Beyond the Map: Learning to Navigate Unseen Urban Dynamics Using Diffusion-Guided Deep Reinforcement Learning

Supplementary Material

Monu Nagar , Debasis Das
Indian Institute of Technology Jodhpur, India
{nagar.3, debasis}@iitj.ac.in

1 Importance of Diffusion learning in Autonomous Driving

We propose a Diffusion-guided Deep Reinforcement Learning (DGDRL) framework to learn a generalized policy (π^*) capable of making optimal decisions in non-deterministic environments. The role of diffusion model in our approach goes beyond the conventional Data Augmentation (DA) techniques. Conventional DA relies on deterministic transformations, which fail to capture the continuous nature of state transitions in dynamic environments, particularly when training an SAC policy for autonomous driving. In our framework, DA is applied at the encoder layer, and the diffusion model is utilized to guide the policy network toward learning a generalized solution. The key motivation for integrating the diffusion model into guiding the policy (π) lies in its ability to progressively refine noisy latent states (o_t) into their original forms. This refinement process enables smooth interpolation between states, allowing the policy model to learn more complex and robust state representations. Such smooth state transitions provide a novel mechanism for jointly training the policy network and diffusion model, effectively addressing state uncertainties. Now, let's theoretically explore the benefits of integrating a diffusion model with the DRL method. Given the policy loss function and its gradient:

$$\begin{aligned} L_\phi^\pi = -\mathbb{E}_{o_t\sim D}\Bigg[\mathbb{E}_{\tilde{a}_t\sim\pi_\phi(\cdot|o_t)}\Bigg(\min_{k=1,2} Q_{\theta_k}(o_t,\tilde{a}_t) \\ -\alpha\log\pi_\phi(\tilde{a}_t|o_t))\Bigg] \\ + \boldsymbol{\lambda}\mathbb{E}_{\boldsymbol{o_t'}\sim\text{Diffusion}(\boldsymbol{o_t})}\left[\mathbf{D}_{\text{KL}}\left(\boldsymbol{\pi_\phi}(\cdot|\boldsymbol{o_t})\|\boldsymbol{\pi_\phi}(\cdot|\boldsymbol{o_t'})\right)\right] \end{aligned} \tag{1}$$

$$\frac{\partial L_\phi^\pi}{\partial\phi} = \frac{\partial L_{\text{SAC}}}{\partial\phi} + \frac{\partial D_{\text{KL}}}{\partial\phi} \tag{2}$$

where:

- D_{KL} is the Kullback-Leibler divergence.
- λ is the regularization coefficient.
- $\pi_\phi(\cdot|o_t)$ is the policy distribution conditioned on state o_t.

For Gaussian policies, the KL divergence between two distributions $\pi_\phi(\cdot|o_t) \sim \mathcal{N}(\mu_{o_t},\sigma_{o_t}^2)$ and $\pi_\phi(\cdot|o_t') \sim \mathcal{N}(\mu_{o_t'},\sigma_{o_t'}^2)$ can be expressed as:

$$D_{\text{KL}_\phi} = \underbrace{\ln\frac{\sigma_{\hat{o}_t}}{\sigma_{o_t}}}_{I} + \underbrace{\frac{\sigma_{o_t}^2+(\mu_{o_t}-\mu_{o_t'})^2}{2\sigma_{o_t'}^2}}_{(II)} - \underbrace{\frac{1}{2}}_{(III)} \tag{3}$$

Where, μ_{o_t} and $\mu_{o_t'}$ are the means, $\sigma(o_t)^2$ and $\sigma(o_t')^2$ are the variances of the respective Gaussian policies. Now, let's perform the derivative with respect to each term.

$$\frac{\partial D_{\text{KL}}}{\partial\phi} = \frac{\partial}{\partial\phi}(I) + \frac{\partial}{\partial\phi}(II) + \frac{\partial}{\partial\phi}(III) \tag{4}$$

