IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Chatgpt: An Ultimate Driving Companion: Systematic Review

[1]Ananya Sadanand Gowda,[2]Mamatha G

[1]Student, Dept. of Information Science and Engineering, JSS Academy of Technical Education, Bengaluru, India

[2]Assistant Professor, Dept. of Information Science and Engineering, JSS Academy of Technical Education, Bengaluru, India

Abstract: Human-machine collaboration presents persistent challenges in aligning human intent with machine comprehension and execution. Large Language Models (LLMs) offer promising solutions by leveraging advanced natural language processing capabilities to bridge this gap. This paper surveys a novel framework that integrates LLMs into a vehicle "Co-Pilot," enabling autonomous systems to interpret and execute driving tasks based on human commands and contextual information. The proposed framework incorporates a robust interaction workflow and a memory mechanism to systematically organize and retrieve task-relevant data. By dynamically selecting appropriate controllers and planning optimal trajectories, the Co-Pilot adapts its operations to fulfill user-defined goals while maintaining safety and efficiency. Simulation experiments demonstrate the framework's ability to understand natural language instructions, plan actions, and execute driving tasks effectively, highlighting both its practical viability and limitations. Furthermore, the study emphasizes the importance of real-time adaptability in addressing complex driving scenarios and explores the concept of human-machine hybrid intelligence. This work illustrates the potential of LLMs to revolutionize autonomous driving by enabling more intuitive and effective human-machine collaboration.

***Index Terms* -** Human-machine collaboration, Large Language Models (LLMs), Vehicle Co-Pilot, Human-machine interaction, Natural language understanding, Trajectory planning, Autonomous driving, Hybrid intelligence.

I. Introduction

Artificial Intelligence (AI) has significantly advanced human-machine collaboration, opening new avenues for enhancing how people interact with technology. Among these innovations, Large Language Models (LLMs) have emerged as transformative tools capable of processing, understanding, and interpreting natural language with remarkable accuracy. In the context of autonomous driving, LLMs present a unique opportunity to bridge the gap between complex machine functions and intuitive human intent. This paper explores a framework titled "ChatGPT as Our Ultimate Driving Companion," which leverages LLMs as a Co-Pilot to facilitate seamless communication between users and autonomous vehicles. By enabling natural language interactions, this system eliminates the need for rigid, pre-programmed inputs and introduces a more flexible and adaptive driving experience.

Conventional autonomous systems often rely on structured commands and predefined algorithms, which can be inadequate in dynamic or unpredictable environments. The Co-Pilot framework addresses these limitations through an interactive workflow that interprets natural language instructions and converts them into executable driving tasks. This intuitive interface democratizes vehicle operation, making it accessible to users regardless of their technical expertise. In addition to basic command execution, the Co-Pilot demonstrates advanced capabilities such as trajectory planning, control selection, and path tracking, aligning vehicle behavior with user-defined goals. These adaptive functions enhance the system's ability to respond in real-time to evolving driving scenarios.

By integrating LLMs into the vehicle control loop, this framework represents a significant step toward the convergence of human intuition and machine intelligence. Through simulation trials, the study demonstrates the feasibility of using natural language as the sole input for complex task execution, underscoring the transformative potential of LLMs in redefining autonomous driving. Ultimately, this paper highlights the synergy between AI-driven adaptability and human-centric design, paving the way for more intelligent, responsive, and user-friendly autonomous vehicle systems.

II. LITERATURE REVIEW

The intersection of natural language processing and autonomous vehicle control has garnered increasing attention, particularly with the emergence of Large Language Models (LLMs) as key enablers of intuitive human-machine collaboration. This section reviews relevant literature in areas directly contributing to the development of the Co-Pilot framework, emphasizing both foundational and recent advancements.

2.1 Advancements in Large Language Models (LLMs): Large Language Models have exhibited significant progress in understanding and generating human-like language, making them instrumental in bridging the gap between human intent and machine execution. Foundational works such as [1] and [2] illustrate how LLMs like GPT and BERT can be fine-tuned for task-specific dialogue, decision support, and instruction following. More recent research [3] has examined LLM integration in interactive robotics and embodied AI, demonstrating their potential in real-time decision-making for autonomous systems.

