is trained using auxiliary tasks. Then the encoder is frozen and an RL algorithm is trained on the encoder latent space. Another example of this disentanglement is CADRE (Zhao et al. 2022). This method first trains offline a co-attention perception module to learn the relationships between the input and the corresponding control commands from a dataset. Then, this module is frozen and is used to feed an efficient distributed Proximal Policy Optimization (PPO) that learns the driving policy. While RL overcomes the distribution gap limitation of IL, it often suffers from sample-inefficiency, requiring significant time and resources for learning.

2.3 Reinforcement Learning From Demonstrations

As stated in Section 1, both IL and RL have inherent limitations, which have led to the growing interest in the concept of RLfD over the years (Hester et al. 2017; Vecerik et al. 2017; Liu et al. 2022). The main objective of RLfD is to combine the sample-efficiency of IL with the exploration capability of RL. For instance, CIRL (Liang et al. 2018) adopts a two-stage training approach. Initially, the agent is trained using IL with human demonstrations, followed by fine-tuning using an RL algorithm. BC-SAC (Lu

et al. 2022) and SAC-IL (Liu et al. 2022) propose methodologies that integrate the Soft Actor-Critic (SAC) algorithm with the IL loss. GRIAD (Chekroun et al. 2021) combines IL and RL under the assumption that all expert demonstrations are optimal and therefore assigned with maximum rewards. With this assumption, they process the expert demonstrations indistinguishable from the experiences of the RL exploration agent. While this assumption is overly optimistic, GRIAD is able to achieve very satisfactory results in both the CARLA Leaderboard and the NoCrash Benchmark (Codevilla et al. 2019). WOR (Chen and Krähenbühl 2022) assumes the world to be on rails, meaning that the actions of the agent affect only its own state and do not influence the environment. With this assumption, they convert the problem into a tabular model-based RL setup and supervise the policy learning with offline demonstrations. While these approaches have shown promising results, they share a common limitation: the use of offline demonstrations, which can introduce a distribution gap between the demonstrations and the training environment. To address this limitation, we propose a novel approach called Reinforcement Learning from Online Demonstrations (RLfOLD). Our method leverages online demonstrations, obtained during the agent's exploration, to bridge the distribution gap and to guide the exploration of the agent.

3 Method

3.1 Learning Framework

The learning process follows a Partially Observable Markov Decision Process (POMDP). The environment was built using the CARLA driving simulator (version 0.9.10.1). At every timestep t, the environment generates an observation o_t, which is passed to the agent. An observation is defined as a stack of three sets of tensors from the last $K = 2$ timesteps. Specifically, $o_t = \{(\boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V})_k\}_{k=0}^{1}$, where $\boldsymbol{I}$ represents a 3×256×256 image, $\boldsymbol{W}$ corresponds to the 2D coordinates with respect to the vehicle for the next $N = 10$ waypoints provided by the global planner from CARLA, and $\boldsymbol{V}$ is a two-dimensional vector containing the current speed and steering of the vehicle. The agent processes o_t and executes an action a_t according to its policy. Finally, the environment returns a reward r_t and the next observation o_{t+1}. The action a_t is composed of three continuous values: throttle and brake, which range from 0 to 1, and steering, which ranges from -1 to 1. Similar to (Coelho, Oliveira, and Santos 2023), we parameterize the throttle and brake commands using a target speed. Specifically, we append a PID controlled at the end of the policy network to generate the throttle and brake commands that correspond to the predicted target speed.

Figure 1 illustrates the architecture of RLfOLD. At a high level, the system can be divided into three main parts: encoder, actor-critic algorithm with IL, and online expert. Additionally, an important part of this work consists of using the online expert to guide the exploration. In general, RLfD algorithms use two replay buffers: one for the exploration agent and one for the demonstration agent (Chekroun et al. 2021; Liu et al. 2022; Lu et al. 2022). However, in our approach, we take advantage of having an online expert and create a single replay buffer, denoted as $\mathcal{D}$, to integrate information from both the exploration agent and the online expert. This replay buffer contains transitions in the form of $\big\{(o_t, a_t, a^*{}_t, r_t, o_{t+1})\big\}$, where a_t corresponds to the action executed by the agent, and a_t^* corresponds to action generated by the expert policy (π^*).

