Demystifying Decision-Making of Deep RL through Validated Language Explanations

Ashwin Dara

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-51
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-51.html

May 13, 2025

Demystifying Decision-Making of Deep RL through Validated Language Explanations

By Ashwin Dara

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

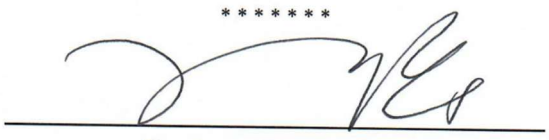
Professor Alexandre M. Bayen
Research Advisor

5/13/2025

(Date)

* * * * * * *

* * * * * * *



Professor Jiantao Jiao
Second Reader

May 12, 2025

(Date)

Demystifying Decision-Making of Deep RL through Validated Language Explanations

by

Ashwin Dara

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexandre Bayen (Advisor), Chair
Professor Jiantao Jiao (Second Reader)

Spring 2025

Abstract

Demystifying Decision-Making of Deep RL through Validated Language Explanations

by

Ashwin Dara

Master of Science in Computer Science

University of California, Berkeley

Professor Alexandre Bayen (Advisor), Chair

Reinforcement learning (RL) controllers have been shown in both simulation and real-world deployments to significantly improve traffic flow and fuel efficiency, even when only a small fraction of vehicles are autonomous. Despite these benefits, real-world adoption remains limited due to a lack of transparency, which leads human operators to distrust and often override RL policies. In response, we introduce CLEAR (Contextual Language Explanations for Actions from RL), a framework that generates step-by-step natural language explanations of RL decisions using large language models (LLMs). To address the risk of hallucinations in high-stakes settings, CLEAR integrates a multi-stage validation pipeline that verifies explanations against policy outputs, tests robustness under input perturbations, and checks for logical consistency. Unlike static fine-tuning methods, CLEAR adapts online to new scenarios and maintains alignment with the underlying policy. When evaluated on real-world highway data from the VanderTest, CLEAR significantly outperformed few-shot prompting and retrieval-based workflows in both predictive accuracy and explanation quality. This work extends a prior conference submission and demonstrates the potential of validated language-based interpretability for safe and trustworthy RL deployment.

Contents

Chapter 1

Introduction

1.1 Automated Vehicles and Traffic Smoothing

Analyses of traffic flow stability in highway settings reveal that human drivers inherently create conditions for phantom jams [4]. These are a subtle form of congestion caused by fluctuations in driver behavior [4]. Natural variability in acceleration and deceleration among vehicles creates local oscillations that can evolve into stop-and-go waves, propagating backward and amplifying over time due to a property known as string instability [7].

Due to their ability to exert precise, consistent, and coordinated control, connected autonomous vehicles (CAVs) have emerged in recent research as a scalable and promising solution for congestion mitigation [11]. Currently, autonomous vehicle deployment remains limited, with a small fraction of potentially fully autonomous vehicles sharing the road alongside basic automated systems such as cruise control and human drivers [11, 34]. This creates an environment of mixed autonomy, shifting the challenge toward leveraging minimal cooperation between AVs to enable traffic-smoothing behaviors.

Ongoing research continues to demonstrate strong potential for congestion reduction in such settings [34]. For example, simulations on the San Francisco Bay Bridge have shown that with only 5 percent AV penetration, where the vehicles followed reinforcement learning (RL) control policies, traffic flow improved by up to 20 percent [28, 29]. Building on these insights, the I-24 VanderTest field experiment in Nashville, TN deployed a mixed platoon of 4 AVs and 7 human-driven vehicles [16]. The study reported reduced traffic perturbations, resulting in an 11 percent decrease in overall energy consumption. This was followed by the MegaVanderTest, which scaled the experiment to 100 CAVs on the same section of I-24 [15]. The larger deployment further confirmed the potential of AVs to significantly improve energy efficiency and enhance overall traffic throughput.

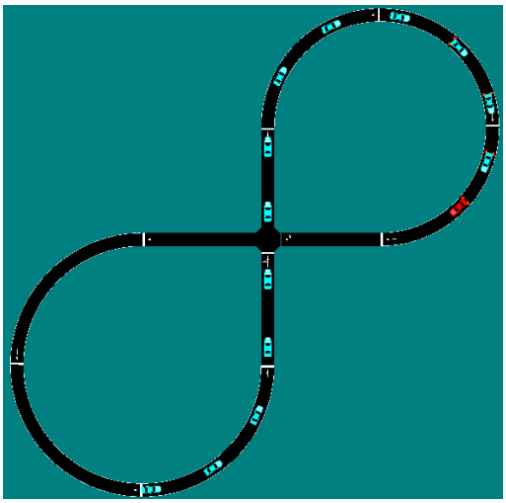


Figure 1: **Emergent Behaviors in RL-Based Traffic Smoothing.** Adapted from [33] In this figure-8 loop intersection scenario, a single reinforcement learning (RL)-controlled vehicle (purple) is trained to optimize traffic flow. The learned policy modulates its velocity to influence surrounding vehicles, reducing stop-and-go patterns and enabling wait-free traversal through the intersection.

Human-AV Challenges in Mixed Autonomy

Deployment of this real-world experiment, however, was significantly hindered by challenges related to the transparency of these specialized deep RL policies. Despite vehicle operators receiving extensive training, having access to failsafes, and possessing expert-level knowledge of the optimization objective, they still frequently disengaged the controller during the MegaVanderTest. On the first day of testing, engagement rates were only 38% [15]. Due to increased comfort and confidence with the system, this quickly rose to 78% [15]. Still, the initial hesitancy highlights a critical barrier to adoption: without interpretable reasoning behind decisions, even experts hesitate to trust and rely on autonomous systems.

Interpretability Challenges with RL

Current RL-based decision-making relies heavily on deep neural networks (DNNs) as the backbone of learned policies [22]. However, DNNs are often considered "black boxes" due to their lack of interpretability when generating actions in complex environments [9]. While recent advances in interpretability research have introduced post-hoc techniques to shed light on model behavior, these approaches fall short of enabling real-time trust in safety-critical settings [9]. In supervised learning, saliency maps [23] and mechanistic interpretability [3] have revealed internal activations and pathways that contribute to decisions, while transformer-based models have leveraged attention visualizations [38], although these are typically limited to early layers due to computational constraints. Similarly, tools like SHAP

[18] provide per-feature importance scores but do not offer a holistic view of the decision-making trajectory. Specifically for RL, interpretability has tended to focus on making sense of critical state-transition points in the trajectory sequence, typically identified through analyzing value functions [9]. However, these techniques remain far removed from human reasoning processes.