2.2 Human Intent Interpretation in Autonomous Driving: Understanding and interpreting human instructions remains a fundamental challenge in autonomous vehicle development. Studies like [4] and [5] focus on natural language interfaces for autonomous driving agents, showing methods to parse user commands into structured driving intents. These approaches enable dynamic task execution and align closely with the interaction layer in the Co-Pilot framework. Furthermore, [6] introduces contextual reasoning mechanisms that allow vehicles to infer intent from partial or ambiguous input—an area of direct relevance to this paper.

2.3 Trajectory Planning and Path Tracking: Trajectory planning and control are central to autonomous navigation. Algorithms such as Model Predictive Control (MPC), Rapidly-exploring Random Trees (RRT), and deep reinforcement learning-based planners have been discussed extensively in [7] and [8]. These studies present methods to optimize trajectories under constraints like safety, efficiency, and obstacle avoidance. Their integration into higher-level planning modules, such as those influenced by language input, is an emerging topic explored in [9].

2.4 Memory Mechanisms in Human-Machine Collaboration: Efficient memory architectures are essential for context-aware task execution, especially in environments requiring sequential decision-making. Work in [6] and [4] explores memory-augmented neural networks, including attention-based systems, to retain temporal data and user interaction histories.

2.5 Applications of Hybrid Intelligence in Autonomous Systems: Hybrid intelligence—combining human cognitive strengths with machine processing capabilities—has shown promise in domains requiring adaptability and user-centered design. Research in [1] and [3] investigates collaborative frameworks where human feedback shapes AI behavior in real time, particularly in safety-critical applications like autonomous driving.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The Co-Pilot Framework aims to bridge the gap between human intent and vehicle execution by leveraging Large Language Models (LLMs) for natural language understanding and control task generation. At its core, the framework facilitates seamless interaction between human users—either passengers or drivers—and autonomous vehicles. When a user issues natural language commands such as “Overtake that car!” or “Avoid the pillar ahead,” the LLM processes these inputs, which are first tokenized and enriched with memory context. The encoded instructions are then decoded into actionable tasks like trajectory planning and motion control.

A central component of the system is its memory mechanism, which enhances adaptability and context-awareness. The Working Memory captures real-time vehicle states such as speed, trajectory, and nearby obstacles, while the Long-Term Memory stores past interactions to support decision-making in evolving scenarios. This layered memory design enables the Co-Pilot to continuously evaluate the current environment, recall relevant historical data, and refine its behavior in real-time. Figure 1 illustrates the overall architecture, including human input, memory modules, LLM-based processing, and vehicle's execution layer.

To ensure optimal performance and safety, the system incorporates expert-guided black-box tuning. Instead of retraining the LLM directly, experts provide high-level feedback that guides adjustments to system behavior, improving adaptability and robustness over time. In summary, the Co-Pilot Framework integrates natural language processing, memory-enhanced decision-making, and expert feedback to create a dynamic, real-time human-machine collaboration platform for autonomous driving.

IV. BACKGROUND

4.1 Controlling the Path Tracking: Autonomous tasks for driving are structured into layers, with tracking the path in the control layer ensuring the automobile/ vehicle follows a predefined path while balancing speed and safety. Therefore involving the functions such as balancing steering, throttle, and brake commands based on the vehicle's state and input from the planning layer.

A closed-loop control system addresses this challenge through two modules: the determination module of reference state, which calculates the path of target, and the module controlled, which uses a feedback controller to align the vehicle's actual state with the target path.

The following four feedback controllers and systems are customarily used for this purpose:

4.1.1 Nonlinear Predictive Control Model i.e (NPCM)}: Nonlinear Predictive Control Model i.e (NPCM) uses a nonlinear system to generate optimal control commands. The system's state **{x} = f(x, u)** is governed by a set of dynamics, and NMPC seeks to minimize a performance index J over a good horizon. The cost function is defined as per Figure 2.