3.2 Encoder

As shown in Figure 1, RLfOLD trains simultaneously the encoder and the policy network. The reason is to ensure that the latent representations produced by the encoder are fully aligned with the driving task. However, as several studies have reported, performing Temporal Differences (TD) learning with a convolution encoder may lead to unstable training, premature convergence, and catastrophic self-overfitting (Cetin et al. 2022; Kostrikov, Yarats, and Fergus 2020). In light of these limitations, we employ the encoder proposed in RLAD, which incorporates techniques to mitigate these problems.

Image Encoder The image encoder is a convolution neural network consisting of approximately 0.65M parameters, which is significantly smaller in size compared to state-of-the-art methods in urban AD (see Table 2). We leverage image augmentations to regularize the value function and to increase the generalization (Yarats et al. 2021). Specifically, we apply color jittering, Gaussian blur, and random crop. At the end of each convolution encoder, we append an Adaptive Local Signal Mixing (A-LIX) layer (Cetin et al. 2022) to mitigate the catastrophic self-overfitting phenomenon. For the convolutional layers, we employed the Delta-Orthogonal initialization technique (Xiao et al. 2018), and for the linear layers, we employed the Orthogonal initialization technique (Saxe, McClelland, and Ganguli 2013).

The image encoder, f_i, can be formalized as $\boldsymbol{i}_t = f_i(\mathrm{aug}(\big[\{\boldsymbol{I}_{t-k}\}_{k=0}^{1}\big]))$, where aug corresponds to the image augmentation applied, and $\boldsymbol{i}_t$ corresponds to the latent representation of the stack of two consecutive images $\big(\big[\{\boldsymbol{I}_{t-k}\}_{k=0}^{1}\big]\big)$.

Waypoint Encoder To encode the waypoints we use WayConv1D (Coelho, Oliveira, and Santos 2023). This method leverages the 2D geometrical structure of the waypoints by applying 1D convolutions with a 2×2 kernel over the 2D coordinates of the next N waypoints. The process can be described as $\boldsymbol{w}_t = f_w(\boldsymbol{W}_t)$, where f_w corresponds to the WayConv1D, and $\boldsymbol{w}_t$ corresponds to the latent representation of the waypoints ($\boldsymbol{W}_t$).

Vehicle Measurements Encoder We apply directly an MLP to the vehicle measurements: $\boldsymbol{v}_t = f_v(\big[\{\boldsymbol{V}_{t-k}\}_{k=0}^{1}\big])$, where f_v is the MLP, and $\boldsymbol{v}_t$ corresponds to the latent representation of the concatenation of the vehicle measurements across two steps $\big(\big[\{\boldsymbol{V}_{t-k}\}_{k=0}^{1}\big]\big)$.

The latent representation of all the inputs ($\boldsymbol{h_t}$) is then obtained by concatenating the latent representation of each input: $\boldsymbol{h}_t = [\boldsymbol{i}_t \quad \boldsymbol{w}_t \quad \boldsymbol{v}_t]$. Throughout the document, to simplify the notation, we will refer to all encoders f_i, f_w, and f_v as $f_{i,w,v}$:

