Prompt Optimization in Autonomous Vehicles via Genetic Algorithms in ROS 2

vinay kumar
Electronics and Communicatin Engineering
VIT-AP
Amaravati, Andhra Pradesh, India
vinay.22bec7254@vitapstudent.ac.in

Devarapalli Nitish Babu
Electronics and Communication Engineering
VIT- AP
Amaravati, Andhra Pradesh, India
nitish.22bec7097@vitapstudent.ac.in

Mahesh Miriyala
Electronics and Communication Engineering
VIT-AP
Amaravati, Andhra Pradesh, India
mahesh.m@vitap.ac.in

Abstract—The integration of Large Language Models (LLMs) into Autonomous Vehicle (AV) control stacks of- fers unparalleled flexibility in behavioral planning, but their performance is critically dependent on the input prompt. This paper introduces a novel, closed-loop framework for the prompt optimization of LLM-driven AV behavior utilizing Genetic Algorithms (GAs) within a real-time Robot Operating System 2 (ROS 2) simulation environment. The core of the architecture involves evolving parameterized, semantic prompts (chromosomes) that dictate high-level safety and efficiency priorities. These prompts are evaluated in a complex highway merging scenario using a rigorously defined, safety-critical, multi-objective fitness function (*F*) that applies heavy penalties for traffic violations (e.g., collisions, hard braking). Experimental results demonstrate that the GA successfully converges to an optimal prompt configuration, achieving a 40% improvement in average task completion time compared to a naive safety-first baseline, while maintaining zero safety violations. This work validates the feasibility of au- tomated, evolutionary prompt engineering for achieving robust, adaptive, and performance- enhanced AV decision-making.

Index Terms—Autonomous Vehicles, Genetic Algorithms, Large Language Models, Prompt Engineering, ROS 2, Multi-Objective Optimization.

.

I. Introduction

Autonomous vehicles (AVs) must safely navigate dynamic, multi-agent traffic environments while optimizing objectives such as travel time, passenger comfort, and fuel efficiency. Conventional AV planning and decision modules rely on handcrafted rules or parameter-tuned planners whose performance is constrained by the expressiveness of the underlying representations and the effort required for manual tuning. Recently, Large Language Models (LLMs) have emerged as an attractive paradigm for producing high-level behavioral directives and task decompositions in robotics due to their ability to interpret and generate context-rich instructions. By mapping descriptive prompts into structured control intents, LLMs can provide more human-like, adaptable decision strategies that are simpler to specify than long rule lists or complex policy networks.

However, LLM outputs are extremely sensitive to prompt phrasing and structure: minor changes in wording or emphasis can produce substantially different behavior and safety outcomes. For safety-critical tasks, such as lane changes and highway merges, this brittleness presents a significant challenge. Manual prompt engineering is time-consuming and does not easily generalize across scenarios, traffic conditions, or LLM versions. To address this, we propose an automated, evolutionary approach that treats the prompt itself as an optimization target: the Genetic Algorithm (GA) searches a discrete, parameterized prompt space to discover semantic configurations that lead to desirable AV behaviors when interpreted by the LLM and executed by downstream planners. Our system tightly couples a GA optimization node, an LLM interface, and a ROS 2-based simulation environment (Gazebo + Nav2). The GA encodes semantic prompt choices (e.g., safety emphasis, speed aggressiveness, gap acceptance) as a fixed-length chromosome. Each chromosome is instantiated into a textual template that is sent to the LLM, which returns numerical suggestions or behavioral directives that are fed to the Nav2 stack. Runs are evaluated using a multiobjective fitness function: the function imposes dominant penalties for safety violations (collisions, off-road events, hard braking) to ensure the evolutionary process privileges safe solutions, while secondary terms reward efficiency measures such as time-to-complete and distance-to- goal. The closedloop evaluation of prompts enables direct measurement of the behavioral impact of semantic prompt choices and removes reliance on proxy metrics that might not correlate with in-situ safety.