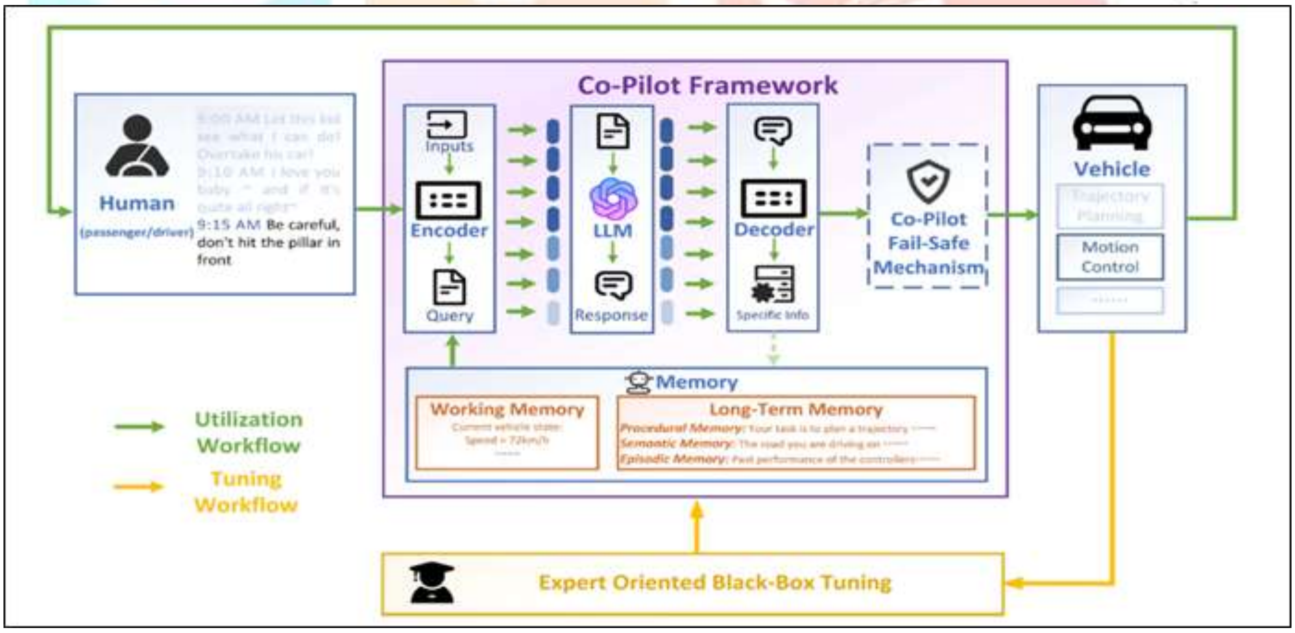
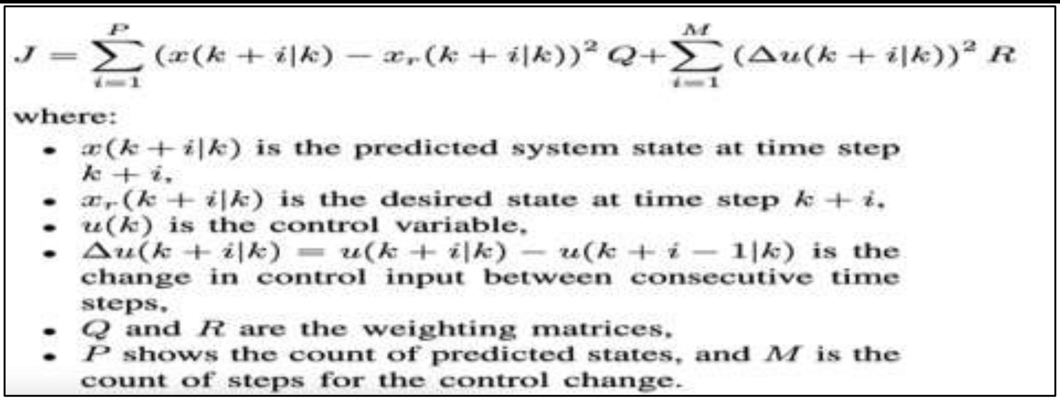


