Imitation Learning For Practical Applications in Autonomous Intersection Navigation

Yung-Chi Kung
Walker Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas
0009-0004-5523-8021

Praneel Seth
Department of Computer Science
University of Texas at Austin
Austin, Texas
0009-0001-4665-0776

***Abstract*—This paper presents a novel approach to autonomous intersection navigation using transformer-based imitation learning for the Cadillac Lyriq electric vehicle. While conventional advanced driver-assistance systems primarily optimize for energy efficiency, our method prioritizes human-like comfort and smooth navigation in complex scenarios such as turns and parking maneuvers. The system employs dual transformer architectures for longitudinal and latitudinal control, trained on diverse driving data collected from multiple university teams participating in the EcoCar EV Challenge. Our evaluation demonstrates that the model successfully replicates human driving patterns while introducing improvements in control stability, particularly in reducing unnecessary steering oscillations during straight-line driving. The results show that transformer architectures can effectively balance the competing demands of accurate trajectory following and passenger comfort. This research contributes to the advancement of autonomous driving systems that maintain human-like behavior while optimizing for smoother vehicle control. Future work will focus on real-time implementation and validation in various practical driving scenarios.**

***Index Terms*—autonomous vehicles, transformers (artificial intelligence), imitation learning, electric vehicles, vehicle dynamics, torque control, neural networks, driver assistance systems, predictive models, vehicle navigation, real-time systems**

I. Introduction

The overall goal of the EcoCar EV Challenge is to enhance the energy efficiency of the Cadillac Lyriq. For the CAV subteam, designing an advanced driver-assistance system (ADAS) involves not only realizing greater energy efficiency gains when driving autonomously but also maintaining a superior level of human comfort. While ADAS functions such as Adaptive Cruise Control with lane centering typically prioritize energy efficiency gains, the challenge shifts in scenarios requiring Autonomous Intersection Navigation (AIN). Here, smooth navigation and human-like comfort take precedence over energy efficiency.

For the University of Texas at Austin (UT Austin) EcoCar EV Challenge team, we sought to explore the potential of transformer architectures to implement an imitation learning approach for AIN. This approach is crucial for tasks such as left turns, right turns, parking lot maneuvers, and other scenarios requiring a human-like driving behavior. Unlike conventional ADAS controls that optimize for energy efficiency, the imitation learning strategy targets human comfort by replicating the decisions of expert drivers.

The motivation behind using transformers lies in their ability to model complex temporal and spatial dependencies effectively. By leveraging attention mechanisms, we aim to create a system capable of predicting torque values with high precision, ensuring smooth and human-comfortable driving. This project demonstrates the transformative potential of machine learning in autonomous driving.

II. Related Works

Recent research in autonomous vehicle control explored various approaches, from end-to-end learning to modular architectures and reinforcement learning methods.

A. DeepPicar Platform

Most closely related was the DeepPicar platform by Bechtel et al., which implemented NVIDIA's DAVE-2 end-to-end learning architecture on a low-cost Raspberry Pi 3. While both the current work and DeepPicar used transformer-based architectures for autonomous control, the approaches differed in two key aspects: First, the current work separated longitudinal and latitudinal control into distinct transformer models, whereas DeepPicar used a single CNN for end-to-end control. Second, the focus was on intersection navigation rather than general road following, which presented different challenges for the control system [1].

B. Expert Demonstrations with Reinforcement Learning

Liu et al. proposed combining Deep Deterministic Policy Gradient (DDPG) with expert demonstrations and supervised loss for autonomous driving. Their approach introduced a supervised loss function to update actor networks while using reward construction to stabilize training [2]. While the current work also utilized expert demonstrations, the transformer-based architecture eliminated the need for explicit reward engineering, instead learning directly from human demonstrations through imitation.

Thanks to the Walker Department of Mechanical Engineering.

C. Modular ML Approaches

Perla et al. presented a more traditional modular approach using separate algorithms for road lane detection, anomaly detection, and disparity mapping. Their system achieved 97% accuracy for road lane detection and 98% accuracy for anomaly detection [3]. In contrast, the end-to-end transformer architecture learned these features implicitly, potentially reducing system complexity while maintaining comparable performance.