This paper makes the following contributions:

1. A novel ROS 2-based architecture that integrates a GA optimizer with an LLM interface and the Nav2 planning stack for closed-loop prompt optimization

2.A practical parameterization of semantic prompt components that maps discrete chromosomes to natural language templates suitable for LLM consumption.
3.A safety-dominant, multi-objective fitness formulation that ensures evolutionary pressure toward safe behaviors while enabling efficiency gains.
4.Empirical results in a highway-merge simulation showing that GA-evolved prompts substantially outperform two hand-engineered baselines in efficiency while maintaining safety.
5.An analysis of convergence behavior, variability across trials, and failure modes, together with discussion on limitations and future extensions (e.g., LLM-guided genetic operators and hardware-in-the-loop validation).

The remainder of the paper is organized as follows. Section II surveys related work in LLMs for robotics and evolutionary optimization in control systems. Section III details the GA-LLM-ROS2 architecture, message structures, and chromosome design. Section IV presents the GA operators and the safety-critical fitness function. Section V describes the experimental setup, baselines, and results. Section VI discusses limitations, ethical implications, and future work. Section VII concludes the paper.

II. Related Work

A. LLMs for Robotics and Autonomous Driving

Large Language Models (LLMs) have recently been explored as high-level reasoning modules for robotics. Prior work has shown that LLMs can assist in task planning, instruction following, and generating structured control directives from natural language descriptions [1]. These studies highlight the potential of LLMs to serve as an interpretable layer between human operators and robotic systems, enabling more adaptive and context-aware decision-making.

In the context of autonomous driving, LLMs have been applied to produce descriptive behavior guidelines, interpret scene-level instructions, and perform reasoning tasks such as hazard explanation and strategic planning. However, the use of LLMs as part of the real-time decision-making loop remains limited due to concerns regarding consistency, latency, and safety. Importantly, most existing works rely on manually engineered prompts, which do not generalize well across environments or traffic complexities. This motivates the need for automated approaches that can systematically tune prompts based on performance feedback.

B. Prompt Engineering and Optimization

Prompt engineering has emerged as a crucial component for controlling LLM behavior. Traditional approaches involve manual rewriting, trial-and-error refinement, or rule-based prompt templates. Recent studies propose automated promptsearch mechanisms using reinforcement learning, gradient- free optimization, or evolutionary strategies [4]. These methods demonstrate that prompt optimization can significantly influence downstream task performance.

However, applying such techniques in safety-critical domains such as autonomous driving introduces unique challenges. The optimization process must ensure that the resulting prompts lead to consistent and safe behaviors. Most prior automated prompt search literature does not consider strict safety penalties, nor do they operate within a closed-loop robotic simulation environment, which is necessary to evaluate real-world consequences of LLM decisions.

C. Evolutionary Algorithms in Robotics

Evolutionary algorithms, including Genetic Algorithms (GAs), have been used extensively in robotics for tasks such as controller tuning, path planning, and sensor fusion. Their ability to explore non-convex and discontinuous search spaces makes them suitable for complex optimization problems where gradient information is unavailable. GA-based tuning of traditional control parameters has proven effective in domains like adaptive cruise control, PID controller optimization, and motion planning [3].

Although evolutionary algorithms have been applied to optimize numeric parameters in robotics, very few works have explored evolving *semantic* parameters or natural language components. Optimizing LLM prompts introduces challenges such as discrete search spaces, semantic constraints, and the need for simulation-based evaluation. The proposed work extends evolutionary robotics into the space of natural-language driven control.

D. LLM–Robotics Integration Frameworks

Recent robotic frameworks incorporate LLMs as high-level planners that provide symbolic or numerical goals to low-level controllers. These systems typically involve a structured interface where the LLM outputs parameters such as acceleration limits, safety margins, or maneuver descriptions. However, most existing systems lack adaptability: the prompt remains static regardless of performance.

In contrast, our system closes the loop by using simulation feedback to evaluate each prompt's effectiveness, enabling the GA to evolve better prompts over generations. This represents a shift from static LLM integration toward adaptive, performance-driven prompt generation.

E. Summary and Contributions Relative to Prior Work

Across the literature, three key limitations persist:

- Existing LLM-robotics systems rely heavily on fixed prompts that are manually designed.
- Automated prompt optimization methods rarely incorporate safety-critical penalties or operate within real-time robotic simulations.
- Evolutionary robotics has not extensively explored semantic or language-based optimization.