Figure 1. Architecture of Co-Pilot framework with key components - interaction layer, memory mechanism, execution modules [1]

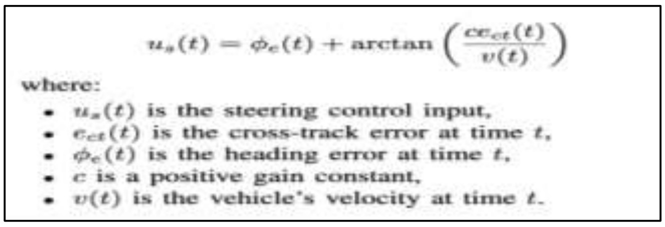


$$J=\sum_{i=1}^{P}(x(k+i|k)-x_r(k+i|k))^2Q+\sum_{i=1}^{M}(\Delta u(k+i|k))^2R$$

where:

- $x(k+i|k)$ is the predicted system state at time step $k+i$,
- $x_r(k+i|k)$ is the desired state at time step $k+i$,
- $u(k)$ is the control variable,
- $\Delta u(k+i|k)=u(k+i|k)-u(k+i-1|k)$ is the change in control input between consecutive time steps,
- Q and R are the weighting matrices,
- P shows the count of predicted states, and M is the count of steps for the control change.

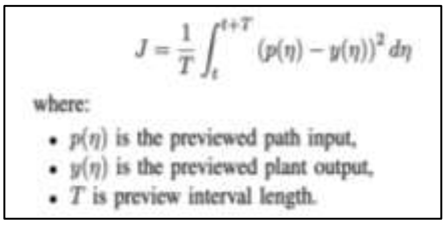
Figure 2: Cost function (NPCM)



$$u_s(t)=\phi_e(t)+\arctan\left(\frac{ce_{ct}(t)}{v(t)}\right)$$

where:

- $u_s(t)$ is the steering control input,
- $e_{ct}(t)$ is the cross-track error at time t,
- $\phi_e(t)$ is the heading error at time t,
- c is a positive gain constant,
- $v(t)$ is the vehicle's velocity at time t.



$$J=\frac{1}{T}\int_t^{t+T}(p(\eta)-y(\eta))^2\,d\eta$$

where:

- $p(\eta)$ is the previewed path input,
- $y(\eta)$ is the previewed plant output,
- T is preview interval length.

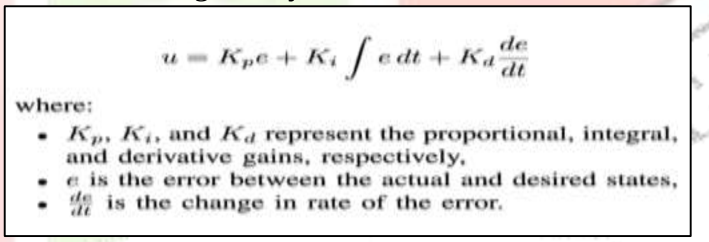
Figure 3: Control input of steering

Figure 4: Cost function (Optimal)

4.1.2 Stanley Controller: The Stanley Controller \cite{ref3}, \cite{ref4} uses a mathematical approach to compute the direction of steering variable u_s based on cross-track error e_{ct} and the heading error ϕ_e between the vehicle/automobile and the target point. The control input of steering is given by (refer Figure 3)

4.1.3 Optimal Controller of Preview at Single-Point: Optimal Controller of Preview at Single-Point [5] uses the model of single-point to calculate the controlling of speed variable u_v by reducing a local performance index J over the current interval (t, t + T). The function of cost or either called as cost function is given by (refer Figure 4)

4.1.4 Derivative at Proportions Controller: The Derivative at Proportions Controller \cite{ref6} is a classic feedback controller that derives the controlling variable ubased on the error e between the desired state and the actual state. The control law is given by:



$$u=K_pe+K_i\int e\,dt+K_d\frac{de}{dt}$$

where:

- K_p, K_i, and K_d represent the proportional, integral, and derivative gains, respectively,
- e is the error between the actual and desired states,
- $\frac{de}{dt}$ is the change in rate of the error.