First term,

$$\frac{\partial}{\partial\phi}(I) = \frac{\partial}{\partial\phi}(\ln\frac{\sigma_{\hat{o}_t}}{\sigma_{o_t'}}) = \frac{1}{\sigma_{\hat{o}_t}}\frac{\partial\sigma_{\hat{o}_t}}{\partial\phi} - \frac{1}{\sigma_{o_t'}}\frac{\partial\sigma_{o_t'}}{\partial\phi} \tag{5}$$

Now, let's compute the derivative of the second term (II) with respect to ϕ.

$$\frac{\partial}{\partial\phi}(II) = \frac{\partial}{\partial\phi}\left(\frac{\sigma_{o_t}^2+(\mu_{o_t}-\mu_{o_t'})^2}{2\sigma_{o_t'}^2}\right) \tag{6}$$

Using the quotient rule, $\frac{\partial}{\partial\phi}\left(\frac{u}{v}\right) = \frac{u'v-uv'}{v^2}$,
where $u = \sigma_{o_t}^2 + (\mu_{o_t}-\mu_{o_t'})^2$ and $v = 2\sigma_{o_t'}^2$. we obtain:

$$\frac{\partial}{\partial\phi}(II) = \frac{1}{(2\sigma_{o_t'}^2)^2}\begin{bmatrix}\frac{\partial}{\partial\phi}\left(\sigma_{o_t}^2+(\mu_{o_t}-\mu_{o_t'})^2\right)\cdot 2\sigma_{o_t'}^2 \\ -\left(\sigma_{o_t}^2+(\mu_{o_t}-\mu_{o_t'})^2\right)\cdot\frac{\partial}{\partial\phi}\left(2\sigma_{o_t'}^2\right)\end{bmatrix}. \tag{7}$$

The derivative of the third term (III) is zero, as the derivative of a constant with respect to any variable is always zero.

$$\frac{\partial}{\partial\phi}(III) = 0 \tag{8}$$

Impact of Large Initial KL Divergence: In the early stages of training, when $\mu(o_t)$ and $\mu(o_t')$ differ substantially, the KL divergence becomes large. The higher value of KL divergence plays a critical role in learning an optimal policy by ensuring consistency between policy distributions across different states. As described in Eq. 1, the KL divergence term encourages the agent to minimize the distributional difference between the policy distribution $\pi_\phi(\cdot|o_t)$ at state o_t and

the policy distribution $\pi_\phi(\cdot|o'_t)$ following a diffusion process. This higher KL divergence guides the agent to take optimal actions consistent with previously learned policies, particularly when it encounters a new or uncertain state. Therefore, it helps to prevent deviations from prior knowledge and enhances generalization by maintaining a balance between exploration and exploitation, allowing the policy to adapt effectively to dynamic environments across various states. Additionally, it reduces the risk of prematurely converging to suboptimal solutions, ensuring long-term stability and robustness in the agent's decision-making process.

Decreasing KL Divergence Over Time: As the diffusion model learns over time, the difference between the states o_t and o'_t reduces. This causes the means (μ) and variances (σ) of each distribution for o_t and o'_t to converge. As a result, the gradient of the KL divergence with respect to the policy parameters, $\frac{\partial D_{\text{KL}}}{\partial \phi}$, tends towards zero. Hence, the following observations can be made:

$$(\mu_{o_t} - \mu_{o'_t}) \to 0 \tag{9}$$

$$\sigma_{o_t} \approx \sigma_{o'_t} \tag{10}$$

Substituting Eq. (9) into (7) yields:

$$\frac{\partial}{\partial \phi}(II) = \frac{1}{(2\sigma^2_{o'_t})^2} \begin{bmatrix} \frac{\partial}{\partial \phi}\left(\sigma^2_{o_t}\right) \cdot 2\sigma^2_{o'_t} \\ - \sigma^2_{o_t} \cdot \frac{\partial}{\partial \phi}\left(2\sigma^2_{o'_t}\right) \end{bmatrix}. \tag{11}$$