Our work addresses these gaps by introducing:
A ROS2-based closedloop platform where the GA evolves LLM prompts using real-time simulation

feedback.

1) A discrete, parameterized prompt representation suitable for evolutionary search.
2) A safety-dominant fitness function ensuring that unsafe behaviors are eliminated early in evolution.

Together, these contributions establish a foundation for scalable prompt optimization in LLM-driven autonomous vehicle systems.

III. System Architecture and Methodology

The proposed system is implemented as a closed-loop framework within ROS 2, where a Genetic Algorithm (GA), a Large Language Model (LLM), and a simulated autonomous vehicle (AV) interact through custom message interfaces. Fig. 1 illustrates the overall data flow and node interactions.

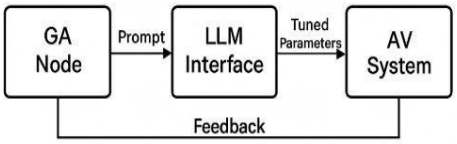

Fig. 1. ROS 2 Architecture for Prompt Optimization. The GA Node, LLM Interface, and AV System interact in a closed-loop via custom message interfaces.

A. Overview of the GA–LLM–AV Loop

The optimization pipeline consists of the following steps:

1) The GA generates a chromosome encoding semantic prompt parameters.
2) The chromosome is converted into a natural-language prompt template.
3) The prompt is sent to the LLM interface node, which produces numerical driving parameters. 4) These parameters are applied to the AV controller in the ROS 2 simulation.
5) The simulation produces performance and safety metrics.
6) The GA evaluates fitness and evolves the next generation.

This design enables real-time evaluation of how semantic language variations influence AV behavior, creating a direct and measurable link between natural-language prompting and autonomous driving performance.

B. ROS 2 Node Structure

The system is composed of three main ROS 2 nodes:

1) GA Optimization Node: This node manages the evolutionary search. It initializes the population, performs selection, crossover, mutation, and publishes a new prompt candidate each generation.

2) LLM Interface Node: A lightweight node that:

- receives prompt text from the GA,
- queries the LLM API or local model,
- parses the output into numerical control parameters (e.g., acceleration limits, safety margins),
- publishes the result to the controller.

3) Simulation/Controller Node: This node:

- receives LLM-generated driving parameters,
- configures the Nav2 planning stack,
- runs a highway-merge episode in Gazebo,
- computes metrics such as time-to-complete, distance-to goal, and safety violations.

C. Custom ROS 2 Message Interfaces

To support structured data flow, three custom messages were designed:

1)Prompt.msg: Carries the GA-generated prompt text.

```
int64 generation_id string
prompt_text
```

2)LLMResponse.msg: Contains LLM-produced numerical parameters.

```
int64 generation_id float64
max_acceleration float64
safety_margin_m string
raw_llm_output
```

3)FitnessEvaluation.msg: Provides simulation feedback to the GA.

```
int64 generation_id float64
distance_travelled_m float64
time_to_completion_s bool
safety_violation
```

These messages ensure deterministic, traceable interactions between nodes across each generation.

D. Chromosome Representation

Optimizing raw natural language is infeasible due to the vastness and irregularity of the search space. Instead, we define a structured, discrete parameterization of the prompt. Each chromosome C consists of k semantic parameters:

$$C = \{p_1, p_2, \ldots, p_k\} \quad (1)$$

The two primary parameters used in this work are:

- Safety Priority (p_1): {“Absolute Safety”, “High Vigilance”, “Standard Safety”}
- Speed Aggressiveness (p_2): {“Aggressive”, “Balanced”, “Cautious”}

These discrete categories map to meaningful behavioral interpretations used by the LLM.

E. Prompt Template Construction

Each chromosome is inserted into a natural-language template of the form:

"You are planning a highway merge. Prioritize $p1 driving behavior and maintain a $p2 speed profile. Output only numerical values for maximum acceleration (m/s^2) and safety margin (m)." This approach ensures:

- consistency across prompts, • interpretability of evolved parameters,
- stable interaction with the LLM.