The core limitation of these approaches lies in their misalignment with how humans naturally reason [24]. People make sense of decisions through causal explanations, analogies to familiar experiences, and coherent statements rooted in prior knowledge [24]. Natural language interfaces offer a promising solution by translating complex model behavior into intuitive, human-readable explanations. Unlike visual or statistical tools, language-based explanations can express not just what the model did, but rationalize the decision, helping bridge between automated decision-making systems and human understanding in a form that aligns with how people naturally evaluate actions and intent.

CLEAR Framework

This paper introduces **CLEAR** (Contextual Language Explanations for Actions from RL), a framework developed in response to the deployment challenges observed during the Mega-VanderTest. CLEAR aims to provide insight into the decision-making and emergent behavior of reinforcement learning (RL) controllers. Our approach is built around two key goals: (1) utilizing the reasoning capabilities of large language models (LLMs) to generate rationales for decisions made by an oracle RL policy, and (2) deepening interpretability by simulating world dynamics under controlled, synthetic perturbations.

In high-stakes scenarios such as driving, hallucinations from LLMs can pose significant risks. To mitigate this, we introduce a series of validators, which are external LLMs equipped with tool-use capabilities that return explicit feedback. Specifically, our framework includes an accuracy validator, which ensures high prediction fidelity with respect to the RL controller's actions; a scenario validator, which uses simulation tools to validate physics-based outcomes; and a logical validator, which ensures inferential consistency in multi-step reasoning.

The main contributions of this paper are the following:

- We propose a general framework, CLEAR, that uncovers the decision-making process of an RL controller solely by observing state-action mappings. CLEAR outperforms current state-of-the-art LLM-based approaches by more than 40
- We demonstrate the self-improving capabilities of CLEAR over time. Our use of explicit validators ensures improvements in accuracy and mitigates the mode collapse commonly observed in traditional multi-agent LLM architectures during open-ended generation tasks.

- We evaluate CLEAR on real-world trajectory data from the I-24 dataset, which includes a diverse set of realistic driving scenarios. CLEAR consistently outperforms existing LLMs, retrieval-augmented generation (RAG) frameworks, and supervised fine-tuning (SFT) baselines.

Chapter 2

Preliminaries

2.1 RL for Decision-Making in Mixed Autonomy Control

Inspired by the way humans learn through trial and error, reinforcement learning (RL) has emerged as a powerful approach for sequential decision-making. RL algorithms have demonstrated impressive results across both continuous and discrete control tasks, including superhuman performance in games, applications in robotic manipulation, and control strategies in autonomous vehicles.

For mixed-autonomy traffic systems, where each autonomous vehicle (AV) operates based on partial, localized information, the problem is more accurately represented as a Partially Observable Markov Decision Process (POMDP). This is defined by the tuple $(S, A, T, R, \Omega, O, \gamma)$, which includes the traffic state space (S), action space (A), transition model (T), reward function (R), observation set (Ω), observation function (O), and discount factor (γ).

In this setting, each AV receives observations consisting of features such as the gap to the lead vehicle, its own velocity, and the relative velocity of the lead vehicle. Based on this partial view of the world, the RL policy selects control actions, typically acceleration commands, at each time step [14]. The VanderTest controller, used in previous deployments, adopted the POMDP framework to explicitly model uncertainty in vehicle observations [16]. The learning objective was to optimize the following reward function:

$$R_t = 1 - c_o E_t - c_1 a_t^2 - c_2 P_t$$

Here, E_t represents instantaneous fuel consumption, a_t is the acceleration term penalizing jerky motions, and P_t is a penalty for extreme space-gap values. The coefficients balance

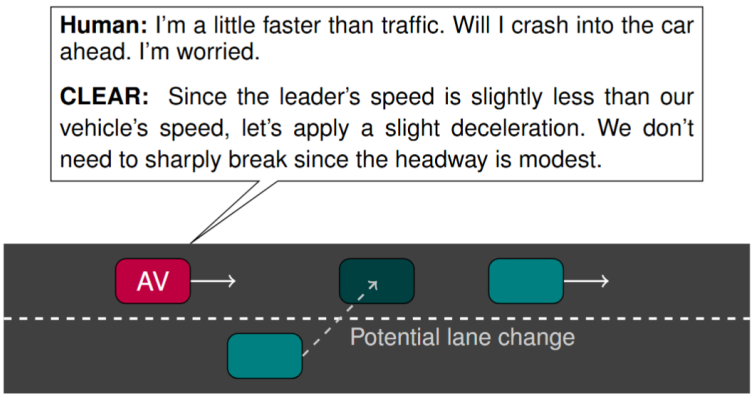


Figure 2: **User Concerns During the MegaVanderTest.** Operators of RL-controlled AVs reported discomfort caused by unusual space gaps compared to typical driving patterns. Several also raised concerns about potential dangers such as cut-ins, unpre- dictable lead vehicle behavior, and varying driving conditions [5]. CLEAR addresses these issues by providing real-time explanations for AV behavior, enhancing transparency and user trust.

these terms to encourage safe, smooth, and energy-efficient driving. The RL controller in VanderTest was trained using the Proximal Policy Optimization (PPO) algorithm [21].

Barriers to RL Deployment

Sim-to-Real Gap and Distribution Shift

Training reinforcement learning (RL) policies in simulation is often necessary due to the high cost, safety risks, and time constraints associated with real-world data collection [22]. Many RL algorithms are data-inefficient, requiring millions of interactions to converge on a robust policy [22]. Simulation offers a scalable and safe environment for generating diverse experiences at high throughput, enabling rapid prototyping and iteration without the risks of real-world deployment. This is particularly critical in domains such as robotics, autonomous vehicles, and industrial control, where physical-world exploration may be dangerous or prohibitively expensive.

However, deploying simulation-trained policies in the real world introduces the well-known simulation-to-reality (*sim-to-real*) gap [39]. This gap arises from distribution shift, which are mismatches between the simulated training environment and the real world in terms of dynamics, sensor noise, perceptual artifacts, and unmodeled edge cases. Policies may overfit to the biases of the simulator, and even minor discrepancies can lead to catastrophic failure upon deployment. The challenge is not just one of visual realism but of generalizing across structural and statistical differences that emerge between simulation and reality.