Using Eq. (10), let $\sigma^2_{o_t} = \sigma^2_{o'_t} = \sigma^2$. Substituting this into (11) gives:

$$\frac{\partial}{\partial \phi}(II) = \frac{1}{(2\sigma^2)^2} \begin{bmatrix} \frac{\partial}{\partial \phi}\left(\sigma^2\right) \cdot 2\sigma^2 \\ - \sigma^2 \cdot \frac{\partial}{\partial \phi}\left(2\sigma^2\right) \end{bmatrix}. \tag{12}$$

Simplifying the terms inside the brackets:

$$\frac{\partial \sigma^2}{\partial \phi} \cdot 2\sigma^2 - \sigma^2 \cdot 2\frac{\partial \sigma^2}{\partial \phi} = 0. \tag{13}$$

Thus, under the conditions $(\mu_{o_t} - \mu_{o'_t}) \to 0$ and $\sigma_{o_t} \approx \sigma_{o'_t}$, we have shown that:

$$\frac{\partial}{\partial \phi}(II) = 0. \tag{14}$$

As a result, the gradients with respect to the policy parameters ϕ, as highlighted in Eq. (3), also decrease to zero:

$$\frac{\partial}{\partial \phi}(I) \to 0, \quad \frac{\partial}{\partial \phi}(II) \to 0, \quad \text{and} \quad \frac{\partial}{\partial \phi}(III) \to 0. \tag{15}$$

This results in the KL divergence term gradually decrease to zero as training progresses, especially in the final stages of training:

$$\lim_{t \to \text{last episode}} D_{\text{KL}}\left(\pi_\phi(\cdot \mid o_t) \parallel \pi_\phi(\cdot \mid o'_t)\right) \approx 0.$$

Since the KL divergence term no longer significantly contributes to the gradient, the gradient of the policy loss with respect to ϕ simplifies to:

$$\frac{\partial L^\pi_\phi}{\partial \phi} = \frac{\partial L_{\text{SAC}}}{\partial \phi} + \frac{\partial D_{\text{KL}}}{\partial \phi} \approx \frac{\partial L_{\text{SAC}}}{\partial \phi}. \tag{16}$$

The reduction in the KL divergence term results in a weaker penalty on the differences between the policies at o_t and o'_t, allowing the policy to shift its focus more towards optimizing specific actions based on subtle differences in the Q-values, rather than merely maintaining consistency. This smooth transition from high to low regularization marks a gradual shift in the policy's learning approach: initially, the policy ensures it behaves reasonably across a wide range of states by exploring and generalizing, but as training progresses and confidence in the learned strategy increases, the policy becomes more specialized and focused on exploiting the optimal actions for particular states. This process enables the agent to balance exploration and exploitation effectively. Therefore, this process improves stability and performance in dynamic environments as the agent refines its decision-making process.

Table 1: Hyperparameter Details

Hyperparameter	**Value**
Learning Rate (lr)	0.001
α (lr)	0.00001
Batch Size	128
Grad Clip	100
Discount Factor (γ)	0.85
Tau (τ)	0.01
Fc Dim	1024
Epsilon (ϵ)	0.000001
Optimizer	Adam
Diffusion Model	
timesteps	1000
Input Diam	304
Fc Dim	512
Out Dim	304
Variance schedule min (β)	0.0001
Variance schedule max (β)	0.02
Vehicle Measurements Encoder	
Input size	(1,2)
Fc Dim	8
Output Dim	16
Waypoints Encoder	
Input size	(10, 2)
Fc Dim	20
Output Dim	32
Image Encoder	
Input Size	(3,256,256)
Output Dim	256

2 Additional Experimental Results

To evaluate the generalization capability of the proposed DGDRL method, we conducted additional experiments in three new towns (Town03 to Town05) and a new weather environment (MidRainSunset). The model was tested across all driving tasks under Empty, Regular, and Dense traffic conditions. The specific hyperparameters used during the training of the DGDRL model for all internal components are detailed in Table 1. Figure 2 illustrates the categorization based on