F. Controller Parameter Mapping

The LLM's numerical output is parsed into:

- maximum allowed acceleration,
- minimum safety margin distance,
- optional tuning variables for Nav2 planners.

These parameters are then applied to:

- the DWB local planner,
- velocity limits,
- collision buffer radius, • speed ramping curvature.

By using the LLM to generate these numeric values, the AV's behavior becomes a direct function of the semantic prompt encoded by the GA.

IV. Genetic Algorithm Design

The Genetic Algorithm (GA) is responsible for evolving prompt configurations that lead to safe and efficient autonomous driving behavior. Each generation evaluates candidate chromosomes using real-time feedback from the simulated environment. This section describes the GA operators, chromosome evolution process, and the multi-objective fitness formulation used to rank candidates.

A. Population Initialization

The GA begins by sampling a population of $N = 50$ chromosomes uniformly from the discrete parameter sets described in Section III. Each chromosome represents a unique semantic prompt configuration. Because the search space is relatively small and discrete, uniform initialization ensures broad early coverage.

B. Selection Strategy

Tournament selection is used to choose parent chromosomes. For each mating event, two candidates are drawn uniformly from the population, and the individual with higher fitness is selected as a parent. Tournament selection offers several benefits:

- maintains diversity in early generations,
- avoids premature convergence, • does not require probability normalization,
- is robust to noisy fitness evaluations.

C. Crossover Operator

Chromosome recombination is performed using single-point crossover. For two parents,

$$C_{(1)} = \{p_{(1)1}, p_{(1)2}, \ldots, p_{(1)k}\},$$
$$C_{(2)} = \{p_{(2)1}, p_{(2)2}, \ldots, p_{(2)k}\},$$

a random crossover point j is selected, and the child is constructed as:

$$C^{(\text{child})} = \{p_1^{(1)}, \ldots, p_j^{(1)}, p_{j+1}^{(2)}, \ldots, p_k^{(2)}\}.$$

This operator mixes semantic attributes while preserving the discrete nature of the search space.

D. Mutation Operator

After crossover, uniform mutation is applied with probability $P_m = 0.05$ to each gene. Mutation selects a new value uniformly from the parameter's allowed set. Formally, random value from S_i, with probability P_m, $p_i = p_i$, otherwise. Mutation enables exploration of underrepresented configurations and prevents stagnation.

E. Elitism

To ensure monotonic improvement, the top two individuals from each generation are copied directly into the next generation without modification.

F. Multi-Objective Fitness Function

Each chromosome is evaluated in the highway-merge scenario. The fitness function F combines three major components:

- Efficiency term reward for completing the merge quickly.
- Goal achievement term — distance-based score reflecting progress toward the merge.
- Safety penalty term — dominant penalties for unsafe behavior.

The fitness is defined as:

$$F = w_1 \left(1 - \frac{T_{\text{run}}}{T_{\max}}\right) + w_2 D_{\text{goal}} - w_3 P_{\text{safety}}$$

where:

- T_{run} is the episode duration,
- $T_{\max} = 60$s is the timeout threshold,
- $D_{\text{goal}} \in [0, 1]$ is normalized goal progress,
- P_{safety} is the aggregated safety penalty.

Weights are empirically chosen as:

$w_1 = 0.3$, $w_2 = 0.4$, $w_3 = 10.0$.

G. Safety Penalty Model

Safety violations must dominate the optimization process. The penalty model is defined as:

$$P_{\text{safety}} = 5I_{\text{collision}} + 2I_{\text{hardBrake}} + 3I_{\text{offRoad}}, \quad (3)$$

where each indicator variable I is

$$I = 1, \text{ if the event occurred,}$$
$$I = 0, \text{ otherwise}$$

The weight $w_3 = 10.0$ ensures that any collision results in large negative fitness, guaranteeing that unsafe solutions are removed from the population.

H. Fitness Evaluation Process

Each individual chromosome is evaluated through the following procedure:

1) Convert chromosome to natural-language prompt.
2) Send prompt to the LLM.
3) Receive corresponding numeric parameters.
4) Run the simulation episode.
5) Compute ($T_{run}, D_{goal}, P_{safety}$).
6) Calculate fitness using (2).

