Commun Transp Res

Research Article

Choreographing CAV Formations on the Fly with Reflection

Jianghong Dong[1], Wenhan Cao[2], Jiawei Wang[3], Yifeng Zhang[4,✉], Mengchi Cai[1], Chaoyi Chen[1], Qing Xu[1], Jianqiang Wang[1], Keqiang Li[1], Guillaume Sartoretti[4]

Cite this article: https://doi.org/10.26599/COMMTR.2026.9640070

ABSTRACT: Formation control of connected and automated vehicles (CAVs) has shown great potential for improving traffic efficiency and safety. However, most existing studies focus on tracking predefined formations for fixed tasks, offering limited adaptability to changing task requirements. To address this limitation, we propose GRF-TAFC, a Grounded and Reflective Framework for Task-Adaptive Formation Choreography of CAVs. GRF-TAFC integrates context-grounded formation generation, optimization-based execution, and execution-driven reflection into a closed loop. A formation generator elicits the formation generation capabilities of pretrained MLLMs and grounds their reasoning over unstructured task inputs in real-time tool outputs, domain knowledge, and experience memory, yielding formations tailored to current task requirements while respecting environmental constraints. An execution module realizes the generated formations through polynomial-time CAV-to-target assignment and joint spatio-temporal trajectory optimization, facilitating collision avoidance and safe execution. A reflection module reasons over task context and actual execution outcomes, distilling reusable experience for iterative refinement. Real-world experiments across dynamic tasks and environments demonstrate GRF-TAFC's effectiveness and practicality, with experimental videos available on the project website: https://dongjh20.github.io/GRF-TAFC.

KEYWORDS: connected and automated vehicles; multi-vehicle coordination; formation control; task adaptability; large language model

Not Accepted

1 Introduction

With the advancement of autonomous driving (Y. Liu et al., 2023) and vehicle-to-everything (V2X) technologies (Zhang et al., 2025), real-time coordination among vehicles, roadside units (RSUs), and the edge cloud has become increasingly feasible. Facilitated by these advances, cooperative driving among multiple connected and automated vehicles (CAVs) has emerged as a growing research direction alongside single-vehicle autonomous driving (Chu et al., 2021; Zhang et al., 2026a). The vehicle–road–cloud integrated system (Li et al., 2020) is a representative example, providing an effective foundational platform for multi-CAV coordination. Classical multi-CAV coordination approaches include single-platoon control (Wang et al., 2021), cross-lane multi-platoon control (Li et al., 2018), and formation control (Firoozi et al., 2021; Gao et al., 2025). Compared with the localized coordination of the former two approaches, formation control treats all CAVs as an integrated whole and jointly coordinates their longitudinal and lateral motions, thereby providing broader coordination capabilities and thus offering greater potential for improving traffic efficiency (Cai et al., 2022).

Formation control aims to coordinate CAV motions such that desired inter-CAV spatial relationships are achieved and maintained. Typical approaches include leader–follower (W. Liu et al., 2023), virtual-structure (Kamel et al., 2020), potential-function-based (Han et al., 2024), graph-based (Gao et al., 2019), and behavior-based methods (Kamel et al., 2020). These approaches offer different advantages and involve corresponding tradeoffs. Leader–follower methods simplify coordination but depend on leader information; virtual-structure methods preserve a coherent geometry but offer limited flexibility during reconfiguration. Potential functions facilitate obstacle avoidance but require suitable potential design, whereas graph-based methods support distributed coordination but depend on the interaction topology. Behavior-based methods enable flexible local responses but complicate global stability analysis. Most of these approaches use feedback laws derived from deviations between the current and desired formations. Depending on the geometric quantities used to characterize these deviations, formation-error formulations include position-based (Kamel et al., 2020), distance-based (Gao et al., 2019), displacement-based (Wu et al., 2024), and bearing-based methods (Cheng and Huang, 2024; Su et al., 2026). These formulations entail different sensing and observability requirements; for example, bearing measurements do not directly provide inter-vehicle distances (Su et al., 2026).

However, most existing studies focus on converging a formation to a predefined configuration, essentially addressing a predetermined formation task. In real-world applications, however, CAV formations need to adapt to changing task requirements while respecting constraints imposed by the traffic environment. For example, the formation may need to promptly clear the way for a high-priority vehicle approaching from behind, such as an ambulance. This motivates the development of context-grounded, task-adaptive formation choreography methods. In particular, human operators

[1] School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China. [2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, Singapore. [3] Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor 48109, USA. [4] Department of Mechanical Engineering, National University of Singapore, Singapore 119077, Singapore.

✉ Corresponding author. E-mail: yifengz@nus.edu.sg

Received: September 9, 2026; Revised: October 12, 2026; Accepted: November 2, 2026

© The Author(s) 2026. This is an open access article under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0, http://creativecommons.org/licenses/by/4.0/).

should be able to express task requirements directly through unstructured multimodal inputs, including text, speech, and images. This calls for an integrated framework that automates the entire pipeline from task interpretation to formation execution.

Large language models (LLMs), with their strong generalization, semantic understanding, and reasoning capabilities, have emerged as a promising solution for applications requiring adaptability (Wei et al., 2026; Zhang et al., 2026b). They have already demonstrated encouraging performance in the CAV domain, including directly generating testing scenarios from user inputs (Q. Shi et al., 2025), interpreting vehicle behaviors (Xu et al., 2024), and guiding scheduling at unsignalized intersections (X. Shi et al., 2025). Multimodal LLMs (MLLMs) further extend these capabilities through their well-established ability to interpret and reason over text, speech, and image inputs (Fourati et al., 2026; Tian et al., 2025).

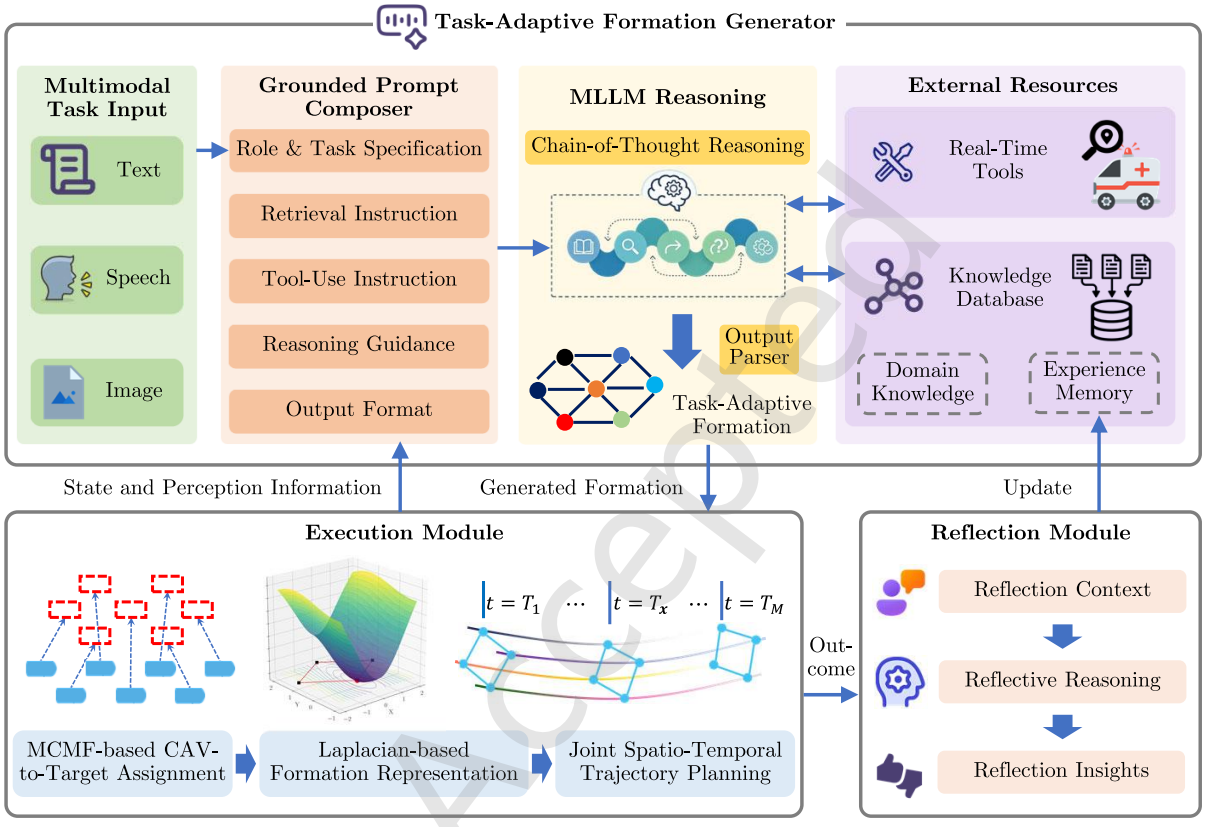


Fig. 1. Overview of the proposed GRF-TAFC framework, consisting of a task-adaptive formation generator, an optimization-based execution module, and a reflection module driven by actual execution outcomes.

Recently, several studies have explored the integration of LLMs into formation control, mostly for UAV formations. These methods can be grouped by LLMs' primary role into formation geometry generation and decision-making for formation coordination. For formation geometry generation, a representative application is to use LLMs to interpret natural-language descriptions and then employ optimization algorithms to realize the corresponding formation shapes for drone shows, as demonstrated by CLIPSwarm (Pueyo et al., 2024). FlockGPT (Lykov et al., 2024) further explores the iterative modification of desired three-dimensional swarm formations by using LLM-generated code to represent the corresponding geometric surfaces. However, these studies largely adopt open-loop architectures, without using actual execution outcomes as feedback for further refinement, thereby allowing previously observed errors to recur (Tian et al., 2025). For decision-making in formation coordination, a very recent study (Lu et al., 2025) explores the integration of LLMs into CAV formation control to assist high-level decision-making. However, it is limited to four predefined tasks and relies on predefined CAV-to-target assignments. Moreover, these studies integrating LLMs into formation control have primarily relied on simulation for validation (Lu et al., 2025; Lykov et al., 2024; Pueyo et al., 2024) and lack real-world experimental evidence of practical applicability.