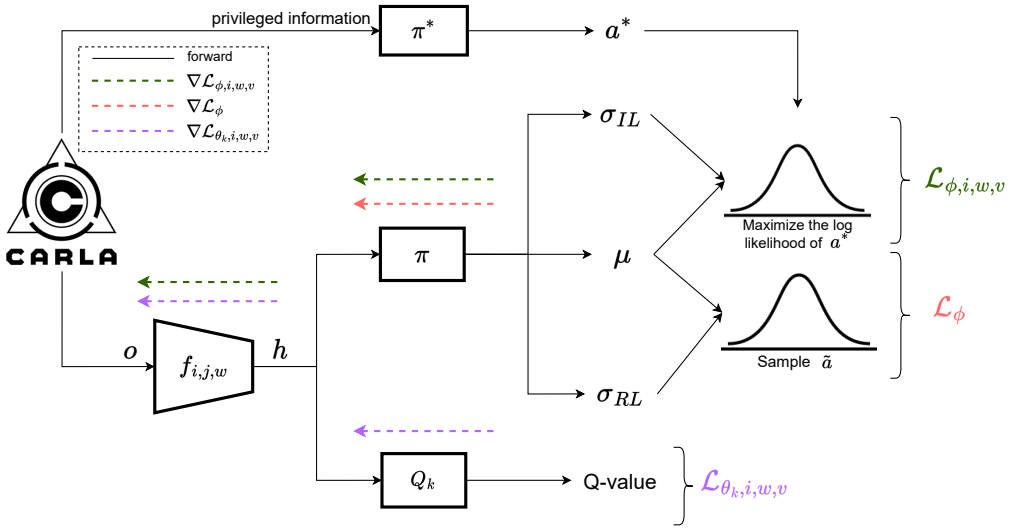

Figure 1: RLfOLD leverages online demonstrations through an expert policy (π^*) with access to privileged information. The encoder ($f_{i,j,w}$) converts the observation (o) into a latent representation ($\boldsymbol{h}$), which serves as input for a modified SAC. The policy (π) outputs the mean (μ) and two standard deviations: σ_{IL} and σ_{RL}. σ_{IL} is used to maximize the log-likelihood of the expert action (a^*), while σ_{RL} is employed to explore the environment.

$$\boldsymbol{h}_t = f_{i,w,v}\left(o_t\right). \tag{1}$$

3.3 Soft Actor-Critic With Imitation Learning

We use the SAC algorithm with a one-step return as the base algorithm. SAC is a model-free off-policy actor-critic algorithm that learns two Q-functions $Q_{\theta_1}, Q_{\theta_2}$, a stochastic policy π_ϕ, and a temperature α to find an optimal policy by optimizing a γ-discounted maximum-entropy objective (Ziebart et al. 2008; Kostrikov, Yarats, and Fergus 2020). The actor policy $\pi_\phi(\tilde{a}_t \mid \boldsymbol{h}_t)$ is a parametric tanh-Gaussian that given $\boldsymbol{h}_t$, samples $\tilde{a}_t = \tanh\left(\mu(\boldsymbol{h}_t) + \sigma_{RL}(\boldsymbol{h}_t)\epsilon\right)$, where $\epsilon \sim \mathcal{N}(0,1)$, and μ and σ_{RL} are the parametric mean and standard deviation. For the target speed, we apply a post-processing transformation to scale the tanh output to the desired range.

In contrast to the original SAC algorithm, our adapted actor policy produces three values: μ, σ_{RL}, and σ_{IL}. This modification proved to be more adequate, as it enables us to utilize distinct standard deviations for the various losses functions (more details provided below).

The double Q-networks are learned by optimizing a one-step of the soft Bellman residual:

$$\mathcal{L}_{\theta_k,i,w,v} = \mathbb{E}_{\substack{o_t,a_t,o_{t+1}\sim\mathcal{D} \\ \tilde{a}_{t+1}\sim\pi_\phi(\cdot|\boldsymbol{h}_{t+1})}}\left[\left(Q_{\theta_k}\left(\boldsymbol{h}_t, a_t\right) - y\right)^2\right], \quad \forall k \in \{1,2\}. \tag{2}$$

with the TD target y defined as:

$$y = r_t + \gamma\Bigg(\min_{k=1,2} Q_{\bar{\theta}_k}\left(\boldsymbol{h}_{t+1}, \tilde{a}_{t+1}\right) - \alpha \log \pi_\phi\left(\tilde{a}_{t+1} \mid \boldsymbol{h}_{t+1}\right)\Bigg), \tag{3}$$

where γ is the discount factor, and $Q_{\bar{\theta}_1}$ and $Q_{\bar{\theta}_2}$ denote the target parameters of Q_{θ_1} and Q_{θ_2}, respectively. The policy is updated to maximize the expected future return plus the expected future entropy:

$$\mathcal{L}_\phi = -\mathbb{E}_{\substack{o_t\sim\mathcal{D} \\ \tilde{a}_t\sim\pi_\phi(\cdot|\boldsymbol{h}_t)}}\Bigg[\min_{k=1,2} Q_{\theta_k}\left(\boldsymbol{h}_t, \tilde{a}_t\right) - \alpha \log \pi\left(\tilde{a}_t \mid \boldsymbol{h}_t\right)\Bigg]. \tag{4}$$

Finally, the parameter α is automatically tuned over the training according to (Haarnoja et al. 2018).