To mitigate this gap, several strategies have been proposed. Domain randomization introduces variability into the simulation to train policies that are robust across a wide range of conditions [27]. Progressive deployment incrementally increases autonomy in real-world operations, allowing systems to adapt to physical environments in a safe, staged manner [13]. Another approach leverages data augmentation by applying synthetic transformations to training data to encourage generalization and reduce overfitting to simulator artifacts. While these methods can significantly improve transfer performance, the sim-to-real problem remains fundamentally unsolved, and designing policies that generalize under distributional shift continues to be an active area of research.

Human Barriers in Real-World Deployment

Successful deployment requires systems that include humans in the loop and explicitly consider human understanding, preferences, and trust. Performance guarantees by themselves are not enough. The MegaVanderTest illustrated this clearly: deployment challenges stemmed not from poor controller performance, but from human discomfort with the decision-making process. The controller often behaved in ways that felt unfamiliar or opaque, leading to hesitation and reduced trust. To build dependable autonomous systems, it is essential to ensure that behavior is not only technically robust but also transparent, predictable, and aligned with human expectations.

2.2 Large Language Models (LLMs)

LLMs and Interpretable Interfaces

Large language models (LLMs) are uniquely suited to generate natural language explanations and support interactive, conversational queries [25]. This is especially valuable because language is the primary medium through which humans express abstract concepts and interpret other forms of information, including visual and sensory inputs. CLEAR leverages this strength to align with how people naturally communicate and develop understanding.

LLMs also enable real-time adaptation and personalization through in-context learning. This capability allows models to generalize to new tasks using just a few examples provided in the prompt, a technique known as few-shot prompting [31]. The flexibility stems from the models' pretraining on large-scale text corpora using next-token prediction, followed by instruction tuning that aligns their behavior with human intent. As a result, LLMs are a strong fit for frameworks like CLEAR that need to operate across a range of tasks without relying on task-specific fine-tuning.

To expand beyond the static knowledge embedded during pretraining, CLEAR integrates retrieval-augmented generation (RAG), which retrieves relevant documents at inference time [8]. This allows the system to handle specialized tasks that require external context beyond

what can be stored in a single prompt. Beyond text generation, LLMs can also rank and evaluate responses, enabling multi-agent workflows in which "judge" models help create feedback loops for continual improvement [2, 37, 6]. However, LLMs are known to exhibit certain biases, such as favoring their own completions, preferring longer outputs, or being sensitive to prompt structure. CLEAR addresses these issues by incorporating explicit validation steps that enforce correctness, consistency, and alignment with human reasoning.

LLMs Applications in Autonomous Driving

Multi-modal foundation models, particularly vision-language models (VLMs), are gaining momentum in autonomous driving due to their ability to handle both visual inputs and natural language [40, 5]. This dual-modality makes perception systems more transparent by allowing models to describe scenes, explain decisions, and support intuitive user interfaces [19]. Companies such as Waymo and Wayve are already deploying these models in production to enhance system debugging and address failures that arise from brittle or opaque perception components [19]. For instance, Waymo's EMMA [10] interface uses VLMs to deliver interpretable feedback to passengers, helping to build trust and promote safety during real-world operation.

Beyond perception, recent research has explored incorporating large language models (LLMs) directly into decision-making loops. Some approaches aim to replace conventional data-driven policies with knowledge-driven agents that reason over explicit rules, emulating human logic and commonsense reasoning [32, 12]. Others use LLMs to assist with auxiliary tasks such as action parameterization [35] and traffic context recognition [26]. What makes LLMs particularly compelling is their emergent reasoning ability. Their chain-of-thought style resembles how humans approach complex problems step by step, and their support for in-context learning allows them to adapt quickly from general knowledge with just a few guiding examples.

Chapter 3

CLEAR Architecture

3.1 Framework Overview

The CLEAR framework is built for deployment in unseen driving scenarios and is designed to improve over time through self-learning. When an autonomous vehicle encounters a new observation, it is first sent to the context cache, which contains previously refined explanations and driving scenarios. From there, the system follows two sequential steps. First, it queries the cache for past explanations that closely resemble the current observation, retrieving relevant few-shot examples. Second, a synthetic on-policy query is generated using a scenario generator that simulates how the policy would act in the present context. This step helps identify additional examples that are semantically similar to both the observation and the policy's expected behavior. These two sources of context work together to enhance the interpretability of the RL controller's decisions, particularly in rare or high-stakes situations.

The retrieved examples are assembled into a prompt for a standard commercial LLM, allowing for flexibility in model selection. The LLM then generates a natural language explanation that combines a step-by-step account of the RL policy's next action with broader predictions about how the scenario might unfold.

To enable continual improvement, CLEAR incorporates a Correctional Layer that refines the LLM's output. This layer consists of three validators: the accuracy validator, the scenario validator, and the logic validator. The explanation is first decomposed and evaluated in parallel by the accuracy and scenario validators, which use external tools to assess factual consistency and scenario-specific correctness. The refined outputs are then passed to the logic validator, which masks key inferential statements and tests whether the explanation remains internally coherent. If inconsistencies are identified, the explanation is revised accordingly. Once finalized, the revised explanation is stored back into the context cache. To maintain efficient deployment, the cache is periodically pruned by removing redundant entries. This is achieved by selecting evictions that maximize the entropy of the remaining embeddings,