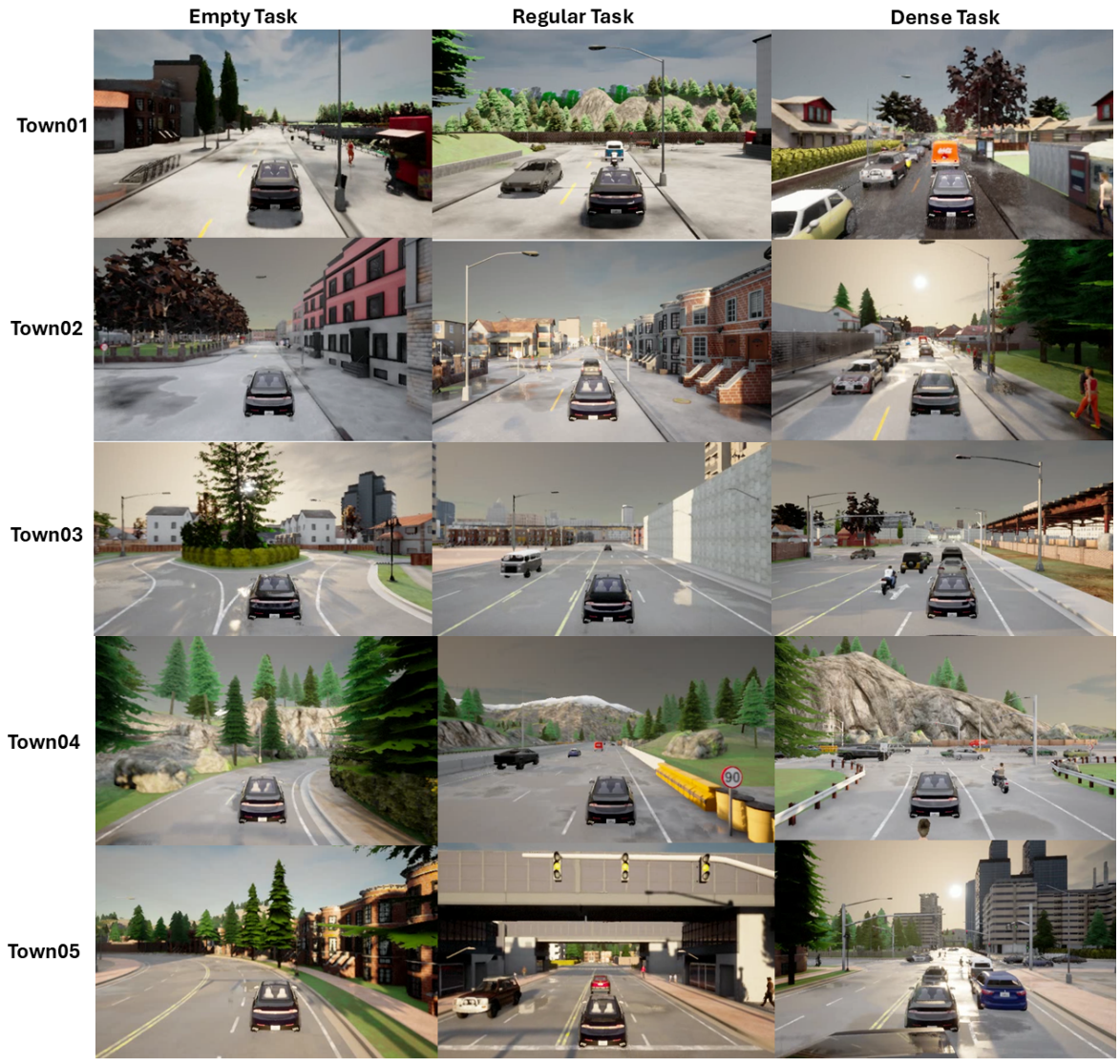


Figure 1: Visualizing the testing environment across all towns (Town01 to Town05) under test weather conditions.

the number of pedestrians (walkers) and vehicles on the road. Now, let's discuss the key features of all the Town maps used in the testing:

1. **Town01:** A small town featuring several T-junctions and a mix of residential and commercial buildings. The town is surrounded by coniferous forests and includes small bridges that cross a river dividing the town into two distinct sections.
2. **Town02:** A compact community town with multiple T-junctions and a variety of buildings. The town includes coniferous trees, park, blend of residential and commercial zones, creating a diverse urban environment.
3. **Town03:** A larger, downtown-style urban area with an extensive road network. Key features include a roundabout, residential areas, four-way and T-junctions, underpasses, overpasses, an elevated metro track, and a large construction site for a building.
4. **Town04:** A small town set against a scenic backdrop of snow-covered mountains and coniferous trees. The town is encircled by a multi-lane figure-of-eight ring road, with short streets and intersections connecting commercial and residential areas. The central crossover of the ring road includes an underpass and circular slip roads.
5. **Town05:** An urban area nestled among conifer-covered hills, featuring elevated highways and expansive multi-lane roads. The roads intersect at numerous large junctions, with dual-lane urban streets connecting intersections, creating a complex and dynamic environment.

We selected RLFOLD, the most recent and one of the top-performing models similar to our approach from our comparative analysis, to evaluate the learning stability of the DGRL method. Figure 2, compares the average accumulated rewards, critic's loss, and policy's learning loss. As shown in Figure 2c, the model initially exhibits higher loss due to the

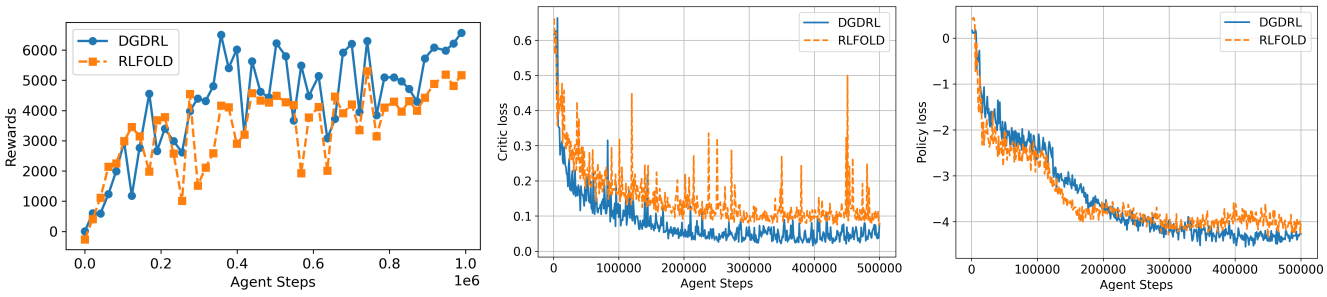


Figure 2: Comparative performance analysis between the proposed model and the most recent state-of-the-art method (RLFOLD). (a) Average rewards during evaluation (up to 10^6 steps), (b) Critic loss, and (c) Policy loss during training (up to 500K steps) versus agent steps.