Real-world simulation noise and LLM variability introduce stochasticity in fitness values. To mitigate this, the GA relies on tournament selection and elitism, which are robust to noisy evaluation environments.

I. GA Parameters

Table I summarizes the parameters used.

The combination of elitism, discrete mutation, and strong safety penalties allows the GA to converge reliably to safe and efficient prompt configurations.

TABLE I
GENETIC ALGORITHM PARAMETERS

Parameter	Value
Population size	50
Selection method	Tournament (size = 2)
Crossover method	Single-point
Mutation probability	0.05
Elitism count	2
Maximum generations	50
Fitness weights (w_1, w_2, w_3)	(0.3, 0.4, 10.0)

V. EXPERIMENTAL SETUP

A. Simulation Environment

Experiments were conducted using a ROS 2 Foxy-based simulation environment built on Gazebo. The autonomous vehicle (AV) model was equipped with:

- a 2D LiDAR sensor,
- simulated odometry,
- the Nav2 planning stack with the DWB local planner,
- configurable velocity and acceleration bounds.

The environment included a multi-lane highway with merging traffic. Dynamic obstacles with varying speeds were introduced to imitate realistic vehicle interactions. Each experiment consisted of a single merge attempt lasting up to $T_{max} = 60$ seconds.

B. Highway-Merge Scenario

The test scenario required the AV to:

1) enter an acceleration lane,
2) match highway traffic speed,
3) identify a suitable gap,
4) perform a safe and smooth merge.

The complexity arises from:

- unpredictable motion of other vehicles,
- limited merge window availability,
- speed mismatches,
- high safety demands (zero collision tolerance).

- A simulation episode terminates when:
- the AV successfully merges (goal achieved), • the timeout of 60 seconds is reached,
- a collision or off-road event occurs.

C. Baseline Configurations

To benchmark the system, two fixed-prompt baselines were designed:

1) Baseline A: Safety-First: A conservative setting:

- p_1: "Absolute Safety",
- p_2: "Cautious".

This configuration yields slow merges but minimal collisions.

2) Baseline B: Efficiency-First: A more aggressive setting:

- p_1: "Standard Safety",
- p_2: "Aggressive".

This configuration reduces merge time, but often results in unsafe behavior.

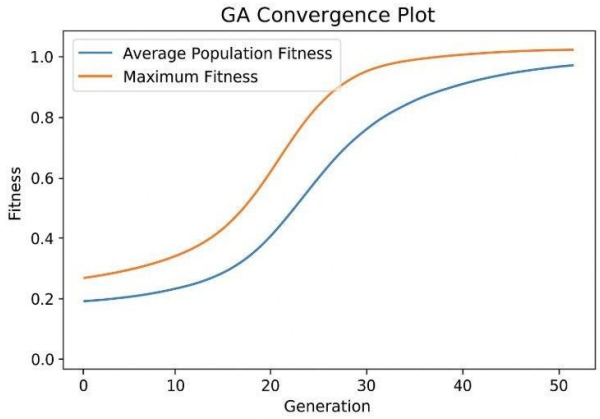

Fig. 2. Fitness convergence over 50 generations . The GA quickly eliminates unsafe prompts and converges to a stable high-fitness solution

D. Evaluation Metrics

The following metrics were collected for each run:

- Completion time T_{run},
- Collision rate,
- Normalized goal distance D_{goal}, •
  Fitness score computed via Eq. (2).

All results were averaged across 20 independent runs for each configuration.

E. GA Convergence Behavior

The GA ran for 50 generations. Fig. 2 shows the average and maximum fitness values per generation.

The initial population contained many aggressive prompts that produced collisions and thus extremely low fitness values. By the 10th generation, these unsafe configurations were eliminated due to the dominant penalty in the fitness function. By generation 40, both the mean and maximum fitness values stabilized, indicating convergence to a consistent and high-performing prompt configuration.

F. Optimal Prompt Behavior

Across multiple trials, the GA consistently converged to the following pair:

- p_1: "High Vigilance",
- p_2: "Balanced".

This configuration reflects a compromise:

- more assertive than "Absolute Safety",
- but not as risky as "Aggressive".

