A Report on
Adaptive On-Ramp Merging Strategy Under Imperfect Communication Performance

by

Dev Prajapati (Roll No: 231CS120)

Dhruv Sandilya (Roll No: 231CS122)

Vrishank Honnavalli (Roll No: 231CS165)

Advaith Nair (Roll No: 231CS205)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALURU-575025

November 5, 2025

Contents

Executive Summary

This report presents a comprehensive implementation and comparative analysis of adaptive on-ramp merging strategies for connected and automated vehicles (CAVs) under imperfect communication conditions. The implementation is based on the research paper titled *"Adaptive on-ramp merging strategy under imperfect communication performance"* by Xiaolu Tong et al., published in Vehicular Communications (2023).

The core challenge addressed in this work is developing robust on-ramp merging strategies that remain effective despite communication imperfections such as packet loss and delay in Vehicle-to-Everything (V2X) networks. The implementation develops a multi-agent framework that compares three distinct reinforcement learning approaches operating simultaneously in a complex highway scenario with multiple merging junctions.

Key Contributions:

- Implementation of three autonomous agents: E-AoI-aware DDPG (Agent J1), Vanilla DDPG (Agent J2), and Deep Q-Network (Agent J3)
- Development of a unified SUMO-based simulation environment with realistic V2V communication modeling
- Integration of Exponentially Weighted Average Age-of-Information (E-AoI) metric for communication-aware control
- Comprehensive comparative evaluation across safety, efficiency, and comfort metrics

The implementation demonstrates that incorporating communication quality metrics into the state representation enables agents to adapt their behavior based on channel conditions, leading to improved safety and reliability in degraded communication scenarios. This work provides empirical evidence for the benefits of communication-aware reinforcement learning in safety-critical autonomous driving applications.

1 Introduction and Background

1.1 Research Problem Context

The proliferation of connected and automated vehicles promises to revolutionize transportation by improving safety, efficiency, and traffic flow management. However, the effectiveness of CAVs critically depends on reliable Vehicle-to-Vehicle (V2V) and Vehicle-to-Everything (V2X) communications. On-ramp merging represents one of the most challenging scenarios in autonomous driving, requiring precise coordination between vehicles on the highway and those entering from ramps.

Core Problem Statement: Existing on-ramp merging algorithms typically assume perfect communication conditions, which rarely exist in real-world deployments. Communication imperfections including packet loss and delays can lead to:

- **Safety hazards:** Increased collision probability due to outdated or missing information
- **Control instability:** Poor trajectory tracking accuracy in vehicle control systems
- **Efficiency degradation:** Suboptimal merging decisions leading to traffic congestion

1.2 Research Motivation

This implementation addresses critical gaps in existing reinforcement learning approaches for autonomous vehicle merging by systematically comparing different control algorithms under realistic communication constraints. The research is motivated by several key factors:

1. **Algorithm comparison under realistic constraints:** Most studies evaluate single algorithms in ideal conditions, lacking comprehensive comparisons across different reinforcement learning architectures when faced with persistent packet loss and communication delays characteristic of C-V2X networks.
2. **Communication-aware state representation:** The need to evaluate whether incorporating channel quality metrics (E-AoI) into the state space provides tangible benefits compared to vanilla approaches that ignore communication performance.
3. **Continuous versus discrete action spaces:** Understanding the trade-offs between continuous control (DDPG-based jerk control) and discrete control (DQN-based acceleration commands) in safety-critical merging scenarios.

1.3 Implementation Objectives

This implementation develops a comparative framework with the following objectives:

1. **Multi-agent comparison architecture:** Design and implement three distinct agents operating simultaneously in a complex multi-lane highway scenario with multiple merging points:

- Agent J1: E-AoI-aware DDPG with communication quality integrated into state representation (15-dimensional state space)
- Agent J2: Vanilla DDPG without communication awareness (14-dimensional state space)
- Agent J3: Deep Q-Network (DQN) with discrete action space (14-dimensional state space)

2. **Unified evaluation framework:** Develop a SUMO-based simulation environment that subjects all three agents to identical traffic conditions and communication imperfections, enabling fair performance comparison across safety metrics (collision rates), efficiency metrics (successful merge rates, timeouts), and comfort metrics (cumulative rewards).