To incorporate IL, we utilize the same batch of transitions used by RL and create a Gaussian distribution (p_ϕ) using the parameters generated by π, namely μ and σ_{IL}. Subsequently, this distribution is employed to maximize the log-likelihood of the action produced by the online expert (a^*):

$$\mathcal{L}_{\phi,i,w,v} = -\mathbb{E}_{o_t,a_t^*\sim\mathcal{D}}\Big[\log p_\phi\left(a_t^* \mid \boldsymbol{h}_t\right)\Big]. \tag{5}$$

By using different standard deviations, the algorithm can adapt to the varying levels of uncertainty in RL and IL. It allows the RL component to explore the state-action space more broadly (with a larger standard deviation), while the IL component can focus on imitating the expert's behavior more closely (with a smaller standard deviation). This adaptability to uncertainty can lead to a better balance between exploration and exploitation, and thus a seamless integration of RL and IL.

As illustrated in Figure 1, each loss updates specific parameters, and this document follows a nomenclature that associates the indexes of the loss function with the corresponding updated parameters.

3.4 Online Expert

The online expert has access to privileged information from the simulator, enabling it to generate expert actions. These actions serve two distinct purposes: assisting the exploration process and contributing to Equation 5. The online expert can take the form of a neural network or a set of heuristics. For this study, we have chosen to implement the online expert as a set of simple heuristics, with future plans to transition to a neural network-based approach.

As previously mentioned, the action is parameterized using target speed and steering. Inspired by (Toromanoff, Wirbel, and Moutarde 2019), the target speed is dynamically computed based on the agent's surroundings. As the distance to the front vehicle decreases, the target speed linearly reduces to 0, and conversely, as the distance increases, the target speed increases accordingly. The same principle applies when approaching obstacles, pedestrians, or traffic lights. For all other situations, the target speed remains set at a constant maximum speed. Regarding the steering, we conducted experiments using different heuristics that considered the agent's position and orientation relative to the waypoints. However, given the limited complexity of the online expert, we found that utilizing the steering from the RL policy (π) produced superior results. Consequently, in this work, we solely rely on the expert action to determine the target speed.

The efficacy of RLfOLD is substantially influenced by the quality of the online expert. The expert's input is crucial, providing accurate ground truth actions for IL and assisting in decision-making when the policy's uncertainty is high. While our dual standard deviation approach is designed to leverage this expert guidance effectively, it is important to acknowledge that the overall performance may vary with the expert's proficiency.

3.5 Expert-Guided Exploration Based on Uncertainty

In RLfD algorithms, the expert actions are only used in the loss functions to update the model parameters. However, by leveraging the online nature of the expert, we extended the usage of the expert actions to the exploration. The idea is to use the σ_{RL} as the uncertainty of the decision taken by the current policy (π). This uncertainty quantifies the confidence level of the policy, allowing us to gauge its competence in exploring the environment effectively. When the policy's uncertainty falls below a predefined threshold (u), the agent executes the action recommended by the policy, fostering efficient exploitation of its learned knowledge. On the other hand, if the policy's uncertainty exceeds the threshold, the agent seeks the guidance of the online expert to make informed decisions in uncertain situations. This decision-making process can be described as follows:

$$a = \begin{cases} \tilde{a} \text{ if } \sigma_{RL} < u \\ a^* \text{ otherwise} \end{cases} . \tag{6}$$

This method establishes a dynamic learning relationship between the agent and the online expert. Similar to a student seeking guidance from a teacher, the agent autonomously explores when confident, and seeks assistance from the expert when uncertain. This adaptive approach promotes efficient learning, safer exploration, and the potential for rapid skill acquisition in complex environments.

For a more comprehensive understanding of our learning framework, we provide the pseudocode implementation in Algorithm 1.

Algorithm 1: Reinforcement Learning from Online Demonstrations (RLfOLD)

Input: initial encoder parameters $f_{i,w,v}$, Q-function parameters Q_{θ_1}, Q_{θ_2}, policy parameters π_ϕ, entropy parameter α, empty replay buffer $\mathcal{D}$