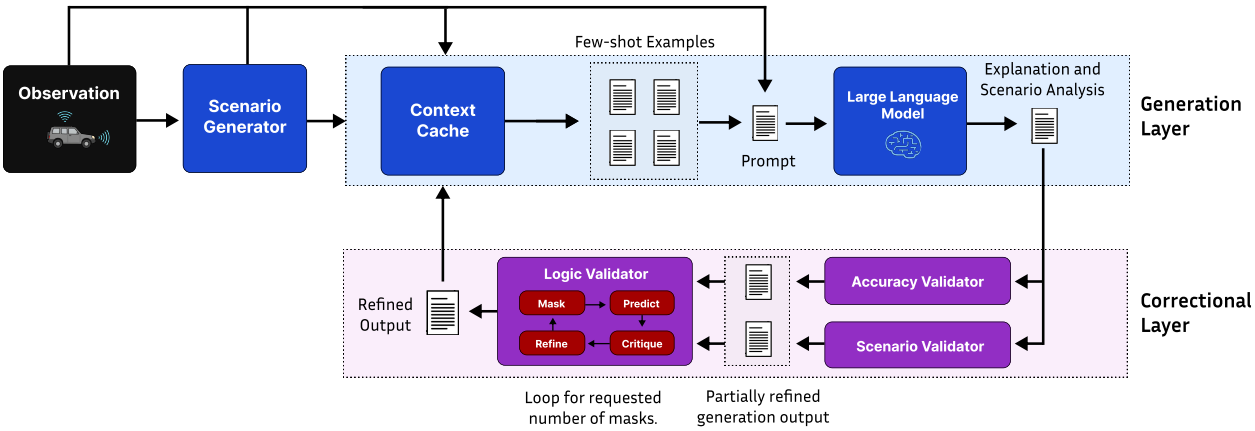


Figure 3: **Overview of CLEAR (Contextual Language Explanations for Actions from RL).** CLEAR consists of two components: the Generation Layer produces language explanations using a context cache of recent driving data, while the Correctional Layer refines each output for clarity and accuracy.

helping preserve a diverse and informative set of examples.

3.2 Generation Layer

Insights from the MegaVanderTest revealed several key interpretability requirements critical for real-world deployment:

- **Transparency**: Many vehicle operators indicated that their comfort would improve if the vehicle explicitly communicated both its intended actions and the reasoning behind them. Making decision-making more understandable was seen as essential.
- **Predictability**: Operators raised concerns about the consistency of the RL smoothing controller, particularly in response to deployment-time distribution shifts and abrupt environmental changes.
- **Confidence Awareness**: There was a consistent lack of feedback regarding the system's level of confidence or uncertainty, leaving operators unsure about the reliability of specific decisions.

To meet these needs, our system produces dual-purpose outputs: a rationale for the selected action and a simulation of hypothetical environmental changes generated by the scenario generator. This combination directly supports greater transparency and predictability. As a

final step, the logic validator acts as a diagnostic layer, identifying and correcting reasoning flaws even when the final decision is technically correct.

Context Cache

The context cache stores the autonomous vehicle's accumulated experiences during operation. Each entry is a tuple of four components: the raw observation captured from sensor data, a synthetically generated scenario, a refined explanation produced by the correctional layer, and validator feedback aggregated across all validation steps.

As more experiences are collected, the cache grows in both size and diversity, enhancing generation performance over time. During explanation generation, relevant past examples are retrieved from the cache and used as few-shot prompts to guide the output. To manage memory limits while preserving utility, an eviction module selectively removes redundant entries based on action similarity. In our setting, action dissimilarity tends to correlate strongly with observation dissimilarity. This property enables us to maintain a diverse and representative set of examples while staying within deployment constraints.

Scenario Generator

As described in Section 3.2, one of CLEAR's key interpretability features is allowing users to query the system about potential future events. To enable this, we use a scenario generator that introduces plausible hypothetical perturbations to the current observed state. These perturbations are drawn from common scenario archetypes such as lead vehicle braking, accelerating, maintaining speed, or performing a sudden cut-in maneuver.

The generator transforms each sampled perturbation into a natural language description, which is then embedded in the LLM prompt to help enrich the rationale with task-relevant dynamics. To validate the model's reasoning, each hypothetical scenario is simulated using a physics-based rollout. The simulation progresses in discrete time steps, updating the lead vehicle's behavior based on the scenario parameters and computing the ego vehicle's response using the original RL policy applied at each timestep.

3.3 Correctional Layer

Although human-in-the-loop evaluation is ideal, several practical limitations make it infeasible in our setting. Recruiting subject-matter experts is particularly difficult, given that our controller targets the specialized task of longitudinal flow smoothing. Additionally, manual annotation is costly and does not scale well. These challenges render human-in-the-loop feedback impractical for supporting self-correction during real-world deployment.

To address this, CLEAR relies on the self-evaluation capabilities of large language models (LLMs) through a structured correctional layer. However, LLMs on their own can exhibit convergence issues, such as continued flawed reasoning, false agreement, or collapsing into repetitive patterns. To prevent these failures, we introduce a set of specialized validators, each equipped with domain-specific tools such as physics engines or logic checkers. These validators independently assess key aspects of the explanation, including factual accuracy, logical coherence, and scenario consistency. Only after passing through this multi-stage refinement is the final explanation stored in the context cache. This tool-augmented, modular evaluation framework allows CLEAR to deliver scalable, high-quality refinement without relying on traditional multi-agent self-critique setups.

Accuracy Validator

This validatorensures that the language model's rationale aligns with the actual behavior of the RL controller. Rather than directly providing the ground-truth action, which can lead to the model rationalizing incorrect explanations, we first prompt the model to predict the controller's action based on the current state. This predicted action is then compared to the controller's true output through a forward pass. If a discrepancy is detected—such as a categorical mismatch or significant deviation—the validator appends corrective feedback and prompts the model to revise its explanation, ensuring it reflects the true decision.

This predict-then-rationalize process helps prevent fabricated justifications and strengthens the connection between reasoning and behavior. It also facilitates a quantifiable evaluation of explanation quality: when the model's initial prediction is incorrect, we can explicitly measure how well the revised rationale adapts to the true action. In this way, the Accuracy Validator serves both as a corrective mechanism and a diagnostic tool, assessing the alignment between LLM-generated reasoning and the controller's behavior.

Scenario Validator

This validator ensures the accuracy of the model's predictions about how the state evolves in hypothetical scenarios, specifically validating the (action, observation) pairs over time. These scenarios are incorporated into the prompt, assuming that the RL controller behaves on-policy under modified environmental conditions. Instead of relying on simple simulated numbers, the validator uses a learned dynamics model that embeds reasoning and simulates how actions influence state progression. This model acts as an oracle, enabling the LLM to simulate action-conditioned rollouts and compare the predicted outcomes with the simulated (action, observation) pairs. To further ensure alignment with both physical plausibility and true policy behavior, a rule-based verifier conducts an additional check on the revised explanation, confirming its consistency with the controller's actions and the simulated conditions.