Figure 5: Control law representation

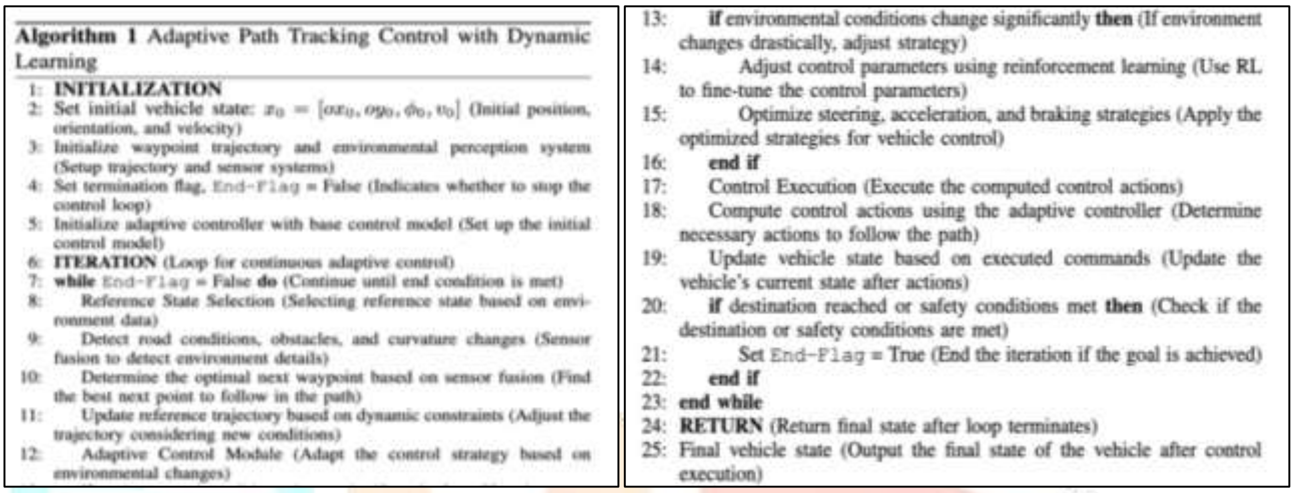
Path tracking for vehicles begins with selecting an optimal trajectory, which can be a predetermined route or a dynamically generated one based on sensor data. This process utilizes tools like LiDAR, cameras, and GPS to ensure the vehicle remains in its lane and adjusts to changing road conditions, minimizing sharp turns and promoting smooth navigation. To maintain precise tracking, feedback control mechanisms such as PID controllers or Model Predictive Control are employed, allowing real-time adjustments to steering, acceleration, and braking if the vehicle deviates from its path due to external disturbances.

4.2 Planning the Trajectory: Planning of Trajectory is most important in autonomous or no-human driving, ensuring safe and efficient paths by adjusting for factors like road geometry and traffic, and also limitations of vehicle. It involves mini problems like planning of path, planning of speed, and maneuvers such as the Double Lane Change (DLC) for obstacle avoidance and stability testing.The fusion of Large Language Models (LLMs) offers potential for improved human-machine interaction, allowing passengers to provide natural language commands. The Framework of Co-Pilot combines traditional control methods with machine learning to enhance vehicle responsiveness to human intent while ensuring safety and efficiency [3].

V. THE CO-PILOT FOR CONTROLLING TRACKED PATH:

The Co-Pilot system, as described in Section III, was initially designed to handle path-tracking control tasks. This system joins the results of multiple old controllers and delivers real-time control decisions based on human input for upcoming road conditions. This section provides a comprehensive overview of the task and details the implementation of the Co-Pilot system [4].

5.1 Task Background and Platform Setup: The Co-Pilot system is implemented using a joint simulation platform that combines Simulink for system operations and CarSim for environment and vehicle simulation. This setup allows for a realistic testing environment, featuring a 2273.8471-meter racetrack with 10 corners and 2261 waypoints. [3].