$Q_{\bar{\theta}_k} \leftarrow Q_{\theta_k}$, for k = 1,2
repeat
  Get observation o_t
  Compute expert action a_t^* using π^*
  Encode o_t into $\boldsymbol{h_t}$ using Equation 1
  Sample policy action $\tilde{a}_t \sim \pi_\phi(\cdot \mid \boldsymbol{h_t})$
  Execute a_t according to Equation 6
  Get next observation o_{t+1} and reward r_t
  Store transition $(o_t, a_t, a_t^*, r_t, o_{t+1})$ in $\mathcal{D}$
  if o_{t+1} is terminal **then**
    Reset environment state
  end if
  if time to update **then**
    Randomly sample a batch of transitions, $\mathcal{B} = \{(o_t, a_t, a^*{}_t, r_t, o_{t+1})\}$ from $\mathcal{D}$
    Update Q_{θ_1}, Q_{θ_2} and $f_{i,w,v}$ using Equation 2
    Update π_ϕ using Equation 4
    Update π_ϕ, and $f_{i,w,v}$ using Equation 5
    Update α according to (Haarnoja et al. 2018)
    Update $Q_{\bar{\theta}_k}$ with
      $Q_{\bar{\theta}_k} \leftarrow (1-\rho)\, Q_{\bar{\theta}_k} + \rho Q_{\theta_k}$, for k = 1,2
  end if
until convergence

4 Experiments

4.1 Setup

Benchmark The algorithms are evaluated on the NoCrash benchmark. This benchmark examines the ability to generalize from Town 1, characterized by one-lane roads and T-junctions with traffic lights, to Town 2, a scaled-down version of Town 1 with different textures. The training process involves four distinct weather types, while the testing phase employs two different weather types. Within this benchmark, three levels of traffic density (empty, regular, and dense) are considered based on the number of vehicles and pedestrians present. The evaluation results are presented in terms of the success rate, representing the percentage of completed routes without any collisions. Additionally, for the ablation study, we also provide information regarding the percentage of route completion, collisions with vehicles, pedestrians, and layout, as well as the number of blockages per kilometer.

Training Details All algorithms are trained on the same hardware, specifically a single NVIDIA RTX 2080 Ti. The training process spans 10^6 environment timesteps, with evaluations conducted every 20k environment timesteps. During each evaluation, episode returns are averaged over 10 episodes. Each experiment was conducted with three different seeds to account for the high variability in RL training. We use the reward function defined in (Zhang et al. 2021). The Deep Learning library used was PyTorch (Paszke et al. 2019). Table 1 contains the main hyperparameters used by RLfOLD.

Parameter	Value
Replay Buffer capacity	100000
Batch size	128
Action repeat	2
Discount factor (γ)	0.85
Optimizer	Adam
Learning rate	10^{-3}
Target Q-network update rate (ρ)	0.01
$\dim(\boldsymbol{i})$	256
$\dim(\boldsymbol{w})$	32
$\dim(\boldsymbol{v})$	16
SAC networks size	1024
Init entropy parameter (α)	0.2
Uncertainty threshold (u)	0.8

Table 1: List of the hyperparameters used by RLfOLD.

State-of-the-Art Algorithms We compare RLfOLD with the state-of-the-art methods that reported their results on the NoCrash benchmark. The comparison includes algorithms of the three types (RL, IL, and RLfD):

- **RL**: IAs (Toromanoff, Wirbel, and Moutarde 2019), CADRE (Zhao et al. 2022);
- **IL**: CILRS (Codevilla et al. 2019), LBC (Chen et al. 2020);
- **RLfD**: GRIAD (Chekroun et al. 2021), WOR (Chen, Koltun, and Krähenbühl 2021);

4.2 Comparative Analysis

The number of parameters of a model is considered an essential metric to gauge computational requirements and memory consumption. However, obtaining this value can be challenging as it is often not reported in many studies. To address this, we use the size of the image encoder as a representative proxy (see Table 2). Since state-of-the-art methods typically employ very large image encoders, this component accounts for a substantial portion of the model's parameters. As reported in Table 2, RLfOLD utilized a significantly smaller encoder when compared to the state-of-the-art methods: approximately 3% of the average encoder size found in those methods. Table 2 also reports the number of cameras used, where all methods used only one camera, with the exception of GRIAD and WOR, which used three cameras.

	# of parameters	# of cameras
IAs	$\sim$30M	1
CADRE	$\sim$25M	1
CILRS	$\sim$22M	1
LBC	$\sim$22M	1
GRIAD	$\sim$14M	3
WOR	$\sim$22M	3
RLfOLD	$\sim$0.65M	1

Table 2: Comparison of the number of parameters in image encoders and the number of cameras used by the state-of-the-art methods.