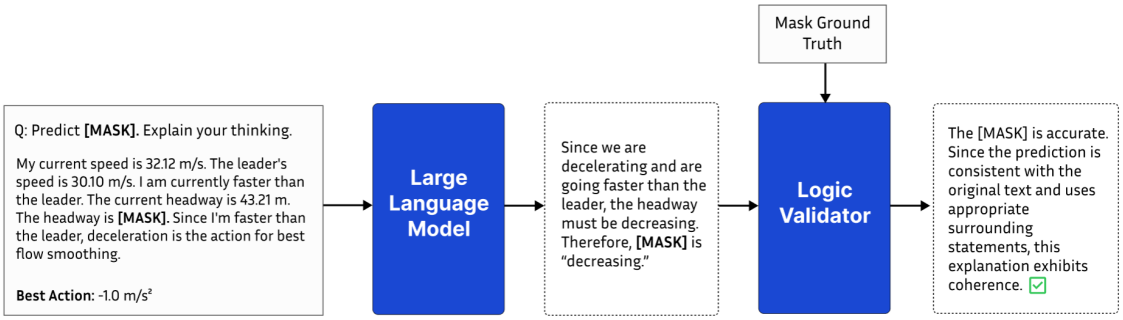


Figure 4: **Logic Validator Evaluating Coherence.** Example of an explanation demonstrating strong logical coherence due to accurate reconstruction. The Logic Validator makes no modifications to the input explanations in this specific case.

Logic Validator

In addition to verifying factual accuracy, the Logic Validator ensures that the model's reasoning process remains logically sound, addressing the challenge of multi-step reasoning where small errors can compound over time [36]. To detect logic flaws, generated statements are categorized as either observational (validated through pattern matching) or inferential (assessed for logical coherence through reconstruction accuracy). The model's reasoning is tested by randomly masking key components, such as verb phrases and relationships, and prompting another LLM to reconstruct each masked segment zero-shot. A total of M masks are selected, and the predictions are aggregated for analysis.

The Logic Validator then reviews the reconstructed reasoning and flags the steps with the most sources of error. When discrepancies are detected, it suggests modifications to the original explanation, adding qualifying statements or additional detail to correct brittle or unsupported inferential statements. This process helps to identify weaknesses in the logic, ensuring that the explanation remains coherent and well-supported by the reasoning process.

Chapter 4

Experiments and Dicussion

4.1 Experiment Setup

Data Collection and Evaluation Tasks

We evaluate CLEAR on two tasks: (Task 1) state-to-action mapping with rationale generation, and (Task 2) hypothetical future state prediction. Our experiments use real-world trajectory data collected during the VanderTest deployment on I-24 (see Figure 6). This data reflects realistic mixed-autonomy traffic scenarios, providing a comprehensive testbed to assess the quality and applicability of our explanations in diverse, real-world driving conditions.

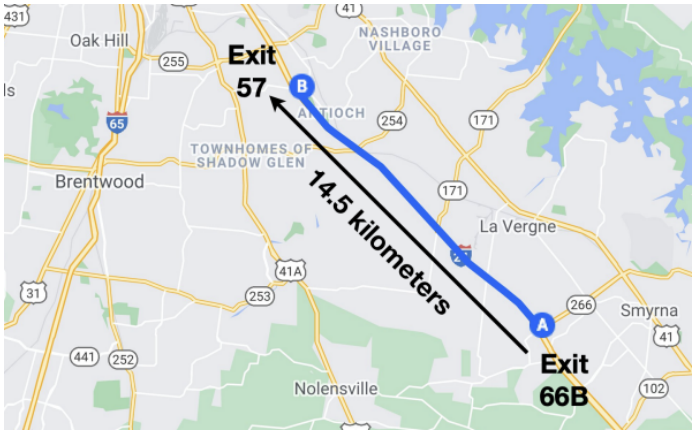


Figure 5: **Segment of I-24 of Data Collection.** This figure is adapted from [16] This is the segment of length 14.5km where the evaluation data is collected from. All data comes from on-policy deployment of the AV and is taken at 10 Hz.

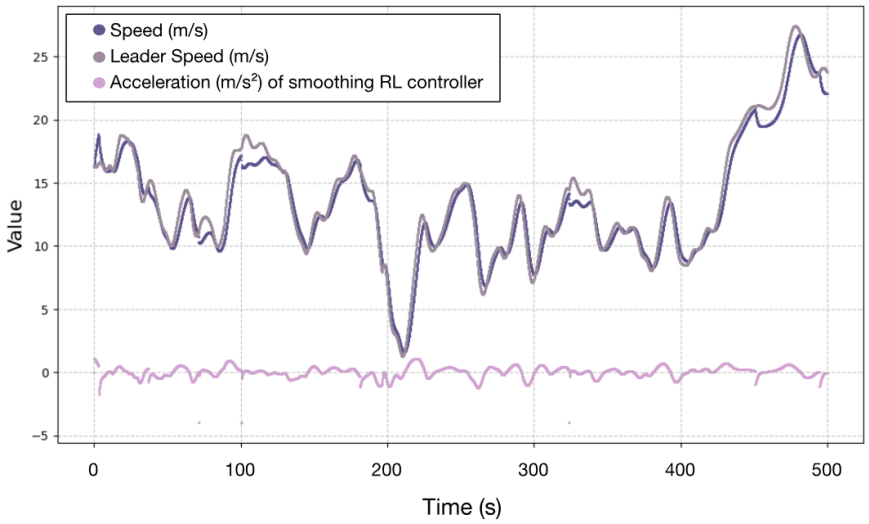


Figure 6: **Example Trajectory for Validation**. This figure represents an example trajectory from an RL-equipped AV. This was collected from the I-24 highway during the VanderTest.

CLEAR is designed to be model-agnostic, allowing for easy integration with various language model providers. For evaluation, we instantiate CLEAR with Gemini Flash 2.0, set to a temperature of 1.0. We selected Gemini Flash 2.0 for its strong reasoning capabilities and long context window, both of which are beneficial for processing sequential trajectory data. We compare CLEAR against four baselines: (a) zero-shot learning, (b) few-shot learning, (c) CLEAR without explicit validators, and (d) supervised fine-tuning (SFT) using LLaMA 3.2 8B as the open-source baseline [1]. The LLaMA model was trained on 770 points for 3 epochs to imitate CLEAR responses with ground-truth action predictions.

For Task 1, we simulate an online-learning scenario where CLEAR is presented with I-24 observations in a randomized order, starting with an empty context cache. This setup tests the framework's ability to learn on-the-fly and adapt to new information while avoiding temporal bias. We allow retrieval of up to two task-specific examples from the context cache by storing a numerical representation of each observation as its embedding and using Euclidean distance as the similarity metric. Performance is measured using mean absolute error (MAE) for action prediction accuracy. We also evaluate the generated explanations by comparing them to a set of manually annotated ground-truth rationales.