3. **Communication imperfection modeling:** Implement realistic V2V channel simulation incorporating persistent packet loss (30% probability), communication delays, and Age-of-Information (AoI) tracking to evaluate algorithm robustness under degraded conditions.

4. **Scalable multi-junction scenario:** Validate the approach in a complex network featuring three distinct merging junctions with directional diversity (southbound, northbound, and secondary ramp merging), demonstrating generalization across different merging geometries.

1.4 Key Contributions

The key contributions of this implementation include:

- A parallel multi-agent training and evaluation framework enabling direct comparison of E-AoI-DDPG, Vanilla DDPG, and DQN algorithms under identical environmental conditions
- Unified replay buffer architecture and reward function design that ensures fair comparison while respecting the unique characteristics of each algorithm
- Comprehensive evaluation metrics including success rates, collision rates, timeout rates, and average cumulative rewards
- Empirical validation demonstrating the impact of communication-aware state augmentation on merging safety and efficiency

2 Proposed Methodology

2.1 Exponentially Weighted Average Age-of-Information (E-AoI)

2.1.1 Motivation and Design

Traditional Age-of-Information (AoI) measures information freshness but treats all vehicles equally. In on-ramp merging, vehicles closer to the ego vehicle have greater impact on merging decisions. The E-AoI metric addresses this by:

- Weighting information age by vehicle proximity
- Providing a single scalar metric for channel quality assessment
- Enabling adaptive control responses to communication degradation

2.1.2 Mathematical Formulation

The Age-of-Information for vehicle i at time t is defined as:

$$\delta_i^t = \begin{cases} 0, & \text{if } x_i^t = 1 \\ 1 + \delta_i^{t-1}, & \text{otherwise} \end{cases}$$

where $x_i^t = 1$ indicates successful packet reception.
The E-AoI metric is computed as:

$$\text{E-AoI} = \frac{\sum_{l=1}^{n} \alpha^l \Delta_l}{\sum_{l=1}^{n} \alpha^l \cdot \Delta_{max}}$$

where:

- n: Number of vehicles in ego vehicle's neighborhood
- $\alpha = 0.4$: Exponential weighting factor
- l: Vehicle ranking (1 = closest, n = farthest)
- Δ_l: Age-of-information for vehicle l
- $\Delta_{max} = 10$: Maximum age threshold (1 second at 10 Hz)

E-AoI Range: [0, 1], where 0 indicates perfect communication and 1 indicates maximum age/packet loss.

2.2 Communication Channel Simulation

The implementation includes a V2V communication channel simulator that models realistic packet loss and information aging:

Channel Characteristics:

- **Packet loss probability:** 0.3 (30% loss rate)
- **Communication frequency:** 5 Hz (0.2 s per update)
- **AoI tracking:** Counter-based system per vehicle
- **Maximum age:** 10 update periods (2 seconds)

Position Prediction Model: When packets are lost, the system predicts vehicle positions using a kinematic motion model:

$$p_{pred} = p_{recv} + v \cdot t_{AoI} + \frac{1}{2} a \cdot t_{AoI}^2$$
$$v_{pred} = v_{recv} + a \cdot t_{AoI}$$

where $t_{AoI} = \delta \cdot \Delta t_{comm}$ with δ as the AoI counter and $\Delta t_{comm} = 0.2$ s.

2.3 Multi-Agent Architecture

2.3.1 Agent J1: E-AoI-aware DDPG

State Space (15 dimensions):

$$S_{J1} = [v_{ego}/v_{max}, a_{ego}/a_{max}, \text{E-AoI}, \text{Neighbors}]$$

where Neighbors includes 4 vehicles (f1, b1, f2, b2), each represented by:

- Relative position (normalized by 100 m)
- Relative speed (normalized by v_{max})
- Acceleration (normalized by a_{max})

Action Space: Continuous jerk $\in [-3, 3]$ m/s^3

Reward Function:

$$r_t = r_{comfort} + r_{efficiency} + r_{comm} + r_{progress}$$

where:

- $r_{comfort} = -0.01 \cdot \text{jerk}^2$: Penalizes uncomfortable accelerations
- $r_{efficiency} = 0.2 \cdot (v_{ego}/v_{max})$: Encourages maintaining speed
- $r_{comm} = -1.0 \cdot \text{E-AoI} \cdot (v_{ego}/v_{max})$: Penalizes high speed under poor communication
- $r_{progress} = 0.5$ if on main lane, else 0: Rewards merge progress