Algorithm 1 Adaptive Path Tracking Control with Dynamic Learning

```
1:  INITIALIZATION
2:  Set initial vehicle state: $x_0 = [ox_0, oy_0, \phi_0, v_0]$ (Initial position, orientation, and velocity)
3:  Initialize waypoint trajectory and environmental perception system (Setup trajectory and sensor systems)
4:  Set termination flag, End-Flag = False (Indicates whether to stop the control loop)
5:  Initialize adaptive controller with base control model (Set up the initial control model)
6:  ITERATION (Loop for continuous adaptive control)
7:  while End-Flag = False do (Continue until end condition is met)
8:      Reference State Selection (Selecting reference state based on environment data)
9:      Detect road conditions, obstacles, and curvature changes (Sensor fusion to detect environment details)
10:     Determine the optimal next waypoint based on sensor fusion (Find the best next point to follow in the path)
11:     Update reference trajectory based on dynamic constraints (Adjust the trajectory considering new conditions)
12:     Adaptive Control Module (Adapt the control strategy based on environmental changes)
13:     if environmental conditions change significantly then (If environment changes drastically, adjust strategy)
14:         Adjust control parameters using reinforcement learning (Use RL to fine-tune the control parameters)
15:         Optimize steering, acceleration, and braking strategies (Apply the optimized strategies for vehicle control)
16:     end if
17:     Control Execution (Execute the computed control actions)
18:     Compute control actions using the adaptive controller (Determine necessary actions to follow the path)
19:     Update vehicle state based on executed commands (Update the vehicle's current state after actions)
20:     if destination reached or safety conditions met then (Check if the destination or safety conditions are met)
21:         Set End-Flag = True (End the iteration if the goal is achieved)
22:     end if
23: end while
24: RETURN (Return final state after loop terminates)
25: Final vehicle state (Output the final state of the vehicle after control execution)
```

Figure 6. Pseudo code of Adaptive path tracking control with Dynamic learning Algorithm

The Dynamic Learning-based adaptive path tracking control algorithm (figure 6.) is an innovative solution for autonomous vehicle navigation, enabling smooth and secure trajectory tracking by adapting to real-time environmental conditions. Unlike traditional path-tracking methods that depend on static control models, this algorithm utilizes reinforcement learning and sensor data from GPS, LiDAR, and cameras to dynamically modify its control parameters. Initially, the vehicle's state, including position and velocity, is established alongside a waypoint trajectory.

The system continuously evaluates environmental factors—like road curvature and obstacles—to update the optimal reference trajectory accordingly. This ongoing process enhances the vehicle's ability to navigate complex scenarios, ensuring safety through emergency responses when necessary. Ultimately, the integration of dynamic learning within the algorithm significantly improves trajectory accuracy and responsiveness, making it well-suited for self-driving technologies.

5.2 Co-Pilot System Design: The Co-Pilot system is designed to dynamically select the most appropriate controller based on human input and road conditions [9]. It goes beyond traditional control methods by integrating a Co-Pilot module that adjusts the controller in real-time, ensuring the vehicle's path-tracking performance aligns with the driver’s intentions.

5.3 Co-Pilot Usage Workflow: The Co-Pilot system processes human input and road conditions through a multi-step workflow. The Encoder module standardizes input data, combining human requests with memory data, which is then sent to the LLM via a socket API. [1].

5.4 Co-Pilot Experiment and Final Evaluation: The final experiment as shown in Fig. 6 evaluates the ability of the Co-Pilot system to accept to evolving human intentions and dynamic road conditions. By simulating real-world driving scenarios in which driver preferences shift, the system demonstrates its ability to select the optimal controller. [3].

The Co-Pilot module plays a critical role by processing human intentions and memories through the use of large language models (LLMs). This allows it to select the most perfect controller based on the current state of the vehicle and the preferences of the driver, providing a Controller ID for the next task. The Feedback Controller module then takes over by using the reference provided by the Co-Pilot module to generate the required control commands. These commands direct the vehicle. By combining these three modules, the Co-Pilot system ensures smooth and responsive path-tracking control, adapting to real-time human inputs and dynamic road conditions.

VI. Review of Implementation and Results

We reviewed experiments on two distinct tasks that preliminarily demonstrate the capability of Large Language Models (LLMs) in facilitating human-machine co-driving tasks (figure 7.). These results underscore the promising potential for applying LLMs to autonomous co-driving systems [8].