For Task 2, we augment the field trajectory data with synthetically generated future scenarios over a 50-timestep horizon to represent diverse and challenging situations, such as sudden braking or vehicle cut-ins. To ensure a meaningful evaluation, we warm-start the context database with around 50 scenarios based on common archetypes, drawn from the 800 timesteps of online learning. This warm start is crucial as it provides a foundation of

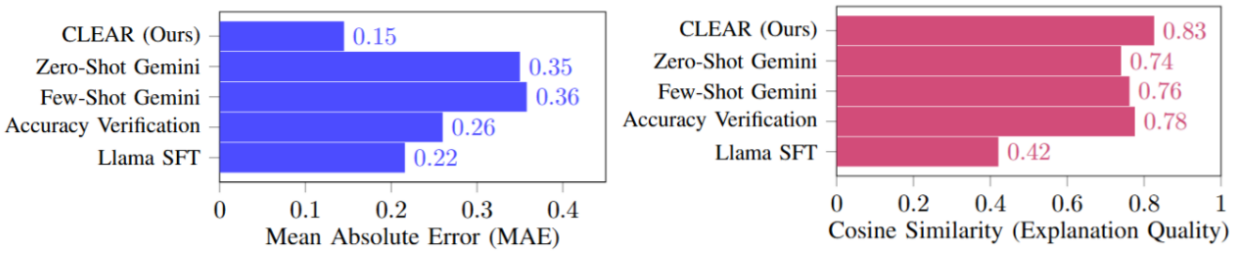


Figure 7: **Left:** Mean Absolute Error (MAE) of each method compared to the ground truth (lower is better). **Right:** Cosine similarity between generated explanations and the ground truth, reflecting explanation quality (higher is better).

relevant past scenarios, allowing the model to retrieve the most pertinent examples based on semantic similarity, thereby enhancing prediction accuracy and interpretability. The evaluation focuses on the last 30 points, where hypothetical situations are constructed for each archetype. Model performance is assessed using the average L1 norm on predicted vehicle states (ego speed, leader speed, headway).

4.2 Results

Action Prediction Accuracy

We evaluate performance by predicting the actions the RL controller would have taken for 800 randomly selected observations drawn from 15,000 trajectories in the validation set (corresponds to 25 minutes of trajectory data). Sampling is randomized to ensure diverse evaluation and prevent bias when accumulating memory in the context cache. For instance, scenarios where the leader vehicle remains stationary for extended periods can skew evaluation.

As shown in Figure 7, CLEAR achieves a significantly lower MAE of 0.15 in predicting the RL controller's actions, outperforming all baselines. A baseline is using LLaMA with trained with SFT achieved a lower MAE of 0.22, though this is largely due to degenerate predictions of almost always predicting slightly negative, near-zero actions, which happens to perform well on average conditions. A CLEAR ablation variant using a multi-agent correction loop without explicit validators achieves 0.26 MAE but still underperformed compared to the full framework, reinforcing the importance of explicit validators as precise alignment mechanisms for faithfully capturing decision logic.

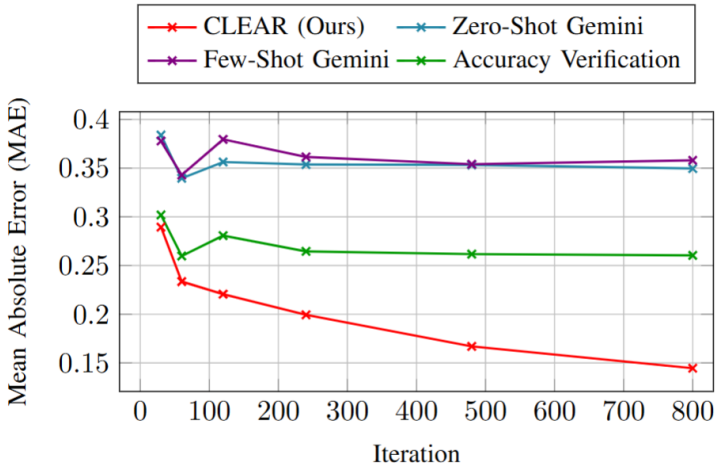


Figure 8: **Self-learning capabilities of CLEAR.** Evolution of the MAE across iterations for various methods, illustrating how learning performance improves with accumulated experience in an online learning setting. Results are shown for Gemini 2.0 Flash and a CLEAR ablation with no explicit accuracy verifier.

Similarity to Curated Ground-Truth Explanations

A high-quality explanation should identify the relevant features in the observation and convey accurate cause-and-effect reasoning behind the decision. To assess this, we curate a set of ground-truth rationales that reflect the key factors a human would cite when justifying the RL controller's behavior. Conventional text similarity metrics such as BLEU [20] and ROUGE [17] are inadequate for evaluating explanation quality, as they focus on lexical overlap rather than semantic content. To better assess alignment with human reasoning, we compute cosine similarity between MiniLM-L6-v2 sentence embeddings [30] of the generated explanations and curated ground-truth rationales.

As shown in Figure 7, CLEAR achieves the highest similarity score (0.83), though its relative performance gap is smaller compared to other metrics. Qualitatively, we observe that most methods correctly identify and reference key features in the observation. However, they often fail to make accurate inferential claims, especially in cases that require multi-step reasoning. This limitation is evident in the significantly higher MAE exhibited by other methods when predicting the final optimal action. Meanwhile, although LLaMa achieves a relatively higher MAE compared to other methods (except CLEAR), its explanations tend to be degenerate. This emphasizes that a primary challenge lies in scaling supervised fine-tuning. Such an approach requires large amounts of annotated data, which doesn't align with the constraints of our problem setting, where annotated data is limited.