D. Unique Aspects

The current work advanced the state-of-the-art in several ways. Unlike DeepPicar's single CNN model [1], the dual transformer architecture allowed for specialized handling of longitudinal and latitudinal control. This separation enabled more precise control and better interpretability compared to end-to-end approaches. Additionally, while Liu et al.'s DDPG approach required careful reward engineering, this system learned directly from human demonstrations, simplifying the training process while maintaining robust performance [2].

III. System Overview

The autonomous driving system developed for the Cadillac Lyriq provided by General Motors leverages two transformer models to predict the torque values required for the vehicle's operation over short intervals. These models, the longitudinal transformer and the latitudinal transformer, are designed to work in tandem to output torque predictions for driving, braking, and steering. The system takes state information derived from Controller Area Network (CAN) data as its input, processes it to extract relevant features, and generates torque values to achieve a desired vehicle state.

The pipeline begins with the extraction of data from multiple university teams participating in the EcoCar EV Challenge. This CAN data is preprocessed into state information describing the vehicle's current and target positions, velocities, and heading differences. Further preprocessing transforms this state information into six key inputs that feed into the transformers. The longitudinal transformer predicts driving and braking torques across five intervals (spaced 0.1 seconds apart), while the latitudinal transformer predicts steering torques for the same intervals. The outputs are then used to calculate control actions for the next 0.5 seconds, ensuring safe and efficient navigation.

A consolidated dataset was created by preprocessing and merging data from multiple competition teams. This dataset includes input features and corresponding ground-truth torque values used for training and validating the models. The combined data ensures robustness and generalizability across diverse driving scenarios.

The CAN data used was sourced from the following university teams: the University of Waterloo, the University of Texas at Austin, the Georgia Institute of Technology, West Virginia University, and the University of Alabama.

Methodology

A. Data Pipeline

The data pipeline integrates datasets from six of the university teams participating in the EcoCar EV Challenge. Each team's dataset underwent independent preprocessing before being merged into a single comprehensive dataset. This process involved:

1) Standardizing input features across datasets.
2) Combining data into unified input and output files.
3) Splitting the data into training and validation sets, ensuring diversity in each subset.

B. Data Preprocessing

The preprocessing pipeline began by transforming raw CAN data into meaningful state information that serves as the foundation for feature extraction. This state information includes:

- Current state: The vehicle's current position, velocity, heading, and steering wheel angle.
- Target state: The desired end state, including target position, velocity, and heading.

Additional preprocessing is applied to generate six transformer input features:

- Target x-position relative to the current position.
- Target y-position relative to the current position.
- Current velocity of the vehicle.
- Target (final) velocity of the vehicle.
- Difference in heading between the current and target positions.
- Current steering angle.

For each timestamp (input state), the local origin is calculated, including position, velocity, and heading. For every input state, five different random time stamps (output states) within the next 1-3 seconds are chosen, effectively multiplying the size of the dataset by five. Relative positions and differences in heading are computed for each state pair. Driving torque, braking torque, and steering torque for the five states after the selected input state are extracted and combined to form ground truth vectors.

Preprocessing steps include normalization of input features to ensure numerical stability and consistency across varying driving scenarios. The processed data is split into one input file (since both models accept the same inputs) and two corresponding ground-truth output files for training the models. The input files consist of the six extracted features, while the ground truth files contain the actual torque values for longitudinal and latitudinal motion.

C. Dual Transformer Architecture

Initial preprocessing takes the raw CAN Data and converts it to a series of vehicle states (S) in 0.1 second interval time steps shown as S_0 through S_N in Fig. 1. These are processed to form the input vectors for both transformers, which are fed into each model.