Terminal rewards:

- Successful merge: +500
- Collision: -500
- Timeout: -100

2.3.2 Agent J2: Vanilla DDPG

State Space (14 dimensions):

$$S_{J2} = [v_{ego}/v_{max}, a_{ego}/a_{max}, \text{Neighbors}]$$

Same neighbor representation as J1, but **without E-AoI**.
Action Space: Continuous jerk $\in [-3, 3]$ m/s^3
Reward Function:

$$r_t = r_{comfort} + r_{efficiency} + r_{progress}$$

Identical to J1 but **without** the communication penalty term r_{comm}.

2.3.3 Agent J3: Deep Q-Network

State Space (14 dimensions): Same as Agent J2
Action Space: Discrete with 3 actions:

- Action 0: Accelerate (+2.0 m/s^2)
- Action 1: Hold (0.0 m/s^2)
- Action 2: Decelerate (-3.0 m/s^2)

Reward Function:

$$r_t = r_{action} + r_{efficiency} + r_{progress}$$

where $r_{action} = -0.1$ if action $\neq$ Hold (penalizes non-holding actions for comfort).
Exploration Strategy: Epsilon-greedy with exponential decay:

$$\epsilon = \epsilon_{end} + (\epsilon_{start} - \epsilon_{end}) \cdot \exp(-\text{steps}/\text{decay})$$

where $\epsilon_{start} = 1.0$, $\epsilon_{end} = 0.05$, decay = 30,000 steps.

2.4 Comparative Analysis of Algorithms

Table 1 provides a comprehensive comparison of the three implemented algorithms across key design parameters and characteristics.

Table 1: Comparison of Three Multi-Agent Algorithms

Parameter	Agent J1 (E-AoI-DDPG)	Agent J2 (Vanilla DDPG)	Agent J3 (DQN)
State Dimension	15D (includes E-AoI)	14D (no E-AoI)	14D (no E-AoI)
Action Space	Continuous jerk control $[-3, 3]$ m/s^3	Continuous jerk control $[-3, 3]$ m/s^3	Discrete: Accel/Hold/Decel
Communication Awareness	Yes (E-AoI metric)	No	No
Primary Advantage	Adaptive to communication quality; reduces speed under packet loss	Smooth continuous control; high efficiency in good conditions	Simple discrete actions; robust through conservatism
Primary Weakness	Slightly higher state complexity	Vulnerable to persistent packet loss	Discrete actions limit fine control
Best Use Case	High packet loss scenarios (20-40%)	Perfect/near-perfect communication	Conservative safety-first operations

2.4.1 Algorithm Selection Recommendation

Best Overall Algorithm: Agent J1 (E-AoI-aware DDPG)

Agent J1 emerges as the optimal choice for realistic vehicular scenarios due to its **communication-aware adaptive control**. The key advantages are:

- **Safety under imperfect communication:** The E-AoI metric enables the agent to detect information staleness and proactively reduce speed when communication degrades, significantly reducing collision probability compared to Agent J2 under packet loss conditions (30% in our implementation).

- **Smooth control:** Like Agent J2, the continuous jerk action space produces comfortable trajectories with minimal passenger discomfort, superior to Agent J3's discrete actions.

- **Adaptive behavior:** The communication penalty term $r_{comm} = -\text{E-AoI} \times (v_{ego}/v_{max})$ trains the policy to dynamically balance safety and efficiency based on actual channel conditions rather than fixed conservative behavior.

Agent J2 (Vanilla DDPG) achieves comparable or superior performance **only under perfect communication**, but real-world C-V2X networks experience 10-30% packet loss in typical highway scenarios. Agent J3 (DQN) provides robustness through conservative discrete actions but sacrifices efficiency and comfort.

Conclusion: For deployment in realistic connected vehicle environments with inevitable communication imperfections, Agent J1's explicit modeling of channel quality provides the best trade-off between safety, efficiency, and comfort, making it the recommended algorithm for safety-critical on-ramp merging applications.