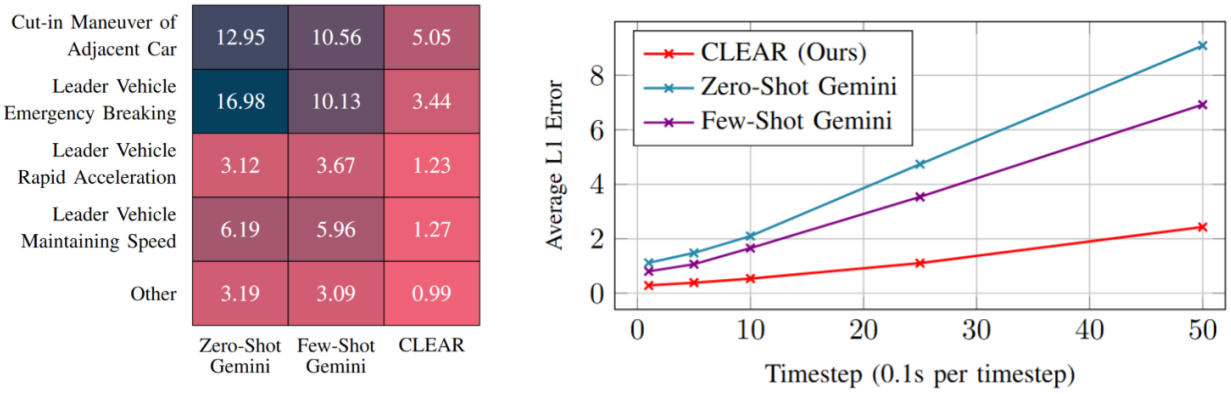


Figure 9: **Left:** Average L1 error of predicted states across different archetypes of hypothetical scenarios, grouped by method. Predictions are made 5 seconds into the future. **Right:** Environment prediction accuracy over varying forecast horizons. This plot illustrates how L1 error evolves with longer prediction horizons, revealing error propagation trends across methods.

Framework Performance with Memory Accumulation

As shown in Figure 8, CLEAR demonstrates significant self-learning capabilities, with its MAE dropping from 0.29 after 30 iterations to 0.22 by iteration 120, showcasing substantial performance gains early on. After 800 iterations, it achieves an MAE of 0.15, highlighting the effectiveness of the Correctional Layer in continuously integrating new data. In contrast, Few-shot Gemini and Zero-Shot Gemini have no improvement capabilities, with MAE values remaining largely static around 0.35. The variant of CLEAR with a context cache but no explicit validators shows some initial improvement but quickly plateaus, underscoring the necessity of explicit feedback for sustained learning and convergence.

State Transition Prediction Accuracy

To assess CLEAR's ability to reason under diverse driving conditions, we evaluate its prediction accuracy across several synthetically perturbed scenario classes using a forward dynamics model calibrated with parameters from the VanderTest. Each scenario is simulated for 50 timesteps (5 seconds) into the future. As shown in Figure 9, CLEAR significantly outperforms baselines across all scenarios, often halving or even reducing error by over 70%. For instance, in the challenging "Leader Vehicle Emergency Braking" scenario, CLEAR achieves an error of 3.44 compared to 16.98 for Zero-Shot Gemini. Similarly, in "Cut-in Maneuver of Adjacent Car," CLEAR reduces error from 12.95 to 5.05. These results demonstrate CLEAR's robustness in modeling complex interactions and dynamic transitions that traditional prompting methods fail to capture reliably.

Prediction Performance Across Varying Horizons

To assess how error evolves with increasing prediction horizons, we evaluate performance across varying future window sizes. This tests the model's ability to maintain stability over time and manage compounding errors. As shown in Figure 9, CLEAR consistently outperforms all baselines, starting at an average L1 error of 0.29 and rising gradually to 2.43 at 50 timesteps. Meanwhile, Zero-Shot Gemini's error sharply increases from 1.11 to 9.09, and Few-Shot Gemini's from 0.80 to 6.92. CLEAR's slower error propagation results in a noticeably flatter curve, demonstrating its ability to maintain consistent performance in long-horizon state forecasting. This ability to provide stable future analysis across both short and long-term windows demonstrates CLEAR's effectiveness in a diverse set of hypothetical driving scenarios, accounting for both situational complexity and time scale.

Chapter 5

Conclusion

This paper introduced CLEAR, a general framework for generating verifiable natural language explanations of deep reinforcement learning (RL) policies. We focused on the mixed-autonomy traffic smoothing setting, where learned policies often exhibit emergent behaviors that are counterintuitive and differ significantly from typical human behavior. These challenges were evident in the MegaVanderTest deployment, where operator discomfort and lack of interpretability led to frequent disengagement and limited system adoption. CLEAR addresses this gap by producing faithful, human-readable rationales for RL controller actions, using a suite of validators that enforce both factual accuracy and logical coherence.

Evaluated on real-world highway data from the VanderTest benchmark, CLEAR consistently outperformed baseline methods across multiple dimensions. It accurately predicted controller behavior, generated semantically meaningful explanations that aligned with human reasoning, and reliably forecasted future traffic conditions across a variety of hypothetical scenarios. These results demonstrate the importance of integrated, multi-layer validation for producing trustworthy and policy-consistent explanations, particularly in high-stakes environments where naive use of language models can lead to hallucinations and unreliable outputs. In future work, we aim to extend CLEAR to a wider range of safety-critical domains such as healthcare, robotics, and finance, where interpretability and verifiability are essential for aligning AI systems with human trust and oversight.

Bibliography

[1] AI@Meta. "Llama 3 Model Card". In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[2] Akari Asai et al. *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. 2023. arXiv: 2310.11511 [cs.CL]. URL: https://arxiv.org/abs/2310.11511.

[3] Leonard Bereska and Efstratios Gavves. *Mechanistic Interpretability for AI Safety – A Review*. 2024. arXiv: 2404.14082 [cs.AI]. URL: https://arxiv.org/abs/2404.14082.

[4] Marouane Bouadi et al. *Stochastic factors and string stability of traffic flow: Analytical investigation and numerical study based on car-following models*. 2022. arXiv: 2203.04877 [physics.soc-ph]. URL: https://arxiv.org/abs/2203.04877.

[5] Can Cui et al. *A Survey on Multimodal Large Language Models for Autonomous Driving*. 2023. arXiv: 2311.12320 [cs.AI]. URL: https://arxiv.org/abs/2311.12320.

[6] Shehzaad Dhuliawala et al. *Chain-of-Verification Reduces Hallucination in Large Language Models*. 2023. arXiv: 2309.11495 [cs.CL]. URL: https://arxiv.org/abs/2309.11495.

[7] Shuo Feng et al. "String stability for vehicular platoon control: Definitions and analysis methods". In: *Annual Reviews in Control* 47 (2019), pp. 81–97. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2019.03.001. URL: https://www.sciencedirect.com/science/article/pii/S1367578819300240.