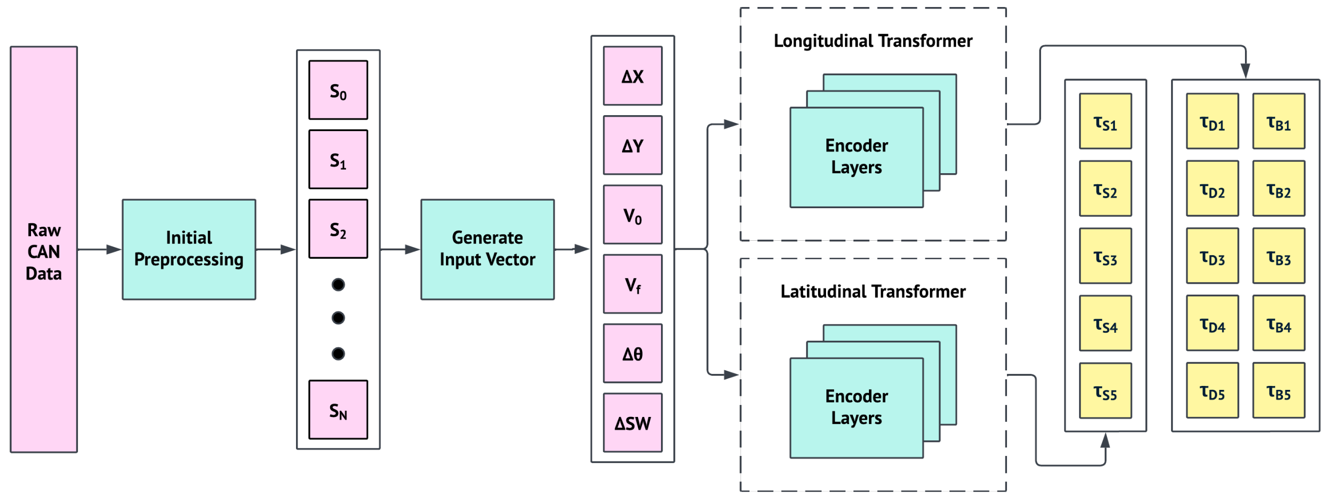

Fig. 1. Architecture Workflow.

The longitudinal model outputs 10 torque value (5 driving and braking torque pairs, labeled τ_{D1} through τ_{D5} and τ_{B1} through τ_{B5} respectively) while the latitudinal model outputs 5 steering torque values (labeled τ_{S1} through τ_{S5}).

The longitudinal transformer predicts driving torque and braking torque values for five intervals (0.1-second spacing) over the next 0.5 seconds. This model outputs ten nodes, representing the five pairs of torques:

- Number of encoder layers: 3
- Embedding dimension: 64
- Number of attention heads: 8
- Feedforward network dimensions: 256
- Dropout rate: 0.1
- Activation function: ReLU

The latitudinal transformer focuses on predicting steering torques for the same intervals, outputting five nodes:

- Number of encoder layers: 3
- Embedding dimension: 32
- Number of attention heads: 4
- Feedforward network dimensions: 128
- Dropout rate: 0.1
- Activation function: ReLU

D. Training Procedure

The training process was implemented in MATLAB. Both models shared the same hyperparameters and training configuration:

1) Data Preparation:
   - Input data and ground truth output data files were prepared. The input data had a size of [M, 6], and the output data had sizes of [M, 10] and [M, 5] for the longitudinal and latitudinal transformers respectively, where M is the number of samples.
   - Data was split into training (85%) and validation (15%) sets using random indexing.
   - Inputs and outputs were transposed and converted into dlarray format for compatibility with MATLAB's training workflow.

2) Hyperparameters and Configuration:
   - Loss function: Mean squared error (MSE)
   - Optimizer: Adam
   - Training hyperparameters:
     - Learning rate: 0.01
     - Batch size: Defined implicitly by data splitting
     - Number of epochs: 150
     - Learning rate schedule: Piecewise, with a drop factor of 0.9 every 10 epochs
     - Shuffle: Every epoch

3) Validation:
   - Validation data was provided to the training loop to monitor performance and save the model with the best validation loss.

E. Evaluation Metric

The dataset was divided into training and validation subsets, with proportions of 85% and 15% respectively, to ensure robust evaluation. The models were evaluated using the standard regression metric of Mean Squared Error (MSE) to assess their accuracy and reliability.

IV. Results

The transformer models demonstrated strong predictive capabilities across both longitudinal and latitudinal control. The longitudinal transformer achieved an MSE of 4.6535e+04 on the test set, while the latitudinal transformer achieved an MSE of 0.2346.