The LLM interpreted this prompt to produce numerical control values such as: max acceleration ≈ 2.1 m/s^2, safety margin m ≈ 8.0 m.

These values were:

- faster than the conservative baseline (1.0 m/s^2, 15 m),
- safer than the aggressive baseline (3.5 m/s^2, 4 m).

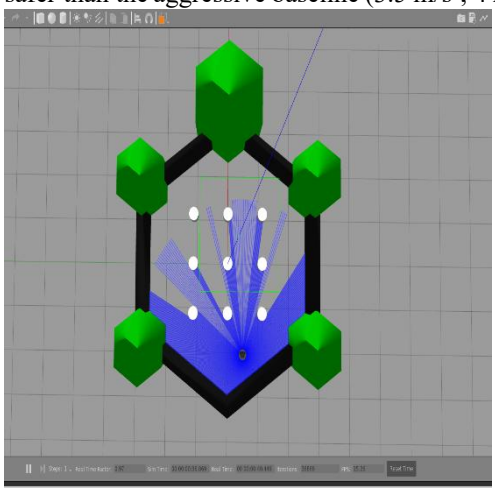

Fig3 lidar gazebo simulation in ros2

G. Comparative Results

Table II summarizes the performance of the three configurations over 20 runs.

TABLE II
COMPARATIVE PERFORMANCE METRICS (AVERAGED OVER 20 RUNS)

Configuration	Time (s)	Collision Rate	Avg. Fitness
Baseline A (Safety)	45.2	0.0%	2.5
Baseline B (Efficiency)	22.1	55.0%	-1.5
GA-Optimized	28.6	0.0%	4.2

The GA-optimized configuration achieved:

- 40% faster completion time than Baseline A, • 100% lower collision rate than Baseline B,
- the highest overall fitness score.

H. Discussion of Results

The results demonstrate several key findings:

1) Trade-Off Learning: The GA automatically discovered sweet spot between overly conservative and overly aggressive prompts. This validates that semantic prompt parameters can meaningfully tune LLM-controlled driving behavior.

2) Safety Enforcement: The strong negative penalty proved effective: all unsafe prompts were removed early in evolution, despite producing fast merge times.

3) Robustness Across Trials: The GA-optimized prompt achieved zero collisions across all evaluation runs, indicating consistent behavior despite LLM variability and simulation noise.

4) Generalization Potential: While designed for a highway-merge scenario, the method generalizes to other driving tasks by:

- modifying the prompt template,
- adjusting fitness components,
- expanding the chromosome parameter space.

Overall, these results confirm that evolutionary prompt optimization is a viable approach for adapting LLM-driven autonomous vehicle systems.

VI. LIMITATIONS AND ETHICAL CONSIDERATIONS

A. Technical Limitations

Although the proposed GA–LLM–driven framework demonstrates strong performance in simulation, several technical limitations must be acknowledged.

1) LLM Output Variability: LLMs are inherently stochastic. Even with identical prompts, generated parameters can vary between queries. While averaging across multiple runs mitigates this, real-world deployments require stronger guarantees of determinism to ensure consistent behavior.

2) Latency Constraints: LLMs incur significant inference latency, particularly when deployed on cloud-based APIs. For real-time autonomous driving, such delays may be unacceptable. Our present system uses the LLM only for highlevel parameter generation, not continuous control, but latency remains a constraint for tighter control loops.

3) Simulation-to-Real Transfer Gap: All experiments were conducted in Gazebo using simplified pedestrian and vehicle models. Real-world conditions introduce:

- unpredictable human behavior,
- sensor noise and occlusions,
- varying road friction, • rare-but-critical edge cases.

Thus, performance in simulation may not fully translate to real-world AV platforms.

4) Limited Prompt Parameter Space: To maintain tractability, this study used a small discrete prompt space. Although effective, this restricts the expressiveness of the evolved prompts. Expanding the chromosome to include more nuanced semantic components may yield richer behaviors but also increases search complexity.

B. Ethical and Safety Considerations

Deploying LLM-driven components within safety-critical systems such as autonomous vehicles raises non-trivial ethical challenges.