To address the aforementioned challenges, we propose GRF-TAFC, a Grounded and Reflective Framework for Task-Adaptive Formation Choreography of CAVs. Here, formation choreography encompasses task interpretation, formation generation, and the realization and maintenance of the generated formations as an integrated process. "On the fly" emphasizes that new formations are generated and realized as needed during ongoing CAV formation operation. GRF-TAFC integrates context-grounded formation generation, optimization-based execution, and execution-driven reflection into a closed loop. As illustrated in Fig. 1, GRF-TAFC comprises a formation generator, an execution module, and a reflection module. The formation generator elicits pretrained MLLMs' latent capacity for formation generation by grounding their reasoning over unstructured multimodal task inputs through real-time tool invocation and the retrieval of domain knowledge and experience memory, yielding formations tailored to current task requirements while respecting environmental constraints. The execution module realizes and maintains the generated formation in a distributed manner using optimization-based methods to facilitate safe execution. The reflection module reasons over task context and actual execution outcomes, distilling reusable insights to update the

Just Accepted

MLLM-accessible experience memory for iterative refinement of formation generation. The main contributions of this paper are summarized as follows.

1) We propose GRF-TAFC, a grounded and reflective framework that choreographs CAV formations on the fly in response to changing task requirements while respecting environmental constraints, moving beyond existing studies that focus on tracking predefined formations for fixed tasks (Wu et al., 2024). To the best of our knowledge, this work is the first to achieve fully automated CAV formation choreography with constraint-aware adaptation to task requirements from unstructured multimodal inputs.
2) Unlike existing works that typically apply LLMs to formation control in an open-loop manner by directly using their generated outputs (Ahmed et al., 2026), GRF-TAFC introduces a reflection module to update LLM-accessible experience memory online based on actual execution outcomes, thereby establishing a closed feedback loop to reduce recurring errors and iteratively refine formation generation.
3) GRF-TAFC grounds MLLM reasoning through the invocation of real-time tools and the retrieval of domain knowledge and experience memory, thereby mitigating hallucination risks and promoting reliable, context-appropriate formation generation, whereas the existing LLM-based work on CAV formation control lacks such comprehensive grounding mechanisms (Lu et al., 2025).
4) We conduct real-world experiments across dynamic tasks and traffic conditions, demonstrating the effectiveness and practical applicability of GRF-TAFC, whereas existing formation studies involving LLMs predominantly rely on simulation-based validation (Li et al., 2024).

The rest of this paper is organized as follows. Section 2 formulates the three design problems. Section 3 presents the proposed methodology: Section 3.1 integrates MLLM reasoning with optimization-based execution, while Sections 3.2 and 3.3 detail the task-adaptive formation generator and execution-driven reflection module, respectively. Section 4 presents the experimental validation. Finally, Section 5 concludes the paper.

2 Problem Formulation

Consider N CAVs operating in a multi-lane driving environment, and let $\mathcal{V}$ denote the CAV set. Formation generation and reflection are activated at discrete task-related events, whereas trajectory planning and tracking control operate continuously. Accordingly, we use t to denote continuous time and k to index task requests. At time t, CAV $i \in \mathcal{V}$ has state $\boldsymbol{x}_{i,t} \in \mathbb{X}_i$, control input $\boldsymbol{u}_{i,t} \in \mathbb{U}_i$, and planar position $\boldsymbol{p}_{i,t} \in \mathbb{R}^2$, where $\mathbb{X}_i$ and $\mathbb{U}_i$ denote the admissible state and control input sets, respectively. Let $\boldsymbol{X}_t = \{\boldsymbol{x}_{i,t}\}_{i\in\mathcal{V}}$ and $\boldsymbol{u}_t = \{\boldsymbol{u}_{i,t}\}_{i\in\mathcal{V}}$ denote the collective state and control input, respectively, and let $\boldsymbol{e}_t$ denote the perceived driving environment. Hereafter, a "formation" $\mathcal{S}$ refers to an identity-free set of target positions, whereas a "formation configuration" $\mathbf{C}^{\mathrm{ref}}$ refers to its CAV-indexed realization after CAV-to-target assignment.

Given a reference formation configuration $\mathbf{C}_t^{\mathrm{ref}}$, the control law κ to minimize the deviation between the real-time and desired formation configurations can be denoted as

$$\boldsymbol{u}_t = \kappa\left(\boldsymbol{X}_t, \mathbf{C}_t^{\mathrm{ref}}; \boldsymbol{e}_t\right). \tag{1}$$

Existing formation control studies primarily focus on designing κ for a predefined and fixed $\mathbf{C}^{\mathrm{ref}}$, which corresponds to a static task. However, in real-world traffic, both the operating environment and task requirements vary over time. A fixed formation configuration $\mathbf{C}^{\mathrm{ref}}$ therefore cannot consistently accommodate dynamic and heterogeneous task requirements.

Hence, a task-adaptive formation generation policy Γ is needed to synthesize an appropriate formation $\mathcal{S}_k$ for each task request, based on the current formation state $\boldsymbol{X}_{\tau_k}$, the previous formation $\mathcal{S}_{k-1}$, the perceived environment $\boldsymbol{e}_{\tau_k}$, and the multimodal task request $\mathcal{Q}_k$, as expressed by

$$\mathcal{S}_k = \Gamma\left(\boldsymbol{X}_{\tau_k}, \mathcal{S}_{k-1}; \boldsymbol{e}_{\tau_k}, \mathcal{Q}_k\right) = \{\boldsymbol{q}_{s,k}^{\mathrm{ref}}\}_{s=1}^{N}. \tag{2}$$

Here, a human operator issues a sequence of task requests $\{\mathcal{Q}_k\}_{k=1}^{K}$ at trigger instants $0 < \tau_1 < \cdots < \tau_K$. Each $\mathcal{Q}_k$ may contain text, speech, images, or a combination thereof, and typically conveys high-level operational intent rather than explicit target coordinates. The vector $\boldsymbol{q}_{s,k}^{\mathrm{ref}} \in \mathbb{R}^2$ denotes the s-th target position in the desired formation. Although MLLMs offer a promising means of achieving adaptability, their direct application to instantiate Γ is susceptible to hallucinations, potentially leading to task failure or even safety incidents.

Subsequently, a bijective CAV-to-target assignment $\pi_k: \mathcal{V} \to \mathcal{S}_k$ needs to be computed, yielding the corresponding formation configuration $\mathbf{C}_k^{\mathrm{ref}}$. This assignment directly affects the degree of inter-CAV trajectory conflicts during the transition from $\mathbf{C}_{k-1}^{\mathrm{ref}}$ to $\mathbf{C}_k^{\mathrm{ref}}$, with poor assignments increasing collision risk. The formation configuration remains unchanged between consecutive task events:

$$\mathbf{C}_t^{\mathrm{ref}} = \mathbf{C}_k^{\mathrm{ref}}, \quad t \in [\tau_k, \tau_{k+1}).$$

The desired formation configuration then needs to be realized and maintained, which has been the primary focus of existing formation control studies. However, an appropriate architecture is needed to integrate LLM-based formation generation in (2) with optimization-based formation realization and maintenance in (1) for safe and stable operation. To characterize formation realization and maintenance, the inter-CAV interactions can be modeled by a fully connected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where the CAV set $\mathcal{V}$ serves as the vertex set, $\mathcal{E}$ is the edge set representing all inter-CAV connections, and $\boldsymbol{W} = [w_{ij}]$ is the associated weight matrix. For each $(i,j) \in \mathcal{E}$, the formation error during task k is

$$\varepsilon_{ij,t,k} = \left\|\boldsymbol{p}_{i,t} - \boldsymbol{p}_{j,t}\right\|_2 - d_{ij,k}^{\mathrm{ref}}, \tag{3}$$

where $d_{ij,k}^{\mathrm{ref}}$ denotes the desired distance between the CAVs i and j in $\mathbf{C}_k^{\mathrm{ref}}$. Task completion requires $\varepsilon_{ij,t,k}$ to converge quickly to a neighborhood of zero while all vehicle, road, actuator, and collision-avoidance constraints remain satisfied. Let $\mathcal{Y}_k$ denote the actual execution outcomes for task k, including the observed formation errors $\varepsilon_{ij,t,k}$. If these outcomes are not effectively used to inform the formation generation policy Γ, suboptimal formation decisions or even erroneous outputs may recur under similar task contexts.

In summary, task-adaptive formation choreography under environmental constraints poses three coupled problems: 1. How can an integrated framework link unstructured multimodal task inputs $\mathcal{Q}_k$ to formation generation and the realization and maintenance of structured formation configurations $\mathbf{C}_k^{\mathrm{ref}}$, enabling safe and continuous operation across successive tasks $\{\mathcal{Q}_k\}_{k=1}^{K}$? 2. How can the formation generation strategy Γ be designed to elicit pretrained MLLMs' formation generation capability while mitigating hallucination risks? 3.

Just Accepted

How can actual execution outcomes $\mathcal{Y}_k$ be effectively used as feedback to iteratively refine the formation-generation strategy Γ?

3 Methodology

In this section, we present the design and implementation details of GRF-TAFC. We first introduce the overall framework architecture, which addresses Problem 1, and then elaborate on the task-adaptive formation generator and the execution-driven reflection module, which address Problems 2 and 3, respectively. For standard formation tasks on structured multi-lane roads, we assume a common desired longitudinal following distance between consecutive CAVs within each lane, shared across all occupied lanes and adjustable across tasks. Unless the user explicitly requests another geometry, we use an interlaced arrangement as the default, with vehicles in adjacent lanes staggered longitudinally. This arrangement facilitates lane changing and safe driving and has been shown to provide the highest robustness in formation maintenance (Dong et al., 2025).

3.1. Integrating Formation Generation with Continuous Execution

3.1.1. Framework Overview

To address Problem 1 in Section 2, GRF-TAFC integrates context-grounded formation generation, optimization-based execution, and execution-driven reflection into a closed loop, as illustrated in Fig. 1. The framework comprises three modules: a task-adaptive formation generator, an optimization-based execution module, and an execution-driven reflection module. This division of responsibilities combines complementary strengths: the generator leverages pretrained MLLMs for generalization and the interpretation of unstructured multimodal task inputs, the execution module uses interpretable optimization-based methods to explicitly account for physical feasibility and safety, and the reflection module supports iterative refinement through actual execution feedback.

The task-adaptive formation generator runs in the cloud and receives unstructured multimodal task inputs $\mathcal{Q}_k$ from human operators, including text, speech, and images. A grounded prompt composer integrates these inputs with task context into a structured prompt template $\mathcal{T}$. Guided by the resulting prompt, the MLLM performs structured reasoning and interacts with two complementary external resources as needed: the real-time tool library $\mathcal{F}$ provides up-to-date environmental information, whereas the knowledge database $\mathcal{D}_k$ supplies domain knowledge and relevant experience memory. An output parser $\mathcal{P}$ then parses the MLLM output $\mathcal{O}_k$ to obtain the generated formation $\mathcal{S}_k$ and forwards it to the execution module for realization. The generator thus translates unstructured multimodal task inputs into structured formation outputs, accomplishing the key transformation from unstructured to structured information.