The driving and braking torque predictions from the longitudinal transformer showed remarkable alignment with ground truth values, particularly for the middle (third) predicted torque values as shown in the plot. The model successfully

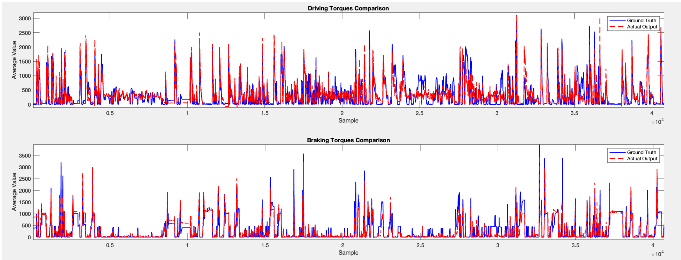

Fig. 2. Driving Torque and Braking Torque Comparison.

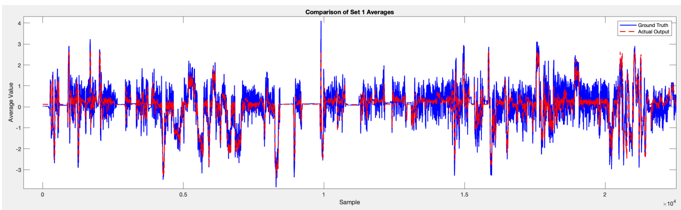

Fig. 3. Steering Torque Comparison.

captured both the timing and magnitude of necessary torque adjustments, indicating effective learning of driving patterns.

The latitudinal transformer's performance revealed an interesting pattern in steering control (also middle value shown in plot). While the model accurately predicted major steering events and turning maneuvers, it showed reduced oscillation during straight-line driving compared to human drivers. This reduction in micro-adjustments suggests the model learned to prioritize stable heading maintenance over continuous minor corrections.

V. Discussion

The evaluation results demonstrate several key findings about the transformer-based approach to autonomous vehicle control:

A. Longitudinal Control Excellence

The close alignment between predicted and actual torque values for both driving and braking indicates successful learning of complex driving behaviors. This suggests the transformer architecture effectively captures the temporal dependencies in vehicle dynamics.

B. Improved Steering Stability

The latitudinal transformer's tendency to reduce steering oscillations during straight-line driving represents an unexpected improvement over human behavior. While the model maintains accuracy during significant turning maneuvers, it exhibits more stable control during regular driving, potentially leading to smoother and more comfortable autonomous operation.

C. Human-Like Decision Making

The models' ability to replicate human driving patterns while simultaneously optimizing for smoother control aligns perfectly with the project's goal of balancing human comfort with efficient autonomous operation. This is particularly evident in the steering behavior, where the system maintains course stability without unnecessary adjustments.

VI. Conclusion

The transformer-based imitation learning approach has successfully demonstrated its capability to learn and reproduce human driving behavior while introducing subtle improvements in control stability. The models effectively balance the competing demands of accurate trajectory following and passenger comfort, particularly evident in the steering control's reduced oscillation during straight-line driving.

The strong performance in both longitudinal and latitudinal control suggests that transformer architectures are well-suited for autonomous vehicle control tasks. The system's ability to predict appropriate torque values while maintaining smoother operation than human drivers indicates potential benefits for both passenger comfort and vehicle efficiency.

Moving forward, the project will focus on implementing these models in a real-world context. The next phase involves porting the system from MATLAB to Simulink for real-time inferencing on the MATLAB Autera computer installed in the Cadillac Lyriq. This will be followed by comprehensive testing in various practical scenarios to validate the model's performance under actual driving conditions in preparation for competition.

References

[1] M. G. Bechtel, E. Mcellhiney, M. Kim and H. Yun, "DeepPicar: A Low-Cost Deep Neural Network-Based Autonomous Car," 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, Japan, 2018, pp. 11-21, doi: 10.1109/RTCSA.2018.00011.

[2] K. Liu, Q. Wan and Y. Li, "A Deep Reinforcement Learning Algorithm with Expert Demonstrations and Supervised Loss and its application in Autonomous Driving," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 2944-2949, doi: 10.23919/ChiCC.2018.8482790.

[3] S. Perla, N. N. K and S. Potta, "Implementation of Autonomous Cars using Machine Learning," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 1444-1451, doi: 10.1109/ICECAA55415.2022.9936102.