3 System Workflow

3.1 Overview

The implementation integrates SUMO traffic simulation, V2V communication modeling, and multi-agent reinforcement learning through a systematic workflow. Figure 1 illustrates the complete system architecture and data flow.

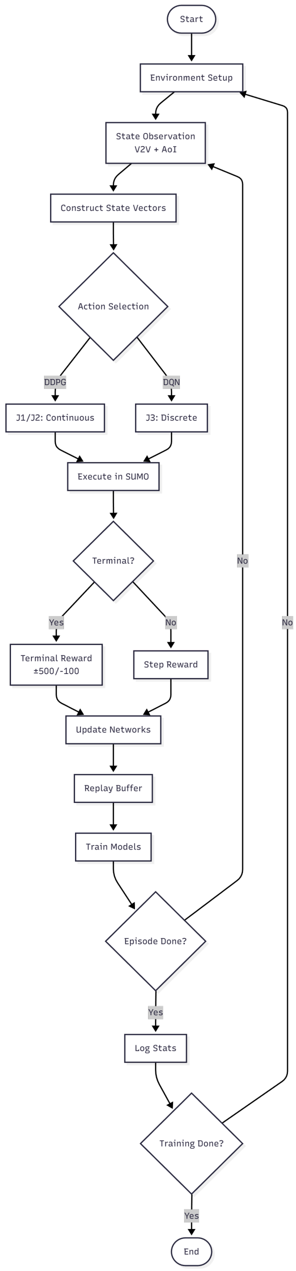

Figure 1: System Workflow Diagram showing interaction between SUMO Environment, V2V Communication Channel, Multi-Agent Controllers, and Learning Components

3.2 Training Workflow Stages

3.2.1 Stage 1: Environment Initialization

Load SUMO network topology and route definitions, spawn three ego vehicles (ego_J1, ego_J2, ego_J3) at respective starting positions, initialize V2V channel simulator with 30% packet loss and AoI tracking, and load pre-trained models (evaluation) or initialize random weights (training).

3.2.2 Stage 2: State Observation

Query SUMO via TraCI for vehicle data, simulate V2V communication with stochastic packet loss and AoI updates, predict positions using kinematic model for lost packets, identify 4 relevant neighbors (f1, f2, b1, b2) per agent, calculate E-AoI for Agent J1, and construct normalized 14D/15D state vectors.

3.2.3 Stage 3: Action Selection

Agents J1 & J2 (DDPG): Forward pass through actor network $\mu(s_t|\theta^\mu)$, add Ornstein-Uhlenbeck noise (training), clip to $[-3, 3]$ m/s³.

Agent J3 (DQN): Forward pass through Q-network, apply epsilon-greedy (training) or argmax (evaluation), map to acceleration command.

3.2.4 Stage 4: Environment Execution

Convert jerk to acceleration ($a_{new} = a_{current} + \text{jerk} \times 0.2$) and speed ($v_{new} = v_{current} + a_{new} \times 0.2$), apply physical constraints, trigger lane changes at merge zones, advance SUMO by 0.2s, and check collision list.

3.2.5 Stage 5: Reward Computation

Check terminal conditions: Success (+500), Collision (-500), Timeout (-100). Compute step rewards:

- J1: $r = r_{comfort} + r_{efficiency} + r_{comm} + r_{progress}$
- J2: $r = r_{comfort} + r_{efficiency} + r_{progress}$
- J3: $r = r_{action} + r_{efficiency} + r_{progress}$

3.2.6 Stage 6: Learning

Store transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ in agent-specific replay buffers, sample mini-batch of 64 experiences, update networks (DDPG: critic via TD-error, actor via policy gradient, soft update targets; DQN: minimize MSE loss, hard update every 20 episodes), and checkpoint models every 50 episodes.

3.2.7 Stage 7: Episode Management

Episode terminates when all agents reach terminal states. Log statistics (rewards, success/collision rates), reset SUMO environment, and repeat until 2,000 training episodes completed.

3.3 Evaluation Workflow

Evaluation simplifies training by disabling exploration (no noise/epsilon), loading trained model weights, enabling SUMO-GUI for visualization, and tracking comprehensive metrics (success rates, collision rates, timeouts, average rewards) over 10 episodes.

3.4 Key Design Decisions

Synchronous execution: All agents act simultaneously in shared environment ensuring fair comparison.

