**Near Accident Avoidance System - Technical Report**

**Executive Summary**

This project successfully developed a hierarchical reinforcement learning system for autonomous intersection navigation, achieving **94% success rate** with **zero collisions** in final evaluation. The system combines Conditional Imitation Learning (CoIL) for low-level control and Proximal Policy Optimization (PPO) for high-level strategy selection, demonstrating robust performance in near-accident scenarios.

**1. Introduction**

**1.1 Problem Statement**

Autonomous navigation through intersections presents significant challenges due to:

* Dynamic obstacle interactions
* Time-critical decision making
* Safety-critical consequences
* Multi-modal behavior requirements

**1.2 Objectives**

* Develop a hierarchical control system for intersection navigation
* Implement safe and efficient collision avoidance
* Enable adaptive behavior based on scenario risk
* Achieve human-level performance in simulation

**1.3 Technical Approach**

* **Hierarchical Architecture**: Separation of high-level strategy and low-level control
* **Imitation Learning**: Leverage expert demonstrations for safe behavior
* **Reinforcement Learning**: Optimize performance through trial and error
* **Multi-Modal Control**: Condition actions on driving style preferences

**2. System Architecture**

**2.1 Overall Design**

High-Level Policy (PPO) → Mode Selection (timid/normal/aggressive) → Low-Level Policy (CoIL) → Continuous Throttle Control → CARLO Environment → Observation & Reward

**2.2 Component Details**

**2.2.1 CARLO Environment**

Key parameters:

* ego\_start\_pos = 0.0
* ego\_start\_vel = 10.0
* ado\_pos\_range = (60.0, 100.0)
* crossing\_pos = 50.0
* success\_pos = 60.0
* max\_steps = 200

**Features:**

* Continuous state space (normalized positions and velocities)
* Continuous action space (throttle: -1 to 1)
* Time-to-collision based reward function
* Collision detection at intersection zone

**2.2.2 Expert Policy**

TTC-based decision making example:

* Compute ego TTC: (50.0 - ego\_pos) / ego\_vel
* Compute ado TTC: abs(ado\_pos - 50.0) / ado\_vel
* Mode-specific parameters:
  + timid → safe\_margin = 4.0
  + normal → safe\_margin = 2.0
  + aggressive → safe\_margin = 1.0

**2.2.3 CoIL Network Architecture**

Input (4) → Shared Layers (128→128) → Mode Branches → Output (1)

* Mode Conditioning applied to shared features

**2.2.4 PPO Agent**

* Actor-Critic with shared backbone
* Generalized Advantage Estimation (λ=0.95)
* Entropy Regularization for exploration
* Gradient Clipping for stability

**3. Methodology**

**3.1 Training Pipeline**

**Phase 1: Expert Data Generation**

* 600 episodes with balanced mode distribution
* 94% expert success rate achieved
* 33,450 training samples collected
* Validation split (80/20) for CoIL training

**Phase 2: CoIL Training**

* Behavior cloning from expert demonstrations
* Mean Squared Error loss function
* Adam optimizer (lr=1e-3)
* 15 epochs with early stopping

**Phase 3: PPO Training**

* 500 episodes of interaction
* Batch updates every 200 steps
* Entropy coefficient: 0.1
* Value coefficient: 0.5

**3.2 Reward Function Design**

* If collision → reward = -20.0
* If success → reward = 50.0 + (max\_steps - time\_step) \* 0.2
* Otherwise → reward = 0.1 \* (ego\_pos / success\_pos) + 0.05 \* (ego\_vel - 5.0)

**3.3 Hyperparameters**

**CoIL**

* Learning Rate: 1e-3
* Batch Size: 256
* Hidden Size: 128
* Epochs: 15

**PPO**

* Learning Rate: 1e-4
* Gamma: 0.99
* Lambda: 0.95
* Epsilon: 0.2
* Entropy Coefficient: 0.1

**4. Experimental Results**

**4.1 Training Performance**

**CoIL Training**

* Final Training Loss: 0.006569
* Final Validation Loss: 0.006253
* Training Time: ~2 minutes
* Model Size: 98KB

**PPO Training**

* Peak Success Rate: 94%
* Average Reward: 77.27
* Convergence: ~50 episodes
* Training Stability: High