Table 3 shows the comparative results in terms of the success rate on the NoCrash benchmark. The success rate values for the methods IAs, LBC, and WOR were obtained from (Chen, Koltun, and Krähenbühl 2021), the values of CADRE were taken from (Zhao et al. 2022), the values of CILRS were taken from (Zhao et al. 2021), and finally, the values of GRIAD were taken from (Chekroun et al. 2021). The proposed method outperforms all single-camera approaches across various tasks, showcasing its superior performance despite employing a significantly smaller encoder. Among the single-camera methods, CADRE emerges as the closest competitor, albeit with a notable 9% performance loss of the average score compared to RLfOLD. Even when compared against three-camera methods, all of which are RLfD algorithms, RLfOLD demonstrates its superiority in performance. With an average score exceeding GRIAD by 6% and WOR by 2%, RLfOLD outperforms its multi-camera counterparts. RLfOLD secures the top rank in more tasks than any other method, outperforming all competitors in seven distinct tasks. These results underscore the effectiveness and efficiency of RLfOLD, solidifying its position as the top-performing approach in the evaluation, even when employing a significantly smaller encoder and a single camera setup.

4.3 Ablation Study

To gain deeper insights into the strengths of RLfOLD, we conducted an ablation study examining its main components. Firstly, we established a RL baseline version without demonstrations (referred to as RL baseline). Next, we evaluated the significance of the two standard deviations by testing a variant of RLfOLD that generates only one standard deviation (σ_{RL}) and employs Mean Squared Error (MSE) loss for the IL training (RLfOLD w/o two SDs). Furthermore, to assess the impact of expert-guided exploration based on uncertainty, we experimented with two versions of RLfOLD: one that excludes expert guidance during exploration (RLfOLD w/o uncertainty (p=0.0)) and another that incorporates expert guidance with a fixed probability of 0.3 for each action taken (RLfOLD w/o uncertainty (p=0.3)). The results of the ablation study, as shown in Table 4, provide insights into the role of different components within RLfOLD. The RL baseline, which lacks the integration of demonstrations, achieved a success rate of 52%,

			RL		IL		RLfD		
Task	Town	Weather	IAs	CADRE	CILRS	LBC	GRIAD*	WOR*	RLfOLD
Empty	train	train	85	95	97	89	98	98	**100**
Regular			85	92	83	87	98	**100**	94
Dense			63	82	42	75	94	**96**	90
Empty	test	train	77	92	66	86	94	94	**100**
Regular			66	78	49	79	**93**	89	92
Dense			33	61	23	53	78	74	**80**
Empty	train	test	-	94	**96**	60	83	90	**96**
Regular			-	86	77	60	87	**90**	84
Dense			-	76	39	54	83	**84**	74
Empty	test	test	-	78	66	36	69	78	**100**
Regular			-	72	56	36	63	82	**86**
Dense			-	52	24	12	52	**66**	**66**
Average	-	-	68	80	60	60	83	87	**89**

* Used 3 cameras as input.

Table 3: Comparison of the success rate (%) on NoCrash benchmark using the state-of-the-art methods. The method IAs was not evaluated under testing weather conditions.

	Success rate %, ↑	Route completion %,↑	Collision pedestrian #/Km, ↓	Collision vehicle #/Km, ↓	Collision layout #/Km, ↓	Agent blocked #/Km, ↓
RL baseline	52±4	98±3	1.03±0.34	1.40±0.11	0.26±0.05	0.36±0.13
RLfOLD w/o two SDs	64±10	90±6	0.33±0.13	0.53±0.09	0.15±0.09	4.45±1.43
RLfOLD w/o uncertainty (p=0.0)	72±2	96±3	0.14±0.04	0.48±0.03	0.12±0.03	3.99±0.47
RLfOLD w/o uncertainty (p=0.3)	80±3	91±1	0.30±0.04	0.45±0.06	**0.00**±0.00	2.76±0.91
RLfOLD	**86**±4	**99**±2	**0.09**±0.03	**0.32**±0.04	0.09±0.03	**0.15**±0.08

Table 4: Ablation study evaluating the success rate and infraction analysis on the regular task under testing conditions (town and weather). Mean and standard deviation over 3 seeds.