The execution module runs on each CAV in a distributed manner and continuously tracks the latest given formation. Upon receiving $\mathcal{S}_k$, it first computes the CAV-to-target assignment π_k to obtain the desired formation configuration $\mathbf{C}_k^{\mathrm{ref}}$. A differentiable formation maintenance cost J_{f}, constructed from the Laplacian representations of the current and desired formations, directly incorporates the formation error into trajectory optimization. A joint spatio-temporal trajectory planner continuously tracks the given formation configuration and therefore automatically optimizes trajectories for tracking $\mathbf{C}_k^{\mathrm{ref}}$ instead of $\mathbf{C}_{k-1}^{\mathrm{ref}}$ considering collision avoidance, formation maintenance, and other factors, facilitating safe execution of newly issued formations. Finally, low-level controllers compute control commands $\boldsymbol{u}_t$ to track the planned trajectories. Consequently, only the desired formation configuration $\mathbf{C}_k^{\mathrm{ref}}$ in J_{f} needs to be updated across successive tasks, while the same planner and controller continue operating. This flexible formation update design enables on-the-fly changes to the desired formation configuration tracked by the continuously operating execution module in response to task-triggered formation generation.

Therefore, GRF-TAFC employs the structured formation representation as an interface that connects task-triggered formation generation with continuous formation realization and maintenance, through which an unstructured multimodal task input $\mathcal{Q}_k$ is transformed into a task-adaptive formation $\mathcal{S}_k$ generated in response to the task and subsequently into the desired formation configuration $\mathbf{C}_k^{\mathrm{ref}}$ tracked by the continuously operating optimization-based execution module. This integration addresses Problem 1 in Section 2, supporting safe and continuous task-adaptive formation choreography across successive tasks.

The reflection module runs in the cloud and closes the generation-execution-reflection loop. It first constructs a reflection context $\mathcal{I}_k^{\mathrm{r}}$ from the original task $\mathcal{Q}_k$, the generated formation $\mathcal{S}_k$ and its generation context, and actual execution outcomes $\mathcal{Y}_k$. It then leverages an MLLM-based reflection policy Ω_ψ, with parameters ψ, to reason over this context and distill reusable insights $\boldsymbol{r}_k$. These insights are used to update the MLLM-accessible experience memory in the knowledge database $\mathcal{D}_k$, yielding $\mathcal{D}_{k+1}$. These insights can then be retrieved to guide subsequent formation generation, enabling iterative refinement through accumulated execution experience.

The complete pipeline of GRF-TAFC is summarized schematically as

$$\begin{aligned} \mathcal{Q}_k &\xrightarrow{\Gamma_k} \mathcal{S}_k \xrightarrow{\pi_k} \mathbf{C}_k^{\mathrm{ref}} \xrightarrow{\kappa} \boldsymbol{u}_t, \\ \boldsymbol{u}_t &\xrightarrow{\text{execution}} \mathcal{Y}_k \xrightarrow{\Omega_\psi} \boldsymbol{r}_k, \\ \boldsymbol{r}_k &\xrightarrow{\mathcal{D}_k \oplus (\cdot)} \mathcal{D}_{k+1} \dashrightarrow \Gamma_{k+1}. \end{aligned} \tag{4}$$

Here, Γ_k denotes the task-k application of the generation policy Γ using knowledge retrieved from $\mathcal{D}_k$, and $\oplus$ updates the experience memory with semantic deduplication. The dashed arrow denotes retrieval-based feedback from $\mathcal{D}_{k+1}$ to generation for task $k+1$, without updating the pretrained MLLM parameters. State information, environmental information, and reflection context are omitted from this schematic equation for brevity.

3.1.2. Optimization-Based Execution

We further elaborate on the implementation details of the execution module. Specifically, for the assignment stage, we adopt the polynomial-time Direction-Preference Min-Cost-Max-Flow (DP-MCMF) algorithm (Dong et al., 2025) to enable rapid computation and timely response to task requests, which directly solves for the optimal assignment by constructing an inherently conflict-free MCMF flow network. By comparison, conventional Hungarian-based methods (Cai et al., 2022) require repeated assignment trials and conflict resolution to minimize lane change conflicts during formation transitions and therefore provide no deterministic guarantee on the overall computation time. To eliminate redundant spatial information and further accelerate computation, both the current CAV positions and the

Just Accepted

target positions are expressed in a common normalized relative coordinate system, with longitudinal and lateral coordinates scaled by half the desired within-lane following distance and by the lane width, respectively. These structure-preserving transformations leave the optimal assignment unchanged; further details are provided in DP-MCMF (Dong et al., 2025). The normalized target formation is denoted by $\tilde{\mathcal{S}}_k$. DP-MCMF computes the conflict-free assignment π_k in the normalized space, which is then applied to the physical-coordinate formation $\mathcal{S}_k$ to obtain the desired formation configuration $\mathbf{C}_k^{\mathrm{ref}}$.

For the trajectory planner, we adopt the recently proposed MINCO trajectory representation (Wang et al., 2022). For classical spline representations, such as Bézier curves, temporal optimization is coupled with various constraints, thereby limiting the computational efficiency of joint spatio-temporal optimization and, consequently, the coordination capabilities of CAVs within the formation. By comparison, MINCO's temporal optimization is inherently decoupled. Accordingly, we formulate the following joint spatio-temporal trajectory optimization problem over MINCO's spatial position parameters $\mathbf{q}$ and temporal duration parameters $\mathbf{T}$,

$$\min_{\mathbf{q},\mathbf{T}} \quad \sum_{x} \lambda_x J_x + \lambda_t \sum_{m=1}^{M} T_m, \tag{5}$$

where $x \in \{\mathrm{col}, \mathrm{obs}, \mathrm{feas}, \mathrm{f}, \mathrm{contr}\}$ indexes the costs J_x for inter-CAV collision avoidance, obstacle avoidance, dynamic feasibility, formation maintenance, and control effort, respectively, and λ_x and λ_t are their weights and the trajectory-duration weight. M denotes the number of trajectory segments, and T_m denotes the duration of the m-th segment. By integrating all these factors into this unified optimization problem (5), the execution module realizes and maintains each received $\mathbf{C}_k^{\mathrm{ref}}$ while explicitly accounting for safety during execution, thereby addressing the execution requirements of Problem 1 in Section 2.

A general construction of J_x is

$$J_x = \sum_{i\in\mathcal{V}} \sum_{m=1}^{M} \frac{T_m}{H_m} \sum_{h=0}^{H_m} \left[\rho_x\left(\boldsymbol{p}_i\left(t_{m,h}\right), \widehat{\boldsymbol{P}}_{-i}\left(t_{m,h}\right)\right)\right]^3, \tag{6}$$

where H_m denotes the number of sampling intervals in segment m, $t_{m,h}$ is the corresponding h-th sampling time, and ρ_x is the pointwise cost. The vector $\boldsymbol{p}_i\left(t_{m,h}\right)$ denotes the sampled position of CAV i, while $\widehat{\boldsymbol{P}}_{-i}\left(t_{m,h}\right) = \{\hat{\boldsymbol{p}}_j\left(t_{m,h}\right)\}_{j\in\mathcal{V}\setminus\{i\}}$ collects the positions of the other CAVs at the same time instant. Specifically, for formation maintenance, the corresponding pointwise cost ρ_{f} is constructed from the squared Frobenius norm of the difference between the Laplacian matrices of graphs representing the current and desired formations:

$$\rho_{\mathrm{f}}(t) = \left\|\boldsymbol{L}_{\mathcal{G}}\left(\boldsymbol{P}(t)\right) - \boldsymbol{L}_{\mathcal{G}}(\mathbf{C}_t^{\mathrm{ref}})\right\|_{\mathrm{F}}^2, \tag{7}$$

where $\boldsymbol{P}(t)$ denotes the positions of all CAVs at time t. With graph edge weights defined as the squared Euclidean distances between the corresponding CAV pairs, ρ_{f} is differentiable with respect to the CAV positions. Setting $\rho_x = \rho_{\mathrm{f}}$ in (6) yields the formation maintenance cost J_{f} as a component of J_x, thereby allowing the formation error to be directly incorporated into the integrated trajectory optimization problem (5). After deriving the analytical gradients, we efficiently solve the resulting unconstrained optimization problem using the LBFGS-Lite optimizer to obtain the optimized trajectories. More details of the planner can be found in (Dong et al., 2025).

The planned trajectories are tracked using PID control in the longitudinal direction and preview control in the lateral direction. For each CAV, the trajectory endpoint is selected on the target lane's centerline according to the planning distance and held fixed during optimization, ensuring that the trajectory leads to the desired lane. We use the straight line from the CAV's current position to this endpoint, with uniform time allocation, as the initial solution to the unconstrained optimization in (5).

3.2. Context-Grounded Formation Generation

The task-adaptive formation generator is deployed in the cloud, which aggregates information from CAVs, roadside infrastructure, and other connected platforms while providing sufficient computational and storage resources for MLLM-based reasoning. In this paper, we use pretrained MLLMs from commercial providers without modifying their architectures or performing task-specific fine-tuning. These models provide the capabilities needed to interpret unstructured multimodal task inputs and produce structured task specifications, while road conditions, vehicle states, and formation requirements can be explicitly represented in natural language. We therefore focus on grounding these capabilities in the current task and environment and integrating formation generation with continuous execution and execution-driven reflection. The formation generation process is triggered upon receiving an unstructured multimodal task input from a human operator, and its main steps are described below.

3.2.1. Grounded Prompt Composer

To generate a feasible formation that adapts to task requirements while respecting environmental constraints, such as road geometry and traffic rules, the MLLM must be supplied with sufficient task and operational context. The grounded prompt composer therefore organizes heterogeneous contextual information into the structured prompt template $\mathcal{T}$ illustrated in Fig. 2, with a complete example provided on our project website. The template comprises the following five components.

1) Role and task specification: The prompt identifies the MLLM as an expert in multi-agent cooperative planning and specifies its task as generating a new desired formation, where the "Task Description" part specifies the relationships among the different components of the prompt. At trigger instant τ_k, the prompt supplies the information required to instantiate the formation generation problem, such as the number of CAVs, {CAV_number} in Fig. 2. Such system-derived information is automatically extracted from the current collective CAV state $\boldsymbol{X}_{\tau_k}$ and incorporated into the prompt. The prompt also provides the previous desired formation $\mathcal{S}_{k-1}$, the current environment $\boldsymbol{e}_{\tau_k}$, and the multimodal task request $\mathcal{Q}_k$. To support generalization across different roads and improve numerical stability, formations are represented using the same dimensionless relative coordinate system as that adopted for $\tilde{\mathcal{S}}_k$. The prompt specifies an interlaced arrangement as the default geometry and explicitly instructs the MLLM to follow a different geometry when requested by the user, subject to applicable road and safety constraints. When sufficient usable lanes are available, the prompt prioritizes translating the entire formation over reconfiguring its geometry, provided that the task and environmental constraints are satisfied. Otherwise, it encourages minimizing the number of lane changes needed to fulfill the task.