**4.2 Final Evaluation (100 episodes)**

* Success Rate: 94 (94.0%)
* Collision Rate: 0 (0.0%)
* Timeout Rate: 6 (6.0%)
* Average Steps: 64
* Average Time: 6.42s

**4.3 Mode Usage Analysis**

* Timid: 26,465 uses (87.9%) → Early braking, conservative
* Normal: 2,144 uses (7.1%) → Balanced approach
* Aggressive: 1,491 uses (5.0%) → Late braking, assertive

**4.4 Ablation Studies**

**Without Hierarchical Structure**

* Success Rate: 68%
* Collision Rate: 22%
* Training Stability: Poor

**Without Expert Pre-training**

* Success Rate: 45%
* Training Time: 2x longer
* Sample Efficiency: Low

**5. Technical Innovations**

**5.1 Hierarchical RL for Autonomous Driving**

* Separation of temporal scales: strategic vs. continuous control
* Transfer learning: pre-trained low-level policies
* Multi-objective optimization: safety vs. efficiency trade-offs

**5.2 Conditional Imitation Learning**

* Mode-conditioned policies for different scenarios
* Shared feature extraction for parameter efficiency
* Smooth mode transitions with no abrupt changes

**5.3 Robust PPO Implementation**

* Stable advantage estimation with normalized GAE
* Adaptive entropy regularization for exploration
* Gradient clipping and normalization

**6. Discussion**

**6.1 Performance Analysis**

* Effective hierarchical decomposition
* High-quality expert demonstrations
* Proper reward shaping
* Robust training procedures

**6.2 Mode Selection Behavior**

* Predominant use of timid mode (87.9%)
* Safety-first approach
* Conservative strategy in uncertain situations
* Appropriate risk assessment at intersections

**6.3 Limitations and Challenges**

* Simulation-reality gap
* Fixed adversary behavior
* Computational requirements
* Generalization to unseen scenarios

**7. Conclusion**

**7.1 Key Achievements**

* High Success Rate: 94% crossings
* Perfect Safety Record: Zero collisions
* Efficient Learning: Rapid convergence
* Adaptive Behavior: Context-aware mode selection
* Technical Robustness: Stable training and evaluation

**7.2 Contributions**

* Novel Architecture: Hierarchical RL with conditional imitation
* Practical Implementation: End-to-end trainable system
* Comprehensive Evaluation: Rigorous testing
* Open Source Release: Code and models available

**7.3 Impact**

* Demonstrates hierarchical RL potential for autonomous driving
* Safety-critical scenario applications

**8. Future Work**

**8.1 Immediate Extensions**

* Multi-agent scenarios with multiple adversarial vehicles
* Real-world deployment and sim-to-real transfer
* Adversarial training for robustness testing
* Explainable AI for interpretability

**8.2 Long-term Directions**

* Urban driving with complex traffic rules
* Pedestrian interaction and prediction
* Weather and lighting condition adaptation
* Fleet learning from multiple vehicles

**8.3 Technical Improvements**

* Model-based RL for better sample efficiency
* Meta-learning for rapid adaptation
* Uncertainty quantification for risk assessment
* Multi-modal sensing integration

**9. References**

1. Codevilla, F., et al. "End-to-end driving via conditional imitation learning." ICRA 2018
2. Schulman, J., et al. "Proximal Policy Optimization Algorithms." arXiv:1707.06347, 2017
3. Kendall, A., et al. "Learning to drive in a day." ICRA 2019
4. Bansal, M., et al. "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst." RSS 2018
5. Mnih, V., et al. "Human-level control through deep reinforcement learning." Nature 2015

**Appendices**

**A. Computational Requirements**

* Training Time: ~15 minutes total
* Memory Usage: <1GB RAM
* Storage: <1MB model files
* Hardware: CPU-only training possible

**B. Code Availability**

* Repository: https://github.com/nvMANYA/Investigation-of-Near-accident-Car-driving-Scenario
* License: MIT Open Source
* Dependencies: PyTorch, NumPy, Matplotlib

**C. Reproduction Instructions**

1. git clone https://github.com/nvMANYA/Investigation-of-Near-accident-Car-driving-Scenario
2. cd near-accident-avoidance
3. pip install -r requirements.txt
4. python main.py