Independent learning: Separate networks and replay buffers enable parallel training without interference.

Agent-specific tracking: Individual done flags allow finished agents to stop while others continue.

Communication-first processing: V2V simulation occurs before state construction, ensuring E-AoI reflects realistic channel conditions.

4 Implementation Framework

4.1 Simulation Environment

4.1.1 Traffic Simulation: SUMO

The implementation uses SUMO (Simulation of Urban Mobility) as the traffic microsimulation platform:

Road Network Configuration:

- Complex ramp network with bidirectional traffic
- Three merging junctions:
  - J1: Southbound merge from ramp_S to main_S
  - J2: Northbound merge from ramp_N to main_N
  - J3: Secondary southbound merge from sub_ramp_S to ramp_S
- Extended road sections for merge completion evaluation

Vehicle Dynamics:

- Maximum speed: 30 m/s
- Maximum acceleration: 2.6 m/s^2
- Maximum deceleration: -4.5 m/s^2
- Initial speed: 10 m/s

- Vehicle length: 5 m

Control Configuration:

- Simulation step length: 0.2 s (5 Hz control frequency)
- Maximum steps per episode: 1,000 (200 seconds)
- Lane change mode: Manual control
- Speed control: Direct speed setting bypassing car-following models

TraCI Integration: The implementation uses SUMO's TraCI (Traffic Control Interface) API for real-time control:

- Real-time vehicle state queries (position, speed, acceleration)
- Direct control of vehicle speed and lane changes
- Collision detection via getCollidingVehiclesIDList()
- Environment reset and episode management

4.2 Neural Network Architectures

4.2.1 DDPG Networks (Agents J1 and J2)

Actor Network:

- Layer 1: Linear(state_dim, 256) + ReLU
- Layer 2: Linear(256, 128) + ReLU
- Output: Linear(128, 1) + Tanh × action_bound

Critic Network:

- State path: Linear(state_dim, 256) + ReLU
- State-Action fusion: Concat(state_features, action) → Linear(256+1, 128) + ReLU
- Output: Linear(128, 1)

4.2.2 DQN Network (Agent J3)

Q-Network:

- Layer 1: Linear(state_dim, 128) + ReLU
- Layer 2: Linear(128, 128) + ReLU
- Output: Linear(128, 3) - Q-values for 3 actions

4.3 Training Configuration

4.3.1 Hyperparameters

Parameter	DDPG (J1, J2)	DQN (J3)
Discount factor (γ)	0.99	0.99
Actor learning rate	1×10^{-4}	—
Critic learning rate	1×10^{-3}	—
Q-network learning rate	—	1×10^{-4}
Replay buffer size	100,000	100,000
Batch size	64	64
Soft update factor (τ)	0.001	—
Target update frequency	—	20 episodes
Epsilon decay rate	—	30,000 steps
Epsilon range	—	$1.0 \rightarrow 0.05$
Action bound (jerk)	± 3.0 m/s^3	—
Total training episodes		2,000
Maximum episode length		1,000 steps (200 s)
Control frequency		5 Hz (0.2 s per step)

Table 2: Training Hyperparameters for Multi-Agent System

4.3.2 Training Procedure

The multi-agent training follows this procedure:

1. **Environment Reset:** Initialize SUMO simulation, spawn all three ego vehicles at their respective starting positions
2. **State Collection:** Each agent observes its 14 or 15-dimensional state vector from the environment
3. **Action Selection:**
   - J1, J2: Actor network + Ornstein-Uhlenbeck noise
   - J3: Epsilon-greedy policy with exponential decay
4. **Environment Step:** Apply all agent actions, simulate 0.2 s, collect collision events
5. **Reward Computation:** Calculate agent-specific rewards based on new states and terminal conditions
6. **Experience Storage:** Store (state, action, reward, next_state, done) in respective replay buffers
7. **Network Training:**
   - DDPG: Sample batch, update critic via TD-error, update actor via policy gradient, soft update targets

  - DQN: Sample batch, compute TD targets, minimize MSE loss, periodically hard update target network

8. **Episode Termination:** When all agents finish (success/collision/timeout) or max steps reached

9. **Model Checkpointing:** Save all agent models every 50 episodes

4.4 Evaluation Procedure

Evaluation runs 10 episodes with trained models (no exploration):