2) Retrieval instructions: The prompt informs the MLLM that an

Just Accepted

Role and Task Specification

[Role] You are an expert in ...

[Task Description] Based on [Task Context] and [User Task Input], use the external resources specified in "Retrieval Instructions" and "Tool-Use Instructions" as needed, and follow "Reasoning Guidance" to generate an identity-free set of target positions satisfying [Task Requirements].

[Task Context] {CAV_number} CAVs are traveling in formation on a {lane_number}-lane road, ...

[User Task Input] {user_task_input}

[Task Requirements] 1. Unless the user explicitly requests another geometry, arrange the CAVs in an interlaced pattern...

Retrieval Instructions

You have access to a structured knowledge database.
[Domain Knowledge] Invoke "explore_knowledge_graph" to retrieve ...
[Experience Memory] Invoke "explore_experience" to retrieve ...

Tool-Use Instructions

Invoke functions registered in "tool_list" when real-time information is required.

Reasoning Guidance

Follow the step-by-step reasoning workflow below:
1. Interpret [User Task Input] and determine the required formation adjustment.
...
4. Verify the generated formation against ...

Output Format

Output the new desired formation within the tags: <formation> </formation>.
Sort the target positions first by ...

Fig. 2. Template $\mathcal{T}$ to be populated by the prompt composer. The interlaced arrangement specified in "Task Requirements" serves as the default and can be overridden by explicit user requests regarding formation geometry.

accessible knowledge database $\mathcal{D}_k$ contains domain knowledge and experience from previous tasks. The prompt further specifies how to formulate a context-dependent query, retrieve relevant entries, and incorporate the retrieved evidence into formation generation. It also explicitly instructs the MLLM to give domain knowledge the highest priority among retrieved knowledge sources.

3) Tool-use instructions: The composer registers the available real-time tools and provides the information required for their selection and invocation, including their functional descriptions and input-output interfaces. The manner in which tool information is provided depends on the specific MLLM API. In our implementation using the Gemini 3.1 Pro API, function-specific descriptions and parameter schemas are supplied through the declarations associated with the corresponding callable functions, rather than being repeated in the textual prompt. Accordingly, the prompt directly specifies the set of tools available to the MLLM.

4) Reasoning guidance: Inspired by zero-shot chain-of-thought (CoT) prompting, where a simple step-by-step cue can improve model reasoning (Kojima et al., 2022), we provide an explicit reasoning workflow rather than relying on unconstrained generation. The workflow guides the MLLM to interpret the task, identify missing information, acquire the required knowledge and real-time facts, reconcile the collected evidence, determine the formation characteristics, and verify the result against the task and contextual constraints. This guidance promotes more consistent generation and reduces unsupported inferences. In particular, it requires the MLLM to perform necessary verification of the generated result before finalizing the output, thereby reducing the risk of erroneous outputs.

5) Output format: To enable automated parsing and downstream processing, the prompt requires the generated formation to be enclosed within the designated `<formation>` and `</formation>` tags. It also instructs the MLLM to omit detailed reasoning traces while using predefined tags to report the retrieved knowledge and experience memory, the invoked tools, and the returned results for post-execution reflection. In addition, because formation control primarily concerns the relative relationships among CAVs, as reflected in (3), relative target coordinates are sufficient for formation representation. Accordingly, the longitudinal coordinates are shifted such that the minimum coordinate is zero. The target coordinates are then sorted lexicographically in ascending order, with the lateral coordinate as the primary key and the longitudinal coordinate as the secondary key, thereby ensuring deterministic post-processing.

Collectively, the carefully designed template $\mathcal{T}$ organizes the complete grounded input $\mathcal{I}_k$ to the MLLM as

$$\mathcal{I}_k = \mathcal{T}\left(\boldsymbol{X}_{\tau_k}, \mathcal{S}_{k-1}, \boldsymbol{e}_{\tau_k}, \mathcal{Q}_k; \boldsymbol{M}_k, \boldsymbol{z}_k\right), \tag{8}$$

where $\boldsymbol{M}_k$ and $\boldsymbol{z}_k$ denote the retrieved knowledge and real-time tool outputs, respectively, as detailed below. Equation (8) compactly represents the accumulated grounded input; in implementation, retrieved knowledge and tool outputs may be incorporated iteratively into the reasoning context before the final response is produced.

3.2.2. *External Resources*

The external resources primarily comprise a real-time tool library $\mathcal{F} = \{g_\ell\}_{\ell=1}^{N_{\text{tool}}}$ and a knowledge database $\mathcal{D}_k$, where N_{tool} is the number of available tools. Together, these resources ground the MLLM in factual task context, thereby reducing unsupported inferences and potential hallucinations during formation generation. Notably, the prompt composer directly inserts essential information, such as the number of available lanes, into the prompt, thereby eliminating the need to query such information during reasoning.

Specifically, the real-time tool library comprises callable functions with predefined input–output interfaces that provide time-sensitive information required by the current task. During reasoning, the MLLM autonomously selects the required functions and generates their input arguments, while the corresponding functions are executed by the external application. Formation tasks commonly require functions for querying the locations of high-priority vehicles and obstacles and obtaining information about traffic incidents and extreme weather conditions. For example, querying the current position of a high-priority vehicle, such as an ambulance, allows the generator to avoid selecting a new formation that would encroach on its lane when the formation is nearby. Similarly, querying the position of the nearest obstacle provides spatial information for excluding target positions that may create collision risks. For task k, the returned real-time information is abstracted as the outputs of a task-dependent subset $\mathcal{F}_k \subseteq \mathcal{F}$:

$$\boldsymbol{z}_k = \left\{ g_\ell\left(\boldsymbol{X}_{\tau_k}, \mathcal{S}_{k-1}, \boldsymbol{e}_{\tau_k}, \mathcal{Q}_k\right) \;\middle|\; g_\ell \in \mathcal{F}_k \right\}. \tag{9}$$

This design ensures that time-sensitive information is obtained from real-time tools rather than inferred from the MLLM's internal knowledge or retrieved from historical database records.

The knowledge database $\mathcal{D}_k$ provides persistent information that complements real-time observations, where $\mathcal{D}_k$ denotes the database state available prior to task k. It comprises domain knowledge and experience memory. The domain knowledge is primarily constructed offline before deployment, whereas the experience memory

Just Accepted

continuously accumulates during operation.

Precisely, to address practical formation task requirements, we organize the domain knowledge into the following three categories:

1) Task interpretation knowledge: This category maps common high-level human intentions to corresponding formation-adjustment objectives. For example, yielding to a high-priority vehicle means clearing the corresponding lane; avoiding an obstacle ahead requires excluding its occupied lane from the new formation; expanding the safety margin requires increasing the longitudinal inter-CAV spacing; and forming a more compact formation requires decreasing that spacing.

2) Traffic regulations: Formation generation is closely related to lane-use restrictions, right-of-way requirements, and other applicable traffic rules. For example, target positions must not be placed in a prohibited emergency lane; the lane occupied by an approaching emergency vehicle should be cleared; and if a real-time tool reports that a lane ahead is closed because of a traffic accident, no target positions should be assigned to that lane.

3) Formation-level safety and feasibility knowledge: This category specifies key safety and feasibility principles at the formation level. For example, longitudinal inter-CAV spacing should be increased under heavy rain; the number of lanes occupied by the formation should be reduced when the road narrows; and the generated formation should respect minimum longitudinal and lateral inter-CAV distances as well as a minimum clearance from obstacles.

Because these knowledge items exhibit explicit logical relationships, we store the domain knowledge as a graph. This representation preserves their semantic dependencies and supports flexible relation-aware and multi-hop retrieval across human intents, formation adjustments, traffic regulations, and safety constraints. For example, the intent *increase the safety margin* is connected through a `MAPS_TO` relation to the formation adjustment *increase the desired longitudinal inter-CAV spacing*.

On the other hand, the experience memory accumulates from reflections on the physical execution outcomes of previously generated formations. It is typically updated automatically using the outputs of the reflection module, while human operators may also add entries manually. By preserving evidence from actual executions, the experience memory provides quantitative references for subsequent related generation instead of relying solely on abstract rules. In particular, it helps the MLLM avoid repeating errors arising from unforeseen factors. Formation generation can thus be continually improved without resource-consuming task-specific fine-tuning of the MLLM.

The experience memory stores concise key-value text pairs, where the key specifies the applicable task context and the value records the recommended formation adjustment and supporting rationale. For a new task, stored keys are ranked by the cosine similarity between their text embeddings and that of the current task context, and the values associated with the top-ranked keys are retrieved. For example, an experience entry can be stored with the key *"Reduction of available lanes from three to two for a seven-CAV formation with a 2–3–2 lane-wise distribution."* and the value *"Allocate four target positions to Lane 2 and three to Lane 1 rather than the reverse, as this reduces the required number of lane changes from four to three."*

Let $\mathcal{R}$ denote the retrieval mapping. Relevant knowledge is obtained as

$$\boldsymbol{M}_k = \mathcal{R}\left(\mathcal{D}_k; \boldsymbol{X}_{\tau_k}, \mathcal{S}_{k-1}, \boldsymbol{e}_{\tau_k}, \mathcal{Q}_k\right). \tag{10}$$

Here, $\mathcal{R}$ uses the task semantics and current operating context to retrieve a compact set $\boldsymbol{M}_k$ of relevant domain knowledge and prior experiences from $\mathcal{D}_k$. The two types of external resources play complementary roles: the real-time tool outputs $\boldsymbol{z}_k$ provide physical facts about what is currently happening, whereas the retrieved knowledge $\boldsymbol{M}_k$ guides the MLLM in determining how the situation should be handled based on either domain knowledge or experience memory.

When retrieved experience memory conflicts with domain knowledge, the domain knowledge takes precedence. It is established offline from domain principles, traffic regulations, and formation-level safety requirements, whereas reflection-generated experience may contain reasoning errors or context-specific recommendations. Experience therefore provides supplementary guidance only when consistent with the applicable domain knowledge.