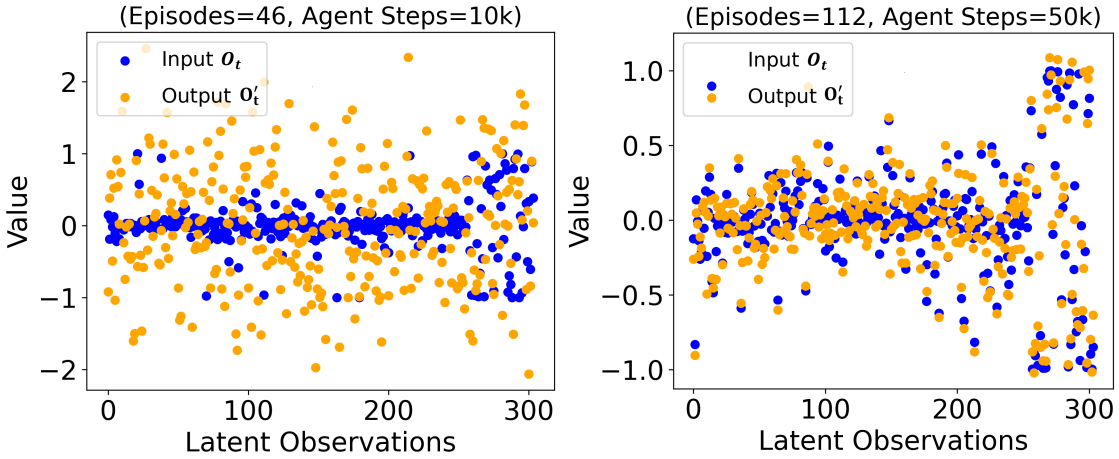


Figure 3: Comparison of input (O_t) and output (O'_t) latent observations across various training stages: observations at 10K steps and 50K steps. The results highlight the diffusion model's ability to enhance the policy network's capability to handle environment uncertainties.

Table 2: Number of Zombie Vehicles and Walkers for Different Background Traffic Conditions

Task	Zombie Vehicles	Zombie Walkers
Empty	0	0
Regular	20	50
Dense	100	250

diffusion learning process, with a notable gap between o_t and o'_t, as highlighted in Figure 3, at different agent steps. It is observed that after 125k environmental steps, RLFOLD fails to maintain optimal decisions, whereas the DGRL method continues to reduce its loss while consistently making optimal decisions. Furthermore, Figures 2a, and 2b illustrate the crucial role of the critic network in guiding the policy network. The critic helps prevent significant deviations in early decision-making, ensuring that the model makes consistent and optimal choices. As a result, the model is able to accumulate more rewards, demonstrating its strong ability to generalize across various weather and town scenarios. Additionally, we analyzed the behavior of the Q-network across different values of λ, as illustrated in Figures 5a, and 5b. These figures depict the average Q-network loss per step during the initial 50,000 training steps, providing insights into how the learning dynamics of the Q-network vary with changes in the importance of λ. When λ is set to 0.01, the Kullback-Leibler Divergence (KLD) loss has minimal influence on the policy network, resulting in instability in the Q-network's learning process. In contrast, increasing the weight of the KLD loss to 0.1 leads to significantly more stable learning behavior. These findings demonstrate the critical role of the diffusion model in stabilizing the learning process, enabling the agent to focus on optimal actions and consistently maintains positive rewards in each states.

2.1 Exploring Unseen States: Strengths and Challenges

We conducted a thorough study of the proposed approach, examining how the model performs in diverse environments and identifying its limitations in handling complex traffic scenarios. Figure 1 presents the testing results for tasks across five towns under the test weather conditions. The DGDRL framework shows a strong ability to adapt to new situations. During training, the model was placed in a controlled no-crash setup in the CARLA environment using the Town01 map, which has a simple two-lane road and does not require lane switch-