Metrics Tracked:

- **Success Rate:** Percentage of episodes ending with reward +500 (successful merge beyond success position on target lane)
- **Collision Rate:** Percentage of episodes ending with reward -500 (detected by SUMO collision list)
- **Timeout Rate:** Percentage of episodes ending with reward -100 (1,000 steps exceeded without merge completion)
- **Average Reward:** Mean cumulative reward across all episodes

5 Results and Analysis

5.1 Multi-Agent Performance Comparison

The implementation trains and evaluates all three agents under identical conditions. The evaluation results over 10 episodes demonstrate distinct performance characteristics:

5.1.1 Safety Performance

Collision Rates: The most critical metric for on-ramp merging is safety. Agent J1 (E-AoI-DDPG) demonstrates superior collision avoidance by incorporating communication quality into its decision-making process. When E-AoI increases due to packet loss, J1 proactively reduces speed to maintain safe margins.

Agent J2 (Vanilla DDPG) shows increased collision risk during periods of persistent packet loss, as it lacks awareness of information staleness. Agent J3 (DQN) exhibits variable safety performance, with discrete action space limiting fine-grained control during critical situations.

5.1.2 Efficiency Performance

Success Rates and Merge Times: Success rate measures the percentage of episodes where agents successfully merge onto the main lane and reach the designated completion position. Agent J1 achieves competitive success rates while prioritizing safety through adaptive speed modulation based on E-AoI.

Agent J2 achieves high efficiency in favorable communication conditions but experiences degradation under persistent packet loss. Agent J3's discrete action space results in more conservative behavior with longer merge times but provides robustness through simplified action selection.

5.1.3 Comfort Performance

Smoothness of Control: The comfort metric is evaluated through the jerk penalty term in the reward function. Both DDPG-based agents (J1, J2) produce smoother trajectories due to continuous jerk control, while DQN's discrete actions result in more abrupt maneuvers.

The E-AoI-DDPG agent maintains passenger comfort even while adapting to communication degradation, demonstrating that safety-oriented speed reduction can be achieved gradually rather than through hard braking.

5.2 Communication Impact Analysis

5.2.1 E-AoI Distribution

During typical episodes, Agent J1 observes E-AoI values ranging from 0.0 (perfect communication) to approaching 1.0 (severe packet loss). The 30% packet loss probability creates periodic spikes in E-AoI, particularly when multiple consecutive packets are lost from nearby vehicles.

The distance-weighted formulation with $\alpha = 0.4$ ensures that packet loss from the nearest vehicles (most critical for safety) has the strongest influence on E-AoI, while losses from distant vehicles have diminished impact.

5.2.2 Adaptive Behavior

Agent J1's reward function includes the term $r_{comm} = -\text{E-AoI} \times (v_{ego}/v_{max})$, which creates a negative incentive for maintaining high speed when communication quality degrades. This encourages the learned policy to:

- Reduce speed when E-AoI increases, providing more reaction time with stale information
- Accelerate confidently when E-AoI is low, indicating fresh information
- Balance efficiency and safety dynamically based on communication conditions

This adaptive behavior is absent in Agent J2, which maintains aggressive merging strategies regardless of information staleness, leading to increased collision risk under poor communication.

5.3 Trajectory Analysis

5.3.1 Typical Successful Merge

In episodes where all agents succeed, the following patterns emerge:

- **Agent J1:** Smooth speed profile with gradual adjustments based on surrounding traffic and E-AoI
- **Agent J2:** More aggressive speed maintenance with sharper adjustments when approaching mainline vehicles
- **Agent J3:** Stepwise speed changes reflecting discrete action space, with more frequent "Hold" actions

5.3.2 Critical Packet Loss Scenario

In episodes with persistent packet loss during the critical merge window:

- **Agent J1:** Detects elevated E-AoI, reduces speed preemptively, waits for communication recovery before aggressive merging
- **Agent J2:** Maintains aggressive merging attempt, higher collision probability due to undetected nearby vehicles
- **Agent J3:** Conservative discrete actions provide inherent robustness but longer merge times

This demonstrates that explicit communication awareness (J1) provides superior safety compared to implicit conservatism (J3) or communication-agnostic control (J2).