3.2.3. MLLM Reasoning

Guided by $\mathcal{I}_k$, the MLLM follows a four-stage reasoning process. *1) Intent structuring:* The MLLM interprets the unstructured multimodal task request and reasons about its underlying intent to formulate a structured and actionable formation adjustment objective. *2) Evidence acquisition:* Based on the available task context, the MLLM identifies and acquires the required domain knowledge and real-time information through retrieval and tool invocation. *3) Formation synthesis:* It selects the usable lanes, determines the number of target positions in each lane, and generates the coordinate sequence according to the task requirements. *4) Self-checking and output:* Before producing the final output, the MLLM is instructed by the prompt template to check the candidate formation against task requirements and mandatory formation-level constraints, including obstacle avoidance, clearance of lanes occupied by high-priority vehicles, and exclusion of incident-affected or unavailable lanes. If it identifies a violation, the reasoning workflow returns to Step 1 to revise the candidate. This internal self-check uses the MLLM's contextual understanding and reasoning to improve generation quality.

The above reasoning process is represented by the MLLM-based formation generation policy Γ_ϕ, with parameters ϕ, which produces the structured output $\mathcal{O}_k$ from the grounded input $\mathcal{I}_k$:

$$\mathcal{O}_k = \Gamma_\phi(\mathcal{I}_k). \tag{11}$$

The output parser $\mathcal{P}$ then extracts the normalized formation from $\mathcal{O}_k$ using the predefined output tags:

$$\tilde{\mathcal{S}}_k = \mathcal{P}(\mathcal{O}_k). \tag{12}$$

After the MLLM produces its output, the output parser independently checks the required output tags and verifies that exactly N two-dimensional target positions with finite numeric coordinates are provided. It converts the relative coordinates to physical coordinates using half the desired within-lane following distance and the lane width, then checks admissible road regions and lane availability, exclusion of lanes that must be cleared for high-priority vehicles or incidents, and prescribed inter-CAV separation and obstacle-clearance limits. These rule-based checks assess formation-level admissibility; dynamic feasibility is addressed by the subsequent trajectory optimizer. Only a validated formation $\mathcal{S}_k$ is passed to execution.

Just Accepted

Role and Task Specification

[Role] You are an expert in execution-outcome-based reflection ...
[Task Description] Based on the reflection context including "What was requested", "How the formation was generated", and "What actually happened", follow "Reasoning Guidance" to generate the reflection insights specified in "Output Format".

What was requested

[Task Context] {CAV_number} CAVs are traveling in formation on a {lane_number}-lane road,...
[User Task Input] {user_task_input}

How the formation was generated

[Retrieved Domain Knowledge] {retrieved_domain_knowledge}
[Retrieved Experience Memory] {retrieved_experience_memory}
[Invoked Real-Time Tools] {invoked_real_time_tools}
[Outputs of Real-Time Tools] {outputs_of_real_time_tools}
[Generated Formation] {generated_formation}

What actually happened

[Task Results] {successful_convergence / execution_failure / timeout}
[Safety Checks] Number of collisions: {collision_num}, ...
[Execution Performance] Number of lane changes: {lane_change_num}, ...

Reasoning Guidance

Follow the step-by-step reasoning workflow below:
1. Assess whether the generated formation satisfies the task requirements and ...
...
5. Derive a concise and reusable reflection insight, including ...

Output Format

Output the reflection insight within the <experience> </experience> tags as a key-value pair, ...

Fig. 3. Reflection template $\mathcal{T}^{\mathrm{r}}$ used in the reflection module.

If an output fails validation, it is rejected and regeneration is triggered. After three consecutive invalid outputs for the same task, the system switches to a rule-based generator that constructs a basic interlaced formation using the currently available lanes. The rule-based candidate undergoes the same validation; if it remains infeasible or no usable lane is available, the system initiates a safe stop. Each CAV also has an independently operating automatic emergency braking (AEB) module as a final safeguard.

For example, given the human request "Watch out for the ambulance!", the MLLM interprets it as a yielding task, invokes a real-time tool to locate the ambulance, retrieves domain knowledge indicating that its lane should be cleared, retrieves experience memory providing a reference layout with minimal lane-change demand, and generates a new formation over the remaining usable lanes. The resulting formation is then transmitted to the execution module for execution only after it passes the prescribed validation checks.

The carefully designed prompt template $\mathcal{T}$ guides pretrained MLLMs through a structured reasoning process encompassing task interpretation, evidence acquisition, formation generation, and output validation, eliciting their capability to translate task intent into context-appropriate formations. The invocation of real-time tools and retrieval of domain knowledge and experience memory ground this reasoning process in task-relevant evidence to mitigate hallucination risks, while programmatic validation checks the resulting outputs against environmental constraints, providing a final safeguard. Together, these mechanisms constitute the formation generation strategy Γ and address Problem 2 in Section 2.

3.3. Execution-Driven Reflection

The execution-driven reflection module also runs in the cloud, as does the task-adaptive formation generator. It first constructs a comprehensive reflection context $\mathcal{I}_k^{\mathrm{r}}$ by integrating three categories of information: (1) "what was requested", including the user task input and task context; (2) "how the formation was generated", including the retrieved domain knowledge and experience memory, the invoked real-time tools and their outputs, and the resulting formation, all of which are extracted from the MLLM output using predefined tags; and (3) "what actually happened", represented by the actual execution outcome $\mathcal{Y}_k$. The former two are available in the cloud, whereas the third is provided by an execution evaluator within the reflection module. Specifically, after a new formation is issued, the evaluator monitors the formation state until successful convergence, execution failure, or timeout, and then summarizes the execution outcomes $\mathcal{Y}_k$ in terms of task results, safety checks, and execution performance. These respectively capture task satisfaction and termination status, safety and traffic-rule compliance, and the number of lane changes, transition time, and transition distance.

After the reflection context is constructed, a new prompt composer integrates relevant information into the reflection prompt template $\mathcal{T}^{\mathrm{r}}$ illustrated in Fig. 3 for reflective reasoning. The prompt first specifies the MLLM's role and task, and then instructs it to perform reflective reasoning based on the constructed reflection context $\mathcal{I}_k^{\mathrm{r}}$ and the provided reasoning guidance, thereby distilling reflection insights in the required output format. A complete example of the reflection prompt is provided on our project website. The reflection policy Ω_ψ derives the reflection insight $\boldsymbol{r}_k$ as

$$\boldsymbol{r}_k = \Omega_\psi\left(\mathcal{T}^{\mathrm{r}}\left(\mathcal{I}_k^{\mathrm{r}}\right)\right). \tag{13}$$

In particular, the reflection insight is required to be directly output as a key-value pair, facilitating integration into the key-value-based experience memory. The generated insight can then be efficiently extracted from the MLLM output via predefined tags and used to update the experience memory in the database as

$$\mathcal{D}_{k+1} = \mathcal{D}_k \oplus \boldsymbol{r}_k. \tag{14}$$

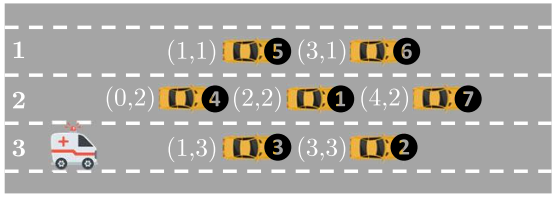
Here, $\oplus$ incorporates semantic deduplication: existing entries are retrieved using key similarity, and their values are compared for equivalent formation adjustments and supporting rationales. If an equivalent entry is found, insertion is skipped; otherwise, the new insight is stored.

This execution-driven mechanism addresses Problem 3 in Section 2 by distilling reusable experience from actual execution outcomes $\mathcal{Y}_k$ and feeding it back to the formation generation strategy Γ through the experience memory, thereby enabling iterative refinement and reducing recurring errors.

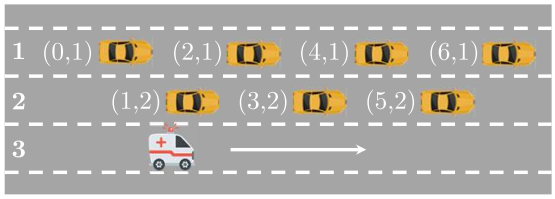
3.4. Timing and Communication Considerations

Normal operation of GRF-TAFC depends on timely task issuance, prompt MLLM responses, and reliable communication between the cloud and the CAVs. For ambulance yielding, formation generation with the Gemini 3.1 Pro API takes 36.77 s, while simulations show that the three-lane-to-two-lane formation transition is completed in 5.52 s at a desired CAV speed of 20 m/s. The estimated end-to-end time from task issuance to completion is therefore approximately 42.3 s. For an ambulance traveling at 20 m/s, the task should thus be issued when it still has at least approximately 846 m to travel before reaching the intended yielding location. Traffic managers or formation operators can issue the yielding task in a timely manner based on location data from the approaching connected ambulance. However, large MLLM

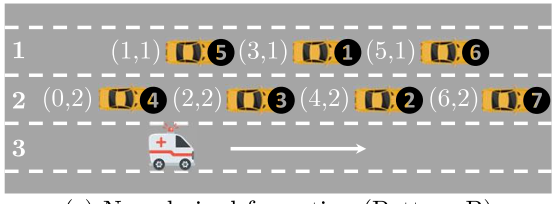
Just Accepted



(a) Current desired formation



(b) New desired formation (Pattern A)



(c) New desired formation (Pattern B)

Fig. 4. Formation generation for an ambulance-yielding scenario with two new desired formation patterns, where Pattern B represents the optimal formation in terms of the minimum number of required lane changes. The CAV indices in (a) and (c) denote the corresponding vehicles in the old and new formations, as determined by the optimal CAV-to-target assignment.

response or cloud-CAV communication delays increase the required lead time and may prevent timely task completion, particularly for tasks with stringent real-time requirements.

4 Experiments

In this section, we evaluate formation generation under different MLLM choices and module configurations, quantify repeated on-the-fly formation generation and execution across diverse tasks, and validate GRF-TAFC in the real world using a scaled experimental platform. For real-world trajectory execution, the weights in (5) for inter-CAV collision avoidance, obstacle avoidance, dynamic feasibility, formation maintenance, control effort, and trajectory duration are set to 10^4, 10^9, 10^4, 10^2, 10^3, and 10^3, respectively. Formation switching and recovery experiments indicate that optimization weights should be adapted to the scenario: for example, the minimum inter-vehicle Euclidean distance generally decreases with increasing desired speed during switching, motivating a higher collision-avoidance weight at higher speeds. Detailed experimental results are provided on the project website. We solve the trajectory optimization using the LBFGS-Lite optimizer with a minimum line-search step size of 10^{-32}. Trajectory replanning runs at 20 Hz.