which stands for a marginal loss of 34% considering the original version of RLfOLD. This difference clearly indicates the challenges of learning complex driving tasks using RL from scratch. The variant RLfOLD w/o two SDs demonstrates the importance of the two standard deviations. This variant achieved a success rate of 64%, which is significantly inferior to the one achieved using the two standard deviations - 86%. Furthermore, the integration of expert-guided exploration based on uncertainty proves to be highly beneficial. When we exclude expert guidance during exploration (RLfOLD w/o uncertainty (p=0.0)), the success rate drops to 72%. This indicates that the online expert provides valuable insights to guide the agent's exploration. Moreover, incorporating the online expert with a fixed probability for each action (RLfOLD w/o uncertainty (p=0.3)) achieves a success rate of 80%, which is 8% better than not using the online expert, but is 6% worse than using the expert-guided exploration based on uncertainty. In conclusion, the ablation study demonstrates the crucial role of each component in RLfOLD, emphasizing the significance of leveraging online demonstrations, the separate standard deviations output, and the expert-guided exploration based on uncertainty.

5 Conclusion

In this paper, we have presented RLfOLD, a novel and effective RLfD algorithm. Our method introduces a seamless integration of IL and RL by leveraging online demonstrations to bridge the distribution gap between the demonstration and the training environment. Unlike conventional policy networks used in actor-critic algorithms, RLfOLD adopts a policy network that outputs two standard deviations: one for exploration and another for IL training. Additionally, we utilize the online expert to guide the exploration process, incorporating an uncertainty-based technique. The results obtained on the NoCrash benchmark underscore the superior effectiveness and efficiency of RLfOLD. Notably, even with a significantly smaller encoder and a single-camera setup, RLfOLD surpasses all tested state-of-the-art methods. The ability to achieve such results with limited resources makes RLfOLD a highly promising solution for real-world applications. In the future, we aim to enhance the complexity of the online expert by transitioning from a rule-based approach to a more advanced neural network-based approach and to test this algorithm in other applications.

Acknowledgements

This work has been supported by FCT - Foundation for Science and Technology, in the context of Ph.D. scholarship 2022.10977.BD and by National Funds through the FCT - Foundation for Science and Technology, in the context of the project UIDB/00127/2020.

References

Cetin, E.; Ball, P. J.; Roberts, S.; and Çeliktutan, O. 2022. Stabilizing Off-Policy Deep Reinforcement Learning from Pixels. In *International Conference on Machine Learning*.

Chekroun, R.; Toromanoff, M.; Hornauer, S.; and Moutarde, F. 2021. GRI: General Reinforced Imitation and its Application to Vision-Based Autonomous Driving. *ArXiv*, abs/2111.08575.

Chen, D.; Koltun, V.; and Krähenbühl, P. 2021. Learning To Drive From a World on Rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15590–15599.

Chen, D.; and Krähenbühl, P. 2022. Learning From All Vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17222–17231.

Chen, D.; Zhou, B.; Koltun, V.; and Krähenbühl, P. 2020. Learning by Cheating. In Kaelbling, L. P.; Kragic, D.; and Sugiura, K., eds., *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, 66–75. PMLR.

Chitta, K.; Prakash, A.; Jaeger, B.; Yu, Z.; Renz, K.; and Geiger, A. 2022. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–18.

Codevilla, F.; Müller, M.; Dosovitskiy, A.; López, A. M.; and Koltun, V. 2017. End-to-End Driving Via Conditional Imitation Learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–9.

Codevilla, F.; Santana, E.; López, A. M.; and Gaidon, A. 2019. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9329–9338.

Coelho, D.; and Oliveira, M. 2022. A Review of End-to-End Autonomous Driving in Urban Environments. *IEEE Access*, 10: 75296–75311.

Coelho, D.; Oliveira, M.; and Santos, V. 2023. RLAD: Reinforcement Learning from Pixels for Autonomous Driving in Urban Environments. *arXiv preprint arXiv:2305.18510*.

Dey, S.; Pendurkar, S.; Sharon, G.; and Hanna, J. P. 2021. A joint imitation-reinforcement learning framework for reduced baseline regret. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3485–3491. IEEE.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In Levine, S.; Vanhoucke, V.; and Goldberg, K., eds., *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, 1–16. PMLR.