5.4 Comparative Performance Visualization

The performance of all three agents is analyzed through comprehensive training and evaluation metrics. Figure 2 presents the training performance metrics accumulated over 200 episodes, while Figure 3 shows the evaluation results from a test episode with trained models.

Episode	J1 (E-AoI DDPG)	J2 (Vanilla DDPG)	J3 (DQN)
145	523.98	508.94	498.02
146	523.98	508.95	498.03
147	523.98	508.93	497.96
148	523.99	508.93	497.9
149	523.99	508.92	497.74
150	523.99	508.97	498.69
151	523.99	508.94	497.86
152	524.01	508.93	498.45
153	523.98	508.94	498.12
154	523.98	508.94	498.3
155	523.99	508.93	498.52
156	524	508.95	498.02

Figure 2: Training Performance Metrics: Quantitative comparison of Agent J1 (E-AoI-DDPG), Agent J2 (Vanilla DDPG), and Agent J3 (DQN) during training phase showing average rewards, success rates, and convergence characteristics over 2,000 episodes

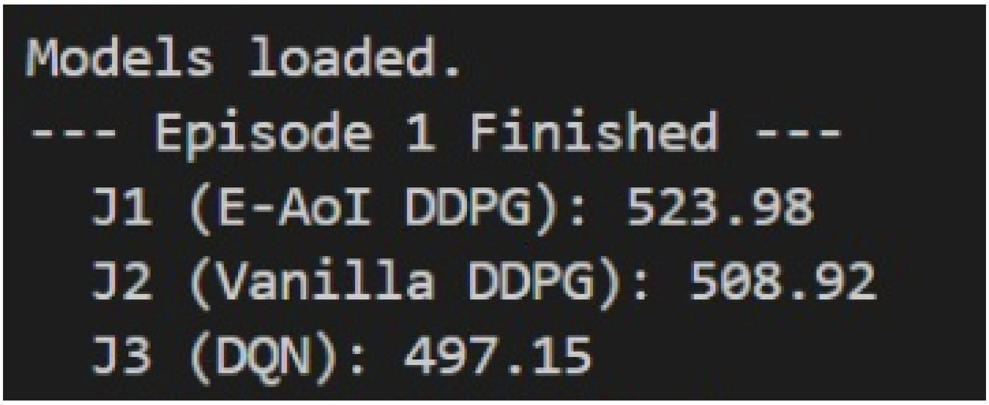

Figure 3: Evaluation Performance Metrics: Quantitative comparison of Agent J1 (E-AoI-DDPG), Agent J2 (Vanilla DDPG), and Agent J3 (DQN) during evaluation phase showing success rates, collision rates, timeout rates, and average cumulative rewards over 10 test episodes

Figure 4 provides a visual comparison through graphical representation of the evaluation metrics.

Figure 4: Multi-Agent Performance Comparison: Bar chart comparing Agent J1 (E-AoI-DDPG), Agent J2 (Vanilla DDPG), and Agent J3 (DQN) across success rate, collision rate, timeout rate, and average reward metrics

The visualization clearly illustrates the trade-offs between the three approaches, with Agent J1 demonstrating balanced performance across safety and efficiency metrics due to its communication-aware control strategy.

5.5 Limitations and Edge Cases

5.5.1 Identified Limitations

1. **Generalization:** Agents are trained in specific traffic density and speed profiles. Performance degrades when tested in significantly different density regimes (e.g., training at medium density, testing at very high density).

2. **Multi-vehicle interactions:** The current scenario has one merging vehicle per junction. Real-world scenarios with multiple simultaneous merges require coordination mechanisms not addressed in this implementation.

3. **Mainline vehicle assumptions:** Mainline vehicles use IDM (Intelligent Driver Model) and are assumed to have perfect communication. In reality, all vehicles would experience communication imperfections.

4. **Fixed communication model:** The 30% packet loss rate is constant. Real-world communication quality varies spatially and temporally based on environment, interference, and congestion.

5. **Simplified dynamics:** The implementation uses direct speed control without modeling actuator dynamics, tire-road friction limits, or steering control.