4.1. Formation Generation Comparison

We consider the representative high-priority vehicle yielding scenario shown in Fig. 4(a) and use the prompt template illustrated in Fig. 2 to compare formation generation results across different MLLMs and configurations. We compare nearly all mainstream models and the reflection module consistently uses Gemini 3.1 Pro in all experiments. A common user task input, "Watch out for the ambulance!", is used throughout the comparison. In this task, to yield to the ambulance in Lane 3, Pattern A shown in Fig. 4(b) requires two CAVs to move to Lane 1 and two to Lane 2, resulting in four lane changes. In contrast, Pattern B shown in Fig. 4(c) requires only one CAV to move to Lane 1 and two to Lane 2, resulting in three lane changes, and therefore represents the better choice for the new desired formation. Lane changes should be minimized because they may increase the likelihood of trajectory conflicts among CAVs and potentially lead to collisions.

Table 1 presents the formation generation results, including an ablation comparison of repeated generation with and without reflection. Kimi K3 and Claude Sonnet 5 both generate the non-optimal Pattern A in the first attempt and retain it in the second attempt without reflection, requiring four lane changes. With reflection enabled after the first attempt, both models instead generate the optimal Pattern B in the second attempt, reducing the required lane changes to three. This ablation experiment demonstrates that the reflection module enables iterative self-refinement of formation generation. We also test a different but similar ambulance-yielding scenario, where both models reuse the learned experience to generate the formation requiring the fewest lane changes. This result demonstrates that experience can be transferred across similar scenarios; experimental details are available on the project website.

Table 1. Formation generation comparison across different MLLMs and ablation configurations.

MLLM	Generated Formation	Lane-Change Count	Generation Time (s)
Grok 4.6	Pattern B (✓)	3	44.75
Kimi K3	Pattern A (✗)	4	24.14
Kimi K3 (second turn, without reflection)	Pattern A (✗)	4	12.05
Kimi K3 (second turn, with reflection)	Pattern B (✓)	3	15.45
Claude Sonnet 5	Pattern A (✗)	4	14.09
Claude Sonnet 5 (second turn, without reflection)	Pattern A (✗)	4	6.02
Claude Sonnet 5 (second turn, with reflection)	Pattern B (✓)	3	7.97
DeepSeek V4 Pro	Pattern B (✓)	3	128.85
Qwen3.8-Max	Pattern B (✓)	3	41.62
GPT-5.6 Sol	Pattern B (✓)	3	16.48
Gemini 3.1 Pro (without tools)	Pattern C (✗)	/	50.72
Gemini 3.1 Pro (without knowledge)	Pattern B (✓)	3	32.04
Gemini 3.1 Pro	Pattern B (✓)	3	36.77

Note: ✗ indicates a non-optimal formation, not necessarily an infeasible one.

For the ablation study with Gemini 3.1 Pro, when real-time tools are unavailable, the MLLM cannot determine the current lane of the ambulance and instead assumes that it is in Lane 1. It consequently generates Pattern C, which has the same geometry as Pattern A shown in Fig. 4(b) but is shifted to Lanes 2 and 3. This clearly demonstrates the importance of real-time tool invocation; without access to up-to-date information, the MLLM may rely on unsupported assumptions and generate infeasible formations. Meanwhile, removing domain knowledge does not prevent successful formation generation, as the latest MLLM already possesses common traffic knowledge, such as the requirement to yield to ambulances. Overall, all configurations except the one without real-time tools generate feasible formations, demonstrating the effectiveness of the task-adaptive formation

Just Accepted

generator design in GRF-TAFC. Across the evaluated models, the median formation generation time is approximately 37 s, which is sufficiently short for common task-level formation adaptation tasks and suggests that MLLM-based reasoning is well suited to such tasks. DeepSeek V4 Pro performs careful and repeated reflection and verification, resulting in the longest generation time.

4.2. *On-the-Fly Formation Choreography in Simulation*

We evaluate on-the-fly formation choreography for seven CAVs traveling at a desired speed of 20 m/s in simulation. We consider ten conditions, as shown in Table 2, covering five task categories: (1) lane unavailability, requiring the formation to vacate Lane 1 or Lane 3; (2) accident-induced closure of Lane 1 or Lane 2 on a four-lane road; (3) increasing and decreasing longitudinal spacing; (4) speech-driven structure switching from an interlaced to a parallel formation, followed by moving the single vehicle at the front of the parallel formation to its rear ("Single vehicle to rear"); and (5) image-guided circular-shape generation while keeping the vehicles on both sides fixed, followed by using the three middle vehicles as the radius reference. Each condition is repeated ten times independently with the same initial experience memory; the lane-unavailability and accident conditions are tested independently, whereas the other task pairs are executed sequentially. These repeated experiments use the Gemini 3.1 Pro API with Low reasoning effort, whereas all models in Table 1 use Medium reasoning effort.

A task is considered complete when the root-mean-square slot-position error falls below 0.5 m. This metric measures the root-mean-square distance between each CAV and its assigned target position after removing the formation's common longitudinal translation. Generation time runs from task submission to receipt of the complete MLLM response, and completion time from formation dispatch to convergence.

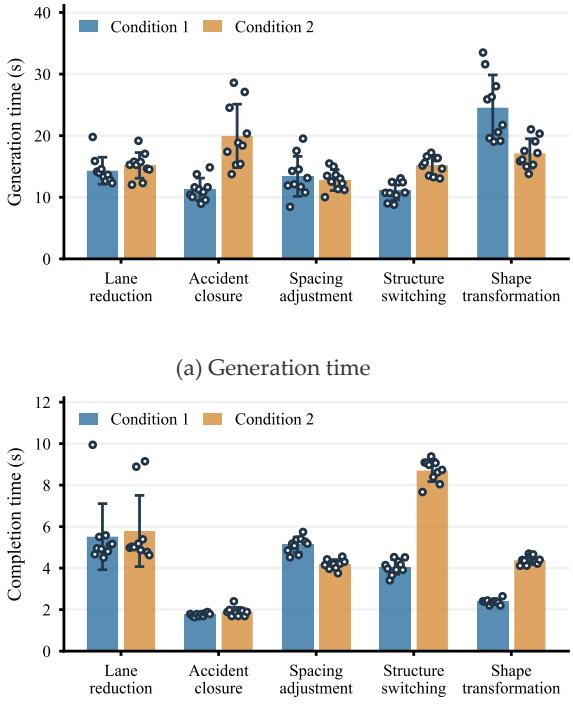
Table 2. Results over ten trials per condition. Times are reported as mean ± sample standard deviation.

Condition	Generation success	Optimal output	Generation time (s)	Completion time (s)
Lane 1 unavailable	10/10	9/10	14.31 ± 2.19	5.52 ± 1.59
Lane 3 unavailable	10/10	8/10	15.18 ± 2.08	5.79 ± 1.72
Accident: Lane 1	10/10	10/10	11.29 ± 1.82	1.76 ± 0.08
Accident: Lane 2	10/10	3/10	19.96 ± 5.15	1.90 ± 0.21
Spacing: Increase	10/10	10/10	13.39 ± 3.24	5.12 ± 0.37
Spacing: Decrease	10/10	10/10	12.79 ± 1.70	4.18 ± 0.23
Interlaced to parallel	10/10	10/10	11.07 ± 1.49	4.06 ± 0.36
Single vehicle to rear	10/10	10/10	15.18 ± 1.51	8.71 ± 0.53
Circle: sides fixed	10/10	10/10	24.54 ± 5.32	2.38 ± 0.13
Circle: middle reference	10/10	10/10	17.12 ± 2.38	4.39 ± 0.21

The experimental results are presented in Table 2 and Fig. 5. Successful generation means passing the explicit format and formation-feasibility checks; optimal outputs prioritize feasible whole-formation translation over reconfiguration and otherwise minimize the number of lane changes. Across 100 task instances, the generation success rate is 100% and the invalid-output rate is 0%; 100 of the 100 dispatched targets are successfully realized. Across the evaluated conditions, mean generation times range from 11.07 to 24.54 s, and mean completion times from 1.76 to 8.71 s. The optimal outputs are not consistently achieved across these independent repetitions, further highlighting the importance of the execution-driven reflection module for iterative self-refinement, since the MLLM alone may not consistently yield optimal outputs. Detailed task instructions and complete experimental videos are available on the project website.

4.3. *Real-World Demonstration*

We validate the effectiveness of the proposed GRF-TAFC framework through real-world experiments, whereas existing formation studies involving LLMs are largely confined to simulation-based validation (Lu et al., 2025). We use the MSH-MCCT experimental platform (Dong et al., 2026) shown in Fig. 6(a), with an operating area of 9 m × 5 m. Each scaled vehicle is equipped with a Raspberry Pi 4B and communicates directly with the cloud through Robot Operating System (ROS) messages, while the execution module runs onboard each vehicle. Markers attached to the vehicles enable their poses to be obtained in real time using the FZMotion motion capture system. The formation generator and reflection module run on a cloud host equipped with an Intel Xeon Gold 5220R CPU and an NVIDIA RTX 3090 GPU, which is connected to the roadside motion capture cameras via a wired network. The cloud host also provides a task input interface through which a human operator can issue dynamic tasks to the running formation. Both formation generation and reflection use the Gemini 3.1 Pro API. For speech and image inputs, the prompt composer sends the recorded audio or the specified image file, respectively, together with the textual prompt to the MLLM.



(a) Generation time

(b) Completion time

Fig. 5. Recorded generation and execution times across five task categories. Conditions 1 and 2 follow the row order within each task category in Table 2, with ten repetitions each. Bars and error bars indicate means and standard deviations; circles show individual measurements.

We conduct real-world experiments involving five categories of dynamic tasks: longitudinal spacing adjustment, yielding to high-

Just Accepted

priority vehicles, obstacle avoidance, formation structure switching, and formation shape transformation. Videos of all experiments are available on our project website. Taking the high-priority vehicle yielding task shown in Fig. 6(a) as an example, when the human operator issues the instruction "Watch out for the ambulance!", the output of the formation generator is shown in Fig. 6(b), and the generated formation corresponds to that shown in Fig. 4(c). The execution module then realizes and maintains the generated formation, transitioning the vehicles from the original three-lane formation to a two-lane formation that clears the ambulance lane.