[8] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

[9] Claire Glanois et al. *A Survey on Interpretable Reinforcement Learning*. 2022. arXiv: 2112.13112 [cs.LG]. URL: https://arxiv.org/abs/2112.13112.

[10] Jyh-Jing Hwang et al. *EMMA: End-to-End Multimodal Model for Autonomous Driving*. 2024. arXiv: 2410.23262 [cs.CV]. URL: https://arxiv.org/abs/2410.23262.

[11] Kathy Jang et al. *Reinforcement Learning Based Oscillation Dampening: Scaling up Single-Agent RL algorithms to a 100 AV highway field operational test*. arXiv:2402.17050 [cs, eess]. May 2024. DOI: 10.48550/arXiv.2402.17050. URL: http://arxiv.org/abs/2402.17050 (visited on 09/01/2024).

[12] Kemou Jiang et al. *KoMA: Knowledge-driven Multi-agent Framework for Autonomous Driving with Large Language Models*. 2024. arXiv: 2407.14239 [cs.AI]. URL: https://arxiv.org/abs/2407.14239.

[13] Philip Koopman and Michael Wagner. "Challenges in autonomous vehicle testing and validation". In: *SAE International Journal of Transportation Safety* 4.1 (2016), pp. 15–24. DOI: 10.4271/2016-01-0132.

[14] Abdul Rahman Kreidieh, Cathy Wu, and Alexandre M. Bayen. "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning". In: *2018 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 1475–1480. DOI: 10.1109/ITSC.2018.8569485. URL: https://flow-project.github.io/papers/08569485.pdf.

[15] Jonathan W Lee et al. "Traffic Control via Connected and Automated Vehicles (CAVs): An Open-Road Field Experiment with 100 CAVs". In: *IEEE Control Systems* 45.1 (2025), pp. 28–60.

[16] Nathan Lichtlé et al. "Deploying traffic smoothing cruise controllers learned from trajectory data". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2884–2890.

[17] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[18] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874.

[19] Ana-Maria Marcu et al. *LingoQA: Visual Question Answering for Autonomous Driving*. 2024. arXiv: 2312.14115 [cs.RO]. URL: https://arxiv.org/abs/2312.14115.

[20] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[21] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[22] Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. "Reinforcement learning algorithms: A brief survey". In: *Expert Systems with Applications* 231 (2023), p. 120495. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120495. URL: https://www.sciencedirect.com/science/article/pii/S0957417423009971.

[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2014. arXiv: 1312.6034 [cs.CV]. URL: https://arxiv.org/abs/1312.6034.

[24] Chandan Singh et al. *Rethinking Interpretability in the Era of Large Language Models*. 2024. arXiv: 2402.01761 [cs.CL]. URL: https://arxiv.org/abs/2402.01761.

[25] Chandan Singh et al. *Rethinking Interpretability in the Era of Large Language Models*. 2024. arXiv: 2402.01761 [cs.CL]. URL: https://arxiv.org/abs/2402.01761.

[26] Shounak Sural, Naren, and Ragunathan Raj Rajkumar. "ContextVLM: Zero-Shot and Few-Shot Context Understanding for Autonomous Driving Using Vision Language Models". In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. 2024, pp. 468–475. DOI: 10.1109/ITSC58415.2024.10920066.

[27] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2017, pp. 23–30. DOI: 10.1109/IROS.2017.8202135.

[28] Eugene Vinitsky et al. "Benchmarks for reinforcement learning in mixed-autonomy traffic". In: *Conference on Robot Learning*. PMLR. 2018, pp. 399–409. URL: http://proceedings.mlr.press/v87/vinitsky18a.html.

[29] Eugene Vinitsky et al. "Optimizing Mixed Autonomy Traffic Flow with Decentralized Autonomous Vehicles and Multi-Agent Reinforcement Learning". In: *ACM Trans. Cyber-Phys. Syst.* 7.2 (Apr. 2023). ISSN: 2378-962X. DOI: 10.1145/3582576. URL: https://doi.org/10.1145/3582576.

[30] Wenhui Wang et al. *MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers*. 2021. arXiv: 2012.15828 [cs.CL]. URL: https://arxiv.org/abs/2012.15828.

[31] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24824–24837.

[32] Licheng Wen et al. *DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models*. 2024. arXiv: 2309.16292 [cs.RO]. URL: https://arxiv.org/abs/2309.16292.

[33] Cathy Wu et al. "Emergent Behaviors in Mixed-Autonomy Traffic". In: *Proceedings of the 1st Annual Conference on Robot Learning*. Vol. 78. PMLR. 2017, pp. 398–407. URL: http://proceedings.mlr.press/v78/wu17a/wu17a.pdf.

[34] Cathy Wu et al. "Flow: A Modular Learning Framework for Mixed Autonomy Traffic". In: *IEEE Transactions on Robotics* 38.2 (Apr. 2022), pp. 1270–1286. ISSN: 1941-0468. DOI: 10.1109/tro.2021.3087314. URL: http://dx.doi.org/10.1109/TRO.2021.3087314.

[35] Zhenhua Xu et al. *DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model*. 2024. arXiv: 2310.01412 [cs.CV]. URL: https://arxiv.org/abs/2310.01412.

[36] Tianci Xue et al. *RCOT: Detecting and Rectifying Factual Inconsistency in Reasoning by Reversing Chain-of-Thought*. 2023. arXiv: 2305.11499 [cs.CL]. URL: https://arxiv.org/abs/2305.11499.

[37] Shi-Qi Yan et al. *Corrective Retrieval Augmented Generation*. 2024. arXiv: 2401.15884 [cs.CL]. URL: https://arxiv.org/abs/2401.15884.

[38] Catherine Yeh et al. *AttentionViz: A Global View of Transformer Attention*. 2023. arXiv: 2305.03210 [cs.HC]. URL: https://arxiv.org/abs/2305.03210.

[39] Jing Zhao et al. "Neural sim-to-real transfer for autonomous driving: A survey". In: *IEEE Transactions on Intelligent Transportation Systems* 24.2 (2022), pp. 1–18. DOI: 10.1109/TITS.2022.3228838.

[40] Xingcheng Zhou et al. *Vision Language Models in Autonomous Driving: A Survey and Outlook*. 2024. arXiv: 2310.14414 [cs.CV]. URL: https://arxiv.org/abs/2310.14414.