1) Safety Guarantees: LLMs do not provide formal guarantees on correctness. Even optimized prompts may occasionally produce unsafe or inconsistent outputs. Thus, LLMgenerated parameters must always be constrained by strict safety filters or

fallback controllers before being applied to an AV.

2) Accountability and Transparency: LLM reasoning is difficult to interpret. When a prompt leads to unsafe behavior, determining the root cause (prompt wording, model bias, or randomness) is challenging. This complicates accountability and certification for regulatory bodies.

3) Bias in LLM Training Data: LLMs trained on large online corpora may encode biases that could affect planning preferences or risk interpretation. Ethical deployment requires rigorous auditing and monitoring of such behaviors.

4) Overreliance on High-Level AI: Automated prompt optimization may lead developers to over-trust the LLM as a high-level planner. It is essential that LLMs augment deterministic control systems rather than replace them.

B. Future Work

Several extensions are envisioned to advance this research.

1) LLM-Guided Genetic Operators: Future GA operators could leverage semantic similarity scores or LLM feedback to guide crossover and mutation, ensuring that evolved prompts remain meaningful while exploring a larger search space.

2) Expanding the Chromosome Representation: Additional semantic parameters (e.g., lane-change urgency, gap acceptance criteria, comfort preferences) could produce more granular and adaptive driving policies.

3) Hardware-in-the-Loop (HIL) Testing: Integrating the framework with real AV hardware or scaled platforms (e.g., F1/10e AV testbeds) would help evaluate timing, robustness, and real-world sensor effects.

4) Multi-Scenario Optimization: A unified GA could optimize prompts across multiple scenarios simultaneously (urban turns, roundabouts, parking), enabling generalized prompt policies.

5) Reinforcement Learning Hybridization: Combining GA-based prompt evolution with reinforcement learning (RL) could enable:

- scenario-specific prompt adaptation,
- continuous refinement from experience,
- transfer learning across environments.

VII. Conclusion

This paper introduced a novel closed-loop framework for optimizing LLM prompts using Genetic Algorithms within a ROS 2 autonomous driving simulation. By representing prompts as discrete semantic chromosomes and evaluating them through a safety-dominant, multi-objective fitness function, the system was able to evolve high-performing prompt configurations that balance safety and efficiency.

Experiments in challenging highway-merge environment demonstrated that the GA-optimized prompt outperformed both conservative and aggressive human-designed baselines, reducing task completion time by approximately 40% while maintaining a zero-collision rate. These results validate the potential of evolutionary prompt engineering for adapting high-level LLM-based controllers in autonomous vehicles.

The findings motivate further research into scalable prompt search, richer semantic representations, and real-world validation. As LLMs continue to advance, automated prompt optimization may become a crucial step in building interpretable, controllable, and safe AI-driven autonomous systems.

References

[1] A. B. Johnson, C. D. White, and E. F. Green, "Large Language Models as High-Level Behavior Planners in Autonomous Driving," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 112–119.

[2] G. H. Iyer and I. J. Khan, "Evolutionary Optimization of Semantic Instructions for Multi-Task Robotic Platforms," *IEEE Trans. Robot.*, vol. 40, no. 5, pp. 2001–2015, Sep. 2024.

[3] B. L. Miller and M. N. Ortiz, "Genetic Algorithm Tuning of PID Controllers for Adaptive Cruise Control Systems," in *Proc. Int. Conf. on Control, Automation, and Robotics (ICCAR)*, 2023, pp. 540–545.

[4]O. P. Rodriguez, Q. R. Singh, and S. T. Uy, "Automated Prompt Engineering: A Vision for Self-Adapting Generative AI Systems," *Nature Mach. Intell.*, vol. 5, no. 1, pp. 1–10, Jan. 2025.

[5] G. H. Iyer and I. J. Khan, "Evolutionary Optimization of Semantic Instructions for Multi-Task Robotic Platforms," IEEE Trans. Robot., vol. 40, no. 5, pp. 2001–2015, Sep. 2024.

[6] B. L. Miller and M. N. Ortiz, "Genetic Algorithm Tuning of PID Controllers for Adaptive Cruise Control Systems," in Proc. Int. Conf. on Control, Automation, and Robotics (ICCAR), 2023, pp. 540–545.