(a) Yielding to high-priority vehicles

<tool> get_nearest_ambulance_lane </tool>
<results> 3 </results>

<knowledge> (Ambulance, IS_A, High-priority vehicle);
(Yielding to a high-priority vehicle, MAPS_TO, Clearing the lane occupied by that vehicle) </knowledge>

<experience> {key: Reduction of available lanes from three to two for a seven-CAV formation with a 2–3–2 lane-wise distribution, value: Place four target positions in Lane 2 and three in Lane 1, rather than the reverse, because this reduces the lane-change count from four to three.} </experience>

<formation> [(1,1), (3,1), (5,1), (0,2), (2,2), (4,2), (6,2)] </formation>

(b) Output from the formation generator



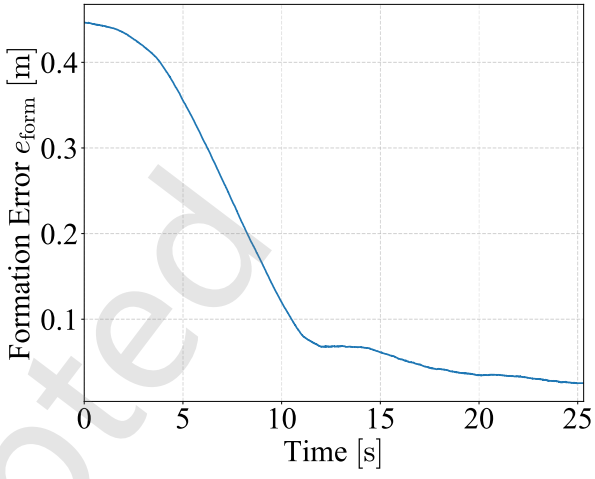
(c) Snapshots of the yielding process

Fig. 6. High-priority vehicle yielding experiment, with experimental videos available on the project website. (a) Experimental scenario, where the ambulance is blocked by the CAV formation. (b) Output of the task-adaptive formation generator, with the human operator task input: "Watch out for the ambulance!" (c) Snapshots of the experiment, with the simulated ambulance circled in red.

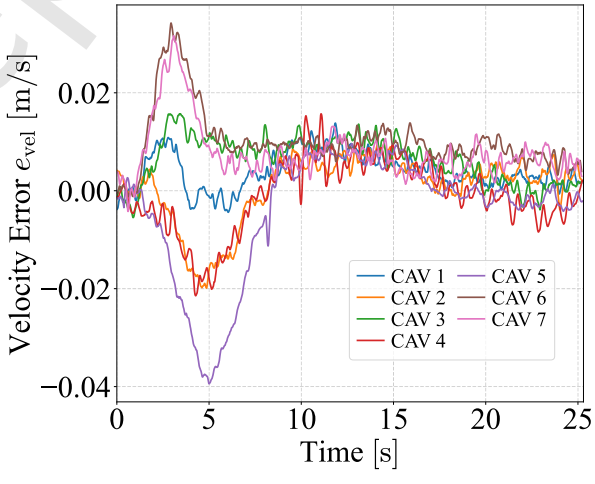
Experimental snapshots are shown in Fig. 6(c), while the corresponding formation and velocity errors are presented in Fig. 7. The CAV indices in the velocity error plot correspond to those shown in Fig. 4, which also presents the optimal CAV-to-target assignment during the formation transition. The formation error is defined as the normalized pairwise distance error among all CAVs:

$$e_{\text{form}}(t) = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|d_{ij}(t) - d_{ij}^{*}\right|, \tag{15}$$

where $d_{ij}(t)$ and d_{ij}^{*} denote the actual and desired distances between CAVs i and j, respectively. The velocity error is the difference between the actual velocity and the desired formation velocity. The experimental results show that all seven CAVs smoothly transition from a three-lane formation to a two-lane formation to yield to the ambulance and subsequently stably maintain the new formation. This demonstrates the effectiveness of GRF-TAFC in adaptively choreographing CAV formations on the fly in response to unstructured task inputs while respecting environmental constraints.



(a) Formation error



(b) Velocity error

Fig. 7. Formation error and velocity error in the high-priority vehicle yielding experiment.

GRF-TAFC adapts to task inputs while respecting varying environmental constraints. In the yielding experiment shown in Fig. 6(c), no lane is available to the left of the formation, while the ambulance approaches from the right, making formation transition unavoidable. Under the same user task input, "Watch out for the ambulance!", Fig. 8 presents a different traffic condition: the ambulance approaches from the left, while an available lane remains on the right. The formation therefore shifts laterally to the right instead of undergoing a formation transition that could introduce vehicle conflicts and potential collision risks. This experiment further demonstrates the constraint-aware adaptability of GRF-TAFC, as the MLLM-based formation generator flexibly adapts the formation to task requirements while strictly respecting the constraints imposed by the current operating environment, rather than relying on a fixed rule-based mapping from tasks to predefined strategies.

Under Review
Accepted

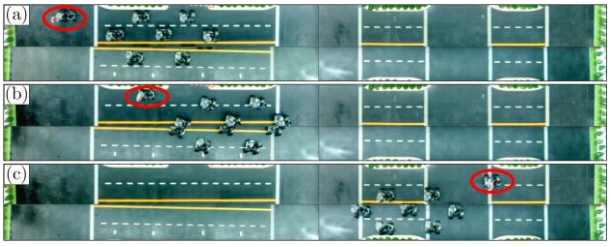


Fig. 8. High-priority vehicle yielding experiment with the user task input "Watch out for the ambulance!" provided via text. Unlike the experiment in Fig. 6(c) with the same task input, an additional lane is available on the right side of the formation, so the entire formation shifts to the right instead of undergoing a formation transition that could introduce vehicle conflicts and potential collision risks.



Fig. 9. Formation structure switching experiment with two consecutive task inputs provided via speech: "Change the interlaced formation to a parallel formation." and "For this parallel shape, move the single vehicle to the rear instead of the front."

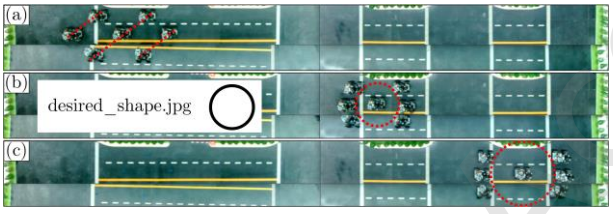


Fig. 10. Formation shape transformation experiment with two consecutive user task inputs: "Change the formation shape to match desired_shape.jpg in the current directory while keeping the vehicles on both sides fixed." and "For the new circle shape, use the three middle vehicles as the radius reference instead of the vehicles on both sides." The first input combines the text instruction with the image desired_shape.jpg; the second is provided via text.

Figs. 9 and 10 show formation structure switching driven by two consecutive speech commands and formation shape transformation driven by an image-based request followed by a text instruction, respectively. In both experiments, the generated formations are executed by the physical vehicles, demonstrating the complete process from multimodal task input to formation realization; full videos and additional details are available on the project website. Overall, these real-world experiments demonstrate the effectiveness and practical applicability of GRF-TAFC. Together with the formation generation comparisons, these results provide empirical evidence that the integrated architecture, context-grounded generation, and execution-driven refinement collectively address Problems 1-3 in Section 2 and thus enable task-adaptive formation choreography under environmental constraints, marking a meaningful step toward practical deployment in real-world traffic environments.

5 Conclusion

In this paper, we propose GRF-TAFC, a Grounded and Reflective Framework for Task-Adaptive Formation Choreography of CAVs, integrating context-grounded formation generation, optimization-based execution, and execution-driven reflection into a closed loop. The generator elicits pretrained MLLMs' formation generation capability through structured reasoning grounded in real-time tool outputs, domain knowledge, and experience memory, thereby mitigating hallucination risks and enabling task-adaptive formation generation that respects environmental constraints. The execution module continuously realizes and maintains the generated formations through conflict-free CAV-to-target assignment and joint spatio-temporal trajectory optimization, enabling on-the-fly formation updates. The reflection module reasons over actual execution outcomes together with the task context to distill reusable insights and then update the MLLM-accessible experience memory, thereby enabling iterative refinement of formation generation. Comparative experiments demonstrate the feasibility of task-adaptive formation generation and the importance of the proposed module designs, while real-world experiments validate the effectiveness and practical applicability of GRF-TAFC for task-adaptive formation choreography under environmental constraints.

In future work, we will continue to advance task-adaptive CAV formation choreography under environmental constraints toward practical deployment in real-world traffic environments. First, we will extend the role of MLLMs beyond the decision level considered in this work to lower-level planning and control, while also broadening their decision scope to include autonomous task selection, driving-speed determination, and other high-level decisions. Second, we will move beyond the scaled experimental platform and evaluate the proposed framework on full-scale CAVs. Finally, we will further enhance the framework to handle more complex scenarios involving concurrent tasks, task conflicts, and other practical challenges.

Author contributions

Jianghong Dong: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft. Wenhan Cao, Jiawei Wang, Yifeng Zhang, Mengchi Cai, and Chaoyi Chen: Methodology, Writing – review and editing. Qing Xu, Jianqiang Wang, Keqiang Li, and Guillaume Sartoretti: Supervision, Writing – review and editing, Funding acquisition.

Replication and data sharing

The data and materials supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

This work is supported by National Natural Science Foundation of China, Science Fund for Creative Research Groups (52221005), and Tsinghua University-Didi Joint Research Center for Future Mobility.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The authors Jianqiang Wang and Keqiang Li are the Editorial Board Members of this journal.

Just Accepted

References

Ahmed, A., Wang, L., Kim, J., Jin, J., Cho, K., Kwon, C., et al., 2026. LLM-guided distributed model predictive control for decentralized UAV formations. IEEE Access, 14, 15226–15240.

Cai, M., Xu, Q., Chen, C., Wang, J., Li, K., Wang, J., et al., 2022. Formation control with lane preference for connected and automated vehicles in multi-lane scenarios. Transportation research part C: emerging technologies, 136, 103513.

Cheng, H., Huang, J., 2024. Bearing-based formation with disturbance rejection. IEEE Transactions on Automatic Control, 69, 7793–7800.

Chu, W., Wuniri, Q., Du, X., Xiong, Q., Huang, T., Li, K., 2021. Cloud control system architectures, technologies and applications on intelligent and connected vehicles: A review. Chinese Journal of Mechanical Engineering, 34, 139.

Dong, J., Wang, J., Cai, M., Yang, Y., Xu, Q., Wang, J., et al., 2025. STFC: Spatio-temporal formation control for connected and autonomous vehicles in multi-lane traffic. Communications in Transportation Research, 5, 100219.

Dong, J., Yang, C., Cai, M., Chen, C., Xu, Q., Wang, J., et al., 2026. Multisource human-in-the-loop digital twin testbed for connected and autonomous vehicles in mixed traffic flow. Journal of Intelligent and Connected Vehicles, 9, 9210084.