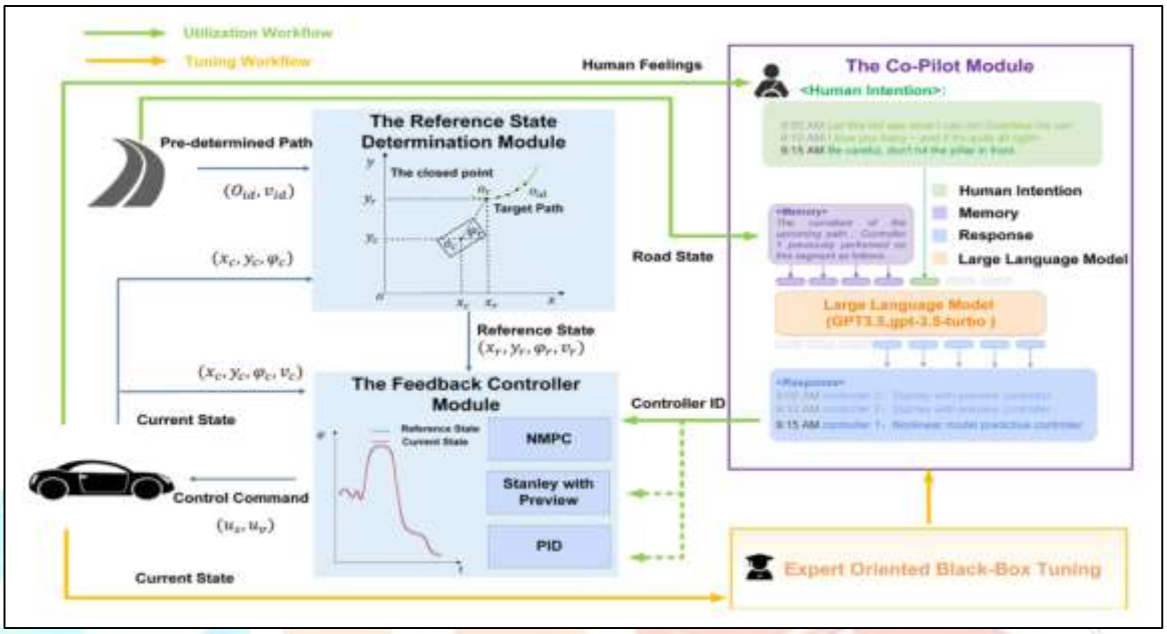


Figure 7. System-level overview of Co-Pilot, with trajectory tracking modules and processing of human commands [1]

6.1 Human-Machine Hybrid Intelligence: Parallel learning accelerates model training through simultaneous execution. In Co-Pilot framework, human experts and LLMs collaborate for prescriptive learning tasks. Human experts provide domain insights while LLMs manage command interpretation and task execution, enhancing decision-making process and adaptability of the system.

6.2 Applications of LLMs in Human-less Driving*:* The Co-Pilot system shows strong potential in addressing co-driving challenges, particularly long-tail problems in autonomous driving. Leveraging memory-based contextual reasoning and common sense knowledge, LLMs effectively handle ambiguous or unfamiliar driving commands. Furthermore, the future integration of large-scale visual-language models capable of processing both textual and visual input holds substantial potential for more immersive autonomous systems.

6.3 Potential Challenges in Autonomous Driving Applications: Despite their advantages, LLMs face several challenges in real-world driving applications:

- Representing complex, dynamic driving environments remains a significant difficulty.
- Lack of sensory perception capabilities restricts the LLM’s environmental awareness.
- Real-time constraints and computational latency may limit deployment on embedded systems.

6.4 The Co-Pilot for Trajectory Planning: We explored an alternative implementation of the Co-Pilot framework tailored to a trajectory planning task. This evaluation provides insight into the system's ability to respond to different human instructions using natural language and highlights its adaptability [1].