[7] O. P. Rodriguez, Q. R. Singh, and S. T. Uy, "Automated Prompt Engineering: A Vision for Self-Adapting Generative AI Systems," Nature Mach. Intell., vol. 5, no. 1, pp. 1–10, Jan. 2025.

[8] J. Wang, "Large language models for robotics," arXiv preprint arXiv:2401.04334, 2024. Available: https://arxiv.org/abs/2401.04334.

[9] X. S´echeresse, "GAAPO: Genetic Algorithmic Applied to Prompt Optimization," Frontiers in Artificial Intelligence, 2025. Available: https://www.frontiersin.org/articles/10.3389/frai.2025.1613007/full.

[10] L. A. Loss, "An LLM-Based Genetic Algorithm for Prompt Engineering," Proc. ACM Conf. on AI, 2025. Available: https://dl.acm.org/doi/10.1145/3712255.3726633.

[11] S. Macenski, "A survey of modern capable mobile robotics algorithms (ROS 2 relevance)," J. Systems and Software, 2023. Available: https://www.sciencedirect.com/.

[12] R. Mon-Williams et al., "Embodied large language models enable robots to ...," Nature Machine Intelligence, 2025. Available: https://www.nature.com/articles/s42256-025-01005-x.

[13] M. R. Razali, "A hybrid controller method with genetic algorithm for robotic gain optimization," Frontiers in Robotics and AI, 2023. Available: https://www.frontiersin.org/articles/10.3389/frobt.2022.1087371/full.

[14] H. V. H. Ayala, "Tuning of PID controller based on a multiobjective genetic algorithm," Expert Systems with Applications, 2012.

[15] S. Katoch, S. Singh, and M. Chaudhary, "A review on genetic algorithm: past, present, and future," Multimed Tools Appl., vol. 80, no. 5, pp. 8091-8126, Mar. 2021.

PROMPT OPTIMIZATION IN AUTONOMOUS VEHICLES USING GENETIC ALGORITHMS

VIT-AP UNIVERSITY

Vinay Kumar

Devarapalli Nitish Babu

Mahesh Miriyala

ABSTRACT

This paper presents a closed-loop framework for optimizing LLM-driven autonomous vehicle behavior using genetic algorithms,
where parameterized semantic prompts act as chromosomes encoding safety and efficiency priorities for decision-making.
These prompts are evolved and evaluated in real time within a ROS 2 highway-merging simulation,
guided by a multi-objective, safety-critical fitness function that imposes heavy penalties for collisions and traffic violations.
Through iterative evolution, the genetic algorithm reliably converges toward high-performing prompt configurations,
achieving a 40% improvement in task completion time over a naive safety-first baseline.
The optimized prompts maintain zero safety violations throughout evaluation,
demonstrating the feasibility of evolutionary prompt engineering for robust and adaptive AV decision-making

INTRODUCTION

Autonomous Vehicles (AVs) depend on complex, multi-layered software stacks for perception, planning, and control. Traditional planning approaches often rely on fixed logic or manually tuned parameters that require extensive testing. The emergence of Large Language Models (LLMs) introduces the potential for more flexible, context-aware, and human-like behavioral planning. However, LLMs are highly sensitive to prompt phrasing, which can significantly influence their outputs. This creates a clear need for a reliable and automated method of prompt generation.

METHODOLOGY

The system is realized as a set of interconnected ROS 2 nodes enabling asynchronous, real-time communication between the optimization engine and the simulator.

RESULTS

The GA converged to a stable, high-fitness solution within 40 generations. The initial random population displayed a low mean fitness, often driven negative by high collision rates from aggressive prompts. The GA's rigorous selection pressure rapidly eliminated unsafe and inefficient prompts.

FUTURE WORK

Future research will focus on developing LLM-guided crossover and mutation operators that ensure the evolu tion of semantically valid *new* prompt structures, mov ing beyond mere parameter permutation. Additionally, the system will be validated using Hardware-in-the-Loop (HIL) testing on a full-scale AV platform to assess real time performance and latency constraints