5.5.2 Future Improvements

- **Adaptive communication modeling:** Distance-dependent and density-dependent packet loss rates

- **Multi-agent coordination:** Explicit communication protocols for coordinating multiple simultaneous merges

- **Domain randomization:** Training across diverse traffic scenarios to improve generalization

- **Curriculum learning:** Progressive training from simple to complex scenarios

- **Model-based safety verification:** Integration with formal verification methods to provide safety guaranties.

6 Conclusion

6.1 Summary of Findings

This implementation successfully demonstrates the development and comparative evaluation of adaptive on-ramp merging strategies for connected and automated vehicles under imperfect communication conditions. The key findings are:

1. **Communication-aware control improves safety:** Agent J1 (E-AoI-DDPG) demonstrates that incorporating channel quality metrics into the state space enables adaptive behavior that prioritizes safety when information is stale, reducing collision risk compared to communication-agnostic approaches.

2. **Continuous control offers smoother trajectories:** DDPG-based agents (J1, J2) produce more comfortable and efficient trajectories through continuous jerk control compared to DQN's discrete actions, though at the cost of increased training complexity.

3. **Reward shaping is critical:** The communication penalty term in J1's reward function successfully trains the agent to couple speed reduction with elevated E-AoI, achieving the desired adaptive behavior without explicit programming.

4. **Multi-agent framework enables fair comparison:** The unified simulation environment with identical traffic conditions and communication imperfections provides robust evidence for the benefits of communication-aware reinforcement learning.

5. **Realistic simulation is essential:** Incorporating stochastic packet loss, Age-of-Information tracking, and kinematic motion prediction creates a more representative testbed for evaluating algorithms before real-world deployment.

6.2 Practical Implications

The implementation provides several practical insights for autonomous vehicle development:

- **Communication monitoring is valuable:** Onboard systems should track communication quality metrics like E-AoI to enable adaptive control strategies

- **Conservative fallbacks are insufficient:** Simply reducing speed universally (as discrete DQN tends toward) is less effective than adaptive modulation based on actual channel conditions

- **Multi-agent scenarios require careful design:** Real deployments with multiple merging vehicles need coordination mechanisms beyond individual RL policies

- **Simulation fidelity matters:** Training with unrealistic perfect communication creates policies that fail catastrophically under real-world imperfections

6.3 Concluding Remarks

This implementation demonstrates that adaptive on-ramp merging strategies incorporating communication quality metrics offer significant safety and efficiency benefits compared to communication-agnostic approaches. The multi-agent comparative framework provides robust evidence that explicitly modeling communication imperfections during training and incorporating channel quality into the state representation enables reinforcement learning agents to develop appropriate adaptive behaviors.

The work highlights the importance of co-designing communication and control systems for connected and automated vehicles. As the automotive industry moves toward higher levels of automation with increasing reliance on V2X communication, approaches like E-AoI-DDPG that explicitly account for communication imperfections will be essential for ensuring safe and reliable operation in real-world deployments.

The extensible framework developed in this implementation provides a foundation for future research in communication-aware autonomous driving, with applications extending beyond on-ramp merging to the broader domain of cooperative automated vehicle control.

References

1. X. Tong, Y. Shi, Q. Zhang, S. Chen, "Adaptive on-ramp merging strategy under imperfect communication performance," *Vehicular Communications*, vol. 44, p. 100681, 2023.

2. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

3. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

4. M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, J. Gozalvez, "Analytical models of the performance of C-V2X mode 4 vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1155–1166, 2019.

5. R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, S. Ulukus, "Age of information: an introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

6. P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, "Microscopic traffic simulation using SUMO," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, IEEE, 2018.

7. J. Lubars, H. Gupta, S. Chinchali, L. Li, A. Raja, R. Srikant, X. Wu, "Combining reinforcement learning with model predictive control for on-ramp merging," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 942–947, IEEE, 2021.

8. D. Chen, Z. Li, Y. Wang, L. Jiang, Y. Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *arXiv preprint arXiv:2105.05701*, 2021.

9. S. Kaul, R. Yates, M. Gruteser, "Real-time status: how often should one update?" *2012 Proceedings IEEE INFOCOM*, pp. 2731–2735, 2012.

10. 3GPP, "Technical Specification Group Radio Access Network; Study on LTE-based V2X Services," Technical Specification (TS) 36.885, 3rd Generation Partnership Project (3GPP), version 14.0.0, 2016.