6.5 Task Configuration: The trajectory planning task involved a double-lane change (DLC) for static obstacle avoidance. The ego vehicle, traveling at 72 km/h in the right lane, detected a stalled vehicle 100 meters ahead at T=0T = 0T=0. The Co-Pilot was tasked with generating a trajectory to avoid the obstacle and safely return to the original lane.

6.6 Co-Pilot Framework and Tuning: The framework used for this task was intentionally simplified to evaluate the system's fundamental capabilities. A supportive module allowed human experts to provide guided inputs for accurate trajectory planning. An evaluation system was introduced to rate trajectories based on the following criteria*:*

- **Plausibility**: Ensures adherence to road boundaries, vehicle dynamics, and physical constraints.
- **Task Completeness**: Validates whether the obstacle was successfully avoided and the original lane restored.
- **Correctness**: Evaluates safety, efficiency, and comfort of the executed maneuver.

6.7 Experiments with Co-Pilot: Three task variants were tested to assess the Co-Pilot's responsiveness and adaptability:

- **Task 1 (Baseline)**: The system executed a standard DLC trajectory with no additional constraints.
- **Task 2 (Comfort-focused)**: Required a smoother and more comfortable ride for passengers.
- **Task 3 (Speed-focused)**: Emphasized overtaking the stalled vehicle quickly with minimal time spent in opposing lane.

The results demonstrated the Co-Pilot's ability to generate distinct trajectories tailored to each instruction. For example, in Task 2, the system minimized acceleration and lateral jerk to ensure comfort. In Task 3, the system initiated the maneuver earlier and increased acceleration to reduce overtaking time.

6.8 Quantitative Performance Metrics

Table 1 summarizes the performance metrics recorded across the tasks:

Metric	Task 1	Task 2	Task 3
Task Completion Rate (%)	100	100	100
Comfort Score (1-10)	6.8	9.1	5.2
Overtaking Time (s)	4.3	5.7	3.2
Lateral Acceleration (m/s²)	1.9	1.2	2.6

These metrics (table: 1) indicate that the Co-Pilot can prioritize comfort or speed depending on human intent, with measurable differences in trajectory parameters.

6.9 Observations and Conclusions*:* The experiments show that with proper human-guided tuning, the Co-Pilot can successfully interpret natural language instructions and generate trajectories aligned with user objectives. While this study focused on a static, controlled environment, it provides a strong foundation for expanding the Co-Pilot framework to more complex, real-time driving tasks. Future work will include real-world simulations, multi-modal perception integration, and testing under dynamic traffic conditions.

VII. Conclusion

This study explored the application of large language models (LLMs) in tasks for co-driving of human-machine through The Framework of Co-Pilot. We studied that LLMs can enhance collaboration between human experts and autonomous systems, improving control and decision-making processes. By combining human insights with LLM execution, the Co-Pilot system shows promise in autonomous driving tasks.The Framework of Co-Pilot utilizes parallel learning for effective human-machine hybrid intelligence, with human experts providing guidance and LLMs handling task execution.

This adaptive framework can continuously improve through fine-tuning, making it well-suited for dynamic driving scenarios.While LLMs have the potential to address more complex tasks, challenges remain, particularly in handling intricate environments and managing uncertainty in LLM outputs. Further review is needed to understand structured memory systems and robust fail-safe mechanisms to ensure safety, reliability, and accuracy in real-world autonomous driving applications.

VIII. References

[1] S. Wang, Y. Zhu, Z. Li, Y. Wang, L. Li, and Z. He, "ChatGPT as Your Vehicle Co-Pilot: An Initial Attempt," *IEEE Access*, vol. 8, pp. 16, 2023.
[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[3] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and Decision-Making for Autonomous Vehicles,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187-210, 2018.
[4] J. Levinson, M. Montemerlo, and S. Thrun, “Map-Based Precision Vehicle Localization in Urban Environments,” *Robotics: Science and Systems Conference*, 2007.
[5] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, D. McAuley, and A. Mouzakitis, “A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829-846, 2018.
[6] M. Chen, S. Mao, and Y. Liu, “Big Data: A Survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171-209, 2014.
[7] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, 2016.
[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, ... and D. Amodei, “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901, 2020.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.