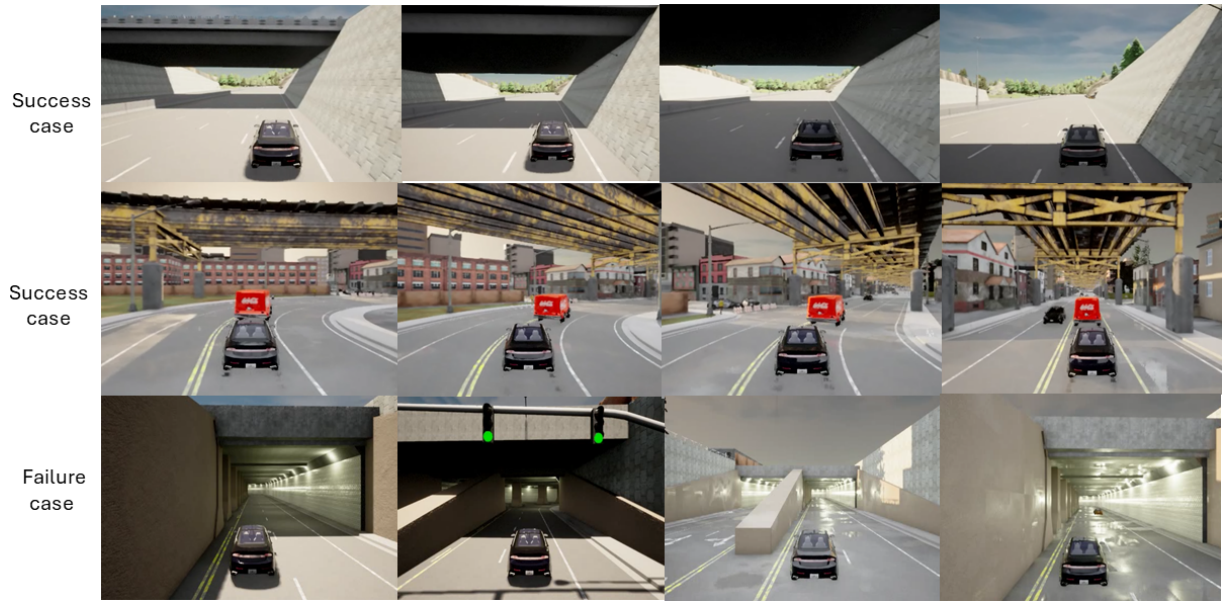
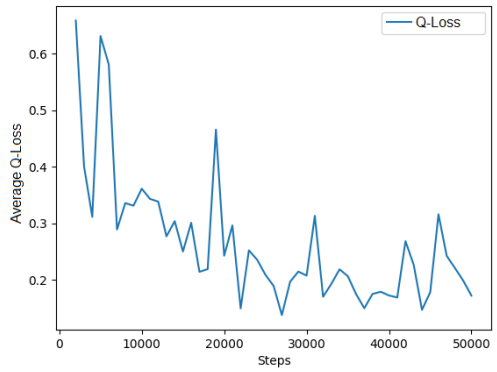
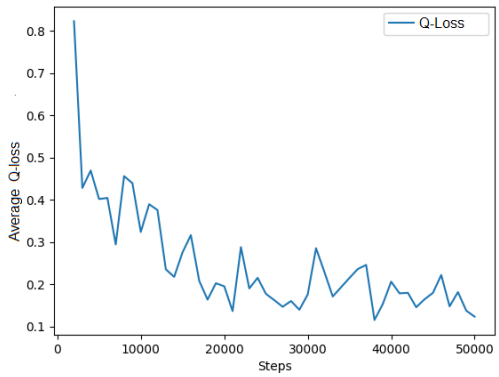


Figure 4: Visualization of the DGDRL model's success and failure cases while exploring unseen states in a new town. The top two rows demonstrate successful navigation in complex scenarios, such as underpasses. The bottom row illustrates a failure case where the model encounters a tunnel and fails to make a decision, halting its progress. These examples highlight the model's adaptability as well as its limitations in specific environments.



(a)



(b)

Figure 5: Comparison of Q-network learning loss curves across different λ values: (a) $\lambda = 0.01$ and (b) $\lambda = 0.1$. The curves depict the average Q-network loss per step during the initial 50k training steps, illustrating the effect of λ on learning dynamics.

ing. However, when tested on a more complex four-lane road, the model could perform lane changes, even though it had not been trained for this task. This behavior demonstrates the model's ability to understand and adjust to its environment, allowing it to handle dense traffic and solve tasks beyond its original training. We also tested the trained model on more complex routes from Town3, Town4, and Town5, which included tunnels, bridges, and underpasses. The results showed that the model successfully made optimal decisions when handling bridges and underpasses. However, when tested on tunnels, the model faced difficulties in interpreting them, leading to a failure in decision-making and causing the model to stop, as shown in Figure 4. Navigating through tunnels remains an open research challenge for improvement. Overall, the results show that the proposed scheme performs exceptionally well in unseen environments, as the model avoids making suboptimal decisions and prefers to stop rather than risk a collision.