Goecks, V. G.; Gremillion, G. M.; Lawhern, V. J.; Valasek, J.; and Waytowich, N. R. 2019. Efficiently Combining Human Demonstrations and Interventions for Safe Training of Autonomous Systems in Real-Time. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 2462–2470.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic Algorithms and Applications. *ArXiv*, abs/1812.05905.

Hansen, N.; Lin, Y.; Su, H.; Wang, X.; Kumar, V.; and Rajeswaran, A. 2022. MoDem: Accelerating Visual Model-Based Reinforcement Learning with Demonstrations. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

Hester, T.; Vecerík, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Osband, I.; Dulac-Arnold, G.; Agapiou, J. P.; Leibo, J. Z.; and Gruslys, A. 2017. Deep Q-learning From Demonstrations. In *AAAI Conference on Artificial Intelligence*.

Kelly, M.; Sidrane, C.; Driggs-Campbell, K.; and Kochenderfer, M. J. 2019. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, 8077–8083. IEEE.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. *ArXiv*, abs/2004.13649.

Li, Q.; Peng, Z.; and Zhou, B. 2022. Efficient learning of safe driving policy via human-ai copilot optimization. *arXiv preprint arXiv:2202.10341*.

Liang, X.; Wang, T.; Yang, L.; and Xing, E. 2018. CIRL: Controllable Imitative Reinforcement Learning for Vision-based Self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Liu, H.; Huang, Z.; Wu, J.; and Lv, C. 2022. Improved Deep Reinforcement Learning with Expert Demonstrations for Urban Autonomous Driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, 921–928.

Lu, Y.; Fu, J.; Tucker, G.; Pan, X.; Bronstein, E.; Roelofs, B.; Sapp, B.; White, B.; Faust, A.; Whiteson, S.; et al. 2022. Imitation Is Not Enough: Robustifying Imitation with Reinforcement Learning for Challenging Driving Scenarios. *arXiv preprint arXiv:2212.11419*.

Menda, K.; Driggs-Campbell, K.; and Kochenderfer, M. J. 2019. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5041–5048. IEEE.

Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; and Abbeel, P. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, 6292–6299. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.;

and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Neural Information Processing Systems*.

Peng, Z.; Li, Q.; Liu, C.; and Zhou, B. 2022. Safe driving via expert guided policy optimization. In *Conference on Robot Learning*, 1554–1563. PMLR.

Prakash, A.; Chitta, K.; and Geiger, A. 2021. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7077–7087.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Saxe, A. M.; McClelland, J. L.; and Ganguli, S. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Shao, H.; Wang, L.; Chen, R.; Li, H.; and Liu, Y. 2023. Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer. In Liu, K.; Kulic, D.; and Ichnowski, J., eds., *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, 726–737. PMLR.

Toromanoff, M.; Wirbel, É.; and Moutarde, F. 2019. End-to-End Model-Free Reinforcement Learning for Urban Driving Using Implicit Affordances. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7151–7160.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Vecerik, M.; Hester, T.; Scholz, J.; Wang, F.; Pietquin, O.; Piot, B.; Heess, N.; Rothörl, T.; Lampe, T.; and Riedmiller, M. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*.

Wu, P.; Jia, X.; Chen, L.; Yan, J.; Li, H.; and Qiao, Y. 2022. Trajectory-guided Control Prediction for End-to-end Autonomous Driving: A Simple yet Strong Baseline. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 6119–6132. Curran Associates, Inc.

Xiao, L.; Bahri, Y.; Sohl-Dickstein, J.; Schoenholz, S.; and Pennington, J. 2018. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, 5393–5402. PMLR.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021. Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning. *ArXiv*, abs/2107.09645.

Zhang, Z.; Liniger, A.; Dai, D.; Yu, F.; and Van Gool, L. 2021. End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15222–15232.

Zhao, A.; He, T.; Liang, Y.; Huang, H.; Broeck, G. V. d.; and Soatto, S. 2021. SAM: Squeeze-and-Mimic Networks for Conditional Visual Driving Policy Learning. In Kober, J.; Ramos, F.; and Tomlin, C., eds., *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, 156–175. PMLR.

Zhao, Y.; Wu, K.; Xu, Z.; Che, Z.; Lu, Q.; Tang, J.; and Liu, C. H. 2022. CADRE: A Cascade Deep Reinforcement Learning Framework for Vision-Based Autonomous Urban Driving. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3): 3481–3489.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.