2.2 Generalization Across Towns

The proposed model shows strong generalizability across different traffic conditions and towns, as seen in the performance statistics in Tables 3 and 4. In dense traffic scenarios (Table 3), the model successfully completed the route in Town1, Town2, Town4, and Town5 under W02 and W01 weather conditions without collisions or errors. Even in Town 3, which was more challenging due to a wide range of scenarios such as tunnels, bridges, and construction sites, the model still completed the route with a high score (0.925 under W02 and 0.678 under W01), showing its ability to handle difficult situations. Similarly, in regular traffic conditions (Table 4), the model performed well in Town1, Town2, and Town5 under both weather conditions. Though Town3 and Town4 had

Table 3: Performance statistics for CARLA NoCrash Dense traffic scenarios across Town1 to Town5 under test (W02) and train (W01) weather conditions, highlighting the optimal routes from each Town.

	Task: Dense, Weather: W02					**Task: Dense, Weather: W01**				
Parameters	**Town1**	**Town2**	**Town3**	**Town4**	**Town5**	**Town1**	**Town2**	**Town3**	**Town4**	**Town5**
Score Route	1.0	1.0	0.93	1.0	1.0	1.0	1.0	0.68	1.0	1.0
Reward	2504	2447	2179	5942	2341	2984	2531	3615	4136	3309
Timeout	0	0	0	0	0	0	0	0	0	0
Is Route Completed	1	1	0	1	1	1	1	0	1	1
Route Completed in km	1.08	0.6	1.16	2.50	1.26	1.08	0.68	2.39	2.50	1.64
Percentage Outside Lane	0	0	0	0	0	0	0	0	0	0
Percentage Wrong Lane	0	0	0	0	0	0	0	0	0	0
Collisions Layout	0	0	0	0	0	0	0	0	0	0
Collisions Vehicle	0	0	1.85	0	0	0	0	0	0	0
Collisions Pedestrian	0	0	0	0	0	0	0	0	0	0
Collisions Others	0	0	0	0	0	0	0	0	0	0
Light Passed	1	2	1	5	1	0	0	2	3	1
Encounter Light	3	6	3	16	12	3	6	8	16	4
Vehicle Blocked	0	0	0.05	0	0	0	0	0.08	0	0

Table 4: Performance statistics for CARLA NoCrash Regular traffic scenarios across Town1 to Town5 under test (W02) and train (W01) weather conditions, highlighting the optimal routes from each Town.

	Task: Regular, Weather: W02					**Task: Regular, Weather: W01**				
Parameters	**Town1**	**Town2**	**Town3**	**Town4**	**Town5**	**Town1**	**Town2**	**Town3**	**Town4**	**Town5**
Score Route	1.0	1.0	0.96	0.752	1.0	1.0	1.0	1.0	0.85	1.0
Reward	1939	1688	1711	2169	2351	2266	1512.951	1229	3725	2952
Timeout	0	0	0	0	0	0	0	0	0	0
Is Route Completed	1	1	0	0	1	1	1	1	0	1
Route Completed in km	1.07	0.59	1.12	1.37	1.32	1.07	0.68	0.73	2.19	1.67
Percentage Outside Lane	0	0	0	0	0	0	0	0	0	0
Percentage Wrong Lane	0	0	0	0	0	0	0	0	0	0
Collisions Layout	0	0	0	0	0	0	0	0	0	0
Collisions Vehicle	0	0	0	0	0	0	0	0	0	0
Collisions Pedestrian	0	0	0	0	0	0	0	0	0	0
Collisions Others	0	0	0	0	0	0	0	0	0	0
Light Passed	1	0	0	1	2	1	2	1	0	2
Encounter Light	3	5	4	8	12	3	6	3	0	4
Vehicle Blocked	0	0	0.08	0.02	0	0	0	0	0.05	0

some vehicle blockages, the model still managed to finish the routes with only a few disruptions. These results show that the model can adapt and perform well in various traffic situations. In terms of safety, the model consistently performed well by avoiding collisions. As shown in Tables 3 and 4, the model did not have any collisions with vehicles, pedestrians, or other objects in any of the towns, and there were no errors like driving outside the lane or in the wrong lane. In Town3 and Town4, the model successfully completed the routes with minimal blockages and no crashes. This focus on safe navigation in different traffic environments is a key strength of the model. Overall, the proposed model offers both strong generalizability in diverse conditions and a high level of safety, making it a reliable choice for autonomous driving.