Firoozi, R., Zhang, X., Borrelli, F., 2021. Formation and reconfiguration of tight multi-lane platoons. Control Engineering Practice, 108, 104714.

Fourati, S., Jaafar, W., Baccar, N., Alfattani, S., Langar, R., 2026. Foundation models for autonomous driving: A comprehensive survey. Engineering Applications of Artificial Intelligence, 176, 114805.

Gao, J., Li, S., Lyu, Y., Jiang, Y., Zhan, G., Yang, Y., et al., 2025. CIDC: A flexible formation control framework for intelligent connected vehicles. IEEE Transactions on Intelligent Vehicles, 10, 3072–3081.

Gao, L., Chu, D., Cao, Y., Lu, L., Wu, C., 2019. Multi-lane convoy control for autonomous vehicles based on distributed graph and potential field, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 2463–2469.

Han, L., Wang, Y., Yan, Z., Li, X., Ren, Z., 2024. Event-triggered formation control with obstacle avoidance for multi-agent systems applied to multi-UAV formation flying. Control Engineering Practice, 153, 106105.

Kamel, M.A., Yu, X., Zhang, Y., 2020. Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review. Annual reviews in control, 49, 128–144.

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y., 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, 22199–22213.

Li, J., Li, K., Chang, X., Gao, B., Xu, Q., Li, S.E., 2020. Principles and typical applications of cloud control system for intelligent and connected vehicles. Journal of Automotive Safety and Energy, 11, 261.

Li, Y., Liao, N., Zhang, G., Du, C., Zhao, S., Tian, Y., 2024. Unmanned vehicle formation control based on large language model, in: China Conference on Command and Control. Springer, pp. 329–339.

Li, Y., Tang, C., Li, K., He, X., Peeta, S., Wang, Y., 2018. Consensus-based cooperative control for multi-platoon under the connected vehicles environment. IEEE Transactions on Intelligent Transportation Systems, 20, 2220–2229.

Liu, W., Wang, X., Li, S., 2023. Formation control for leader–follower wheeled mobile robots based on embedded control technique. IEEE Transactions on Control Systems Technology, 31, 265–280.

Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., Qu, X., 2023. Can language models be used for real-world urban-delivery route optimization? The Innovation, 4.

Lu, L., He, Z., Chu, D., Wang, R., Peng, S., Zhou, P., 2025. ConvoyLLM: Dynamic multi-lane convoy control using LLMs, in: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 12659–12666.

Lykov, A., Karaf, S., Martynov, M., Serpiva, V., Fedoseev, A., Konenkov, M., et al., 2024. FlockGPT: Guiding UAV flocking with linguistic orchestration, in: 2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE, pp. 485–488.

Pueyo, P., Montijano, E., Murillo, A.C., Schwager, M., 2024. CLIPSwarm: Generating drone shows from text prompts with vision-language models, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 11917–11923.

Shi, Q., Meng, Q., Cheng, H., Xu, Q., Wang, J., 2025. Linguasim: Interactive multi-vehicle testing scenario generation via natural language instruction based on large language models, in: 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1036–1043.

Shi, X., Chen, S., Jang, K., Chen, T., 2025. Conflict-aware platoon scheduling at unsignalised intersections via traveling salesman problem: A large language model-guided heuristic solution, in: 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1718–1723.

Su, H., Yang, Z., Zhu, S., Chen, C., Guan, X., Xie, L., 2026. Bearing-based multi-agent formation control: A survey and taxonomy. Annual Reviews in Control, 61, 101043.

Tian, Y., Lin, F., Li, Y., Zhang, T., Zhang, Q., Fu, X., et al., 2025. UAVs meet LLMs: Overviews and perspectives towards agentic low-altitude mobility. Information Fusion, 122, 103158.

Wang, J., Zheng, Y., Chen, C., Xu, Q., Li, K., 2021. Leading cruise control in mixed traffic flow: System modeling, controllability, and string stability. IEEE Transactions on Intelligent Transportation Systems, 23, 12861–12876.

Wang, Z., Zhou, X., Xu, C., Gao, F., 2022. Geometrically constrained trajectory optimization for multicopters. IEEE Transactions on Robotics, 38, 3259–3278.

Wei, S., Wang, Q., Yang, K., 2026. Virtual traffic police: Large language model-augmented traffic signal control for

Just Accepted

unforeseen incidents. arXiv preprint arXiv:2601.15816.

Wu, Q., Wang, X., Qiu, X., 2024. Embedded technique-based formation control of multiple wheeled mobile robots with application to cooperative transportation. Control Engineering Practice, 150, 106002.

Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K.-Y.K., et al., 2024. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. IEEE Robotics and Automation Letters, 9, 8186–8193.

Zhang, X., Li, J., Zhou, J., Zhang, S., Wang, J., Yuan, Y., et al., 2025. Vehicle-to-everything communication in intelligent connected vehicles: A survey and taxonomy. Automotive Innovation, 8, 13–45.

Zhang, Y., Chen, J., Zhou, T., Duhan, T., Dong, J., Cao, Y., et al., 2026a. COIN: Collaborative interaction-aware multi-agent reinforcement learning for self-driving systems. arXiv preprint arXiv:2603.24931.

Zhang, Y., Li, P., Zhou, T., Fan, M., Sartoretti, G., 2026b. LATS: Large language model assisted teacher-student framework for multi-agent reinforcement learning in traffic signal control. arXiv preprint arXiv:2603.24361.

Author biography



Jianghong Dong received his B.E. degree and Ph.D. degree from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2020 and 2025, respectively. During his Ph.D. research, he was also a visiting Ph.D. student with the Multi-CAV Robotic Motion Laboratory at National University of Singapore from 2023 to 2024. He is currently a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, College of Design and Engineering, National University of Singapore. His research interests include connected and autonomous vehicles, multi-vehicle formation control, vehicle-road-cloud integration systems and digital twin.



Wenhan Cao received the Ph.D. degree from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2025, and the B.E. degree from the School of Electrical Engineering, Beijing Jiaotong University, Beijing, China, in 2019. He is currently an Eric and Wendy Schmidt AI in Science Postdoctoral Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. He was a finalist for the Best Student Paper Award at the 2021 IFAC MECC. His research interest is probabilistic physical AI.



Jiawei Wang received the B.E. degree in Automotive Engineering from Tsinghua University, Beijing, China, in 2018, and the Ph.D. degree in Mechanical Engineering from Tsinghua University in 2023. During his Ph.D. research, he was also a visiting Ph.D. student with the Automatic Control Laboratory at École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland from 2021 to 2022. He is currently a Postdoctoral Research Fellow with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor. Dr. Wang was a recipient of Outstanding Ph.D. Graduate, Excellent Doctoral Dissertation Award, and National Scholarship at Tsinghua University. He received the Distinguished Doctoral Dissertation Award from China Society of Automotive Engineers in 2023, and the Best Paper Award at the 18th COTA International Conference of Transportation Professionals in 2018. His research interests include connected vehicles, intelligent vehicles, distributed control and optimization, and traffic control.



Yifeng Zhang is currently a Research Fellow in the Department of Mechanical Engineering at the National University of Singapore (NUS), Singapore. He received the B.Eng. degree in Vehicle Engineering from Hunan University (HNU), Changsha, China, in 2020, the M.Sc. degree in Mechanical Engineering from NUS in 2021, and the Ph.D. degree in Mechanical Engineering from NUS in 2025. His research interests include intelligent transportation systems and multi-robot systems, with a focus on reinforcement learning and embodied intelligence.



Mengchi Cai received his B.E. degree and Ph.D. degree from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2018 and 2023, respectively. He was recipients of the National Scholarship and Comprehensive Excellent Scholarship in Tsinghua University. He has been honored the Shuimu Tsinghua Scholarship, and is currently a postdoctoral researcher at Intelligent and Connected Vehicles Lab, School of Vehicle and Mobility, Tsinghua University. His research interests include connected and automated vehicles, multi-vehicle formation control, and unsignalized intersection coordination.



Chaoyi Chen received the B.E. degree from Tsinghua University, Beijing, China, in 2016, the M.S. degree from Tsinghua University, Beijing, China and RWTH Aachen University, Aachen, Germany in 2019, and the Ph.D. degree from Tsinghua University, Beijing, China in 2023. He is currently a Postdoctoral Research Fellow

Just Accepted

with the School of Vehicle and Mobility, Tsinghua University. He was a recipient of the Shuimu Scholarship, Scholarship of Strategic Partnership RWTH Aachen University and Tsinghua University. His research interests include vehicular network, control theory and cooperative control.



Qing Xu received his B.S. and M.S. degrees in automotive engineering from Beihang University, Beijing, China, in 2006 and 2008, respectively, and the Ph.D. degree in automotive engineering from Beihang University in 2014. During his Ph.D. research, he worked as a Visiting Scholar with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign. From 2014 to 2016, he had his postdoctoral research in Tsinghua University. He is currently working as an Associate Research Professor with the School of Vehicle and Mobility, Tsinghua University. His main research interests include decision and control of intelligent vehicles.



Jianqiang Wang received the B.Tech. and M.S. degrees from Jilin University of Technology, Changchun, China, in 1994 and 1997, respectively, and the Ph.D. degree from Jilin University, Changchun, in 2002. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University, Beijing, China. He has authored over 150 papers and is a co-inventor of 99 patent applications. He was involved in over 10 sponsored projects. He was a recipient of the Best Paper Award in the 2014 IEEE Intelligent Vehicle Symposium, the Best Paper Award in the 14th ITS Asia Pacific Forum, the Best Paper Award in the 2017 IEEE Intelligent Vehicle Symposium, the Changjiang Scholar Program Professor in 2017, the Distinguished Young Scientists of NSF China in 2016, and the New Century Excellent Talents in 2008. His active research interests include intelligent vehicles, driving assistance systems, and driver behavior.



Keqiang Li received the B.Tech. degree from Tsinghua University of China, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from the Chongqing University of China, Chongqing, China, in 1988 and 1995, respectively. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. He is leading the national key project on ICVs (Intelligent and Connected Vehicles) in China. Dr. Li has authored more than 200 papers and is a co-inventor of over 80 patents in China and Japan. Dr. Li served as Member of the Chinese Academy of Engineering, Fellow Member of Society of Automotive Engineers of China, editorial boards of the International Journal of Vehicle Autonomous Systems, Chairperson of Expert Committee of the China Industrial Technology Innovation Strategic Alliance for ICVs (CAICV), and CTO of China ICV Research Institute Company Ltd. (CICV). He has been a recipient of Changjiang Scholar Program Professor, National Award for Technological Invention in China, etc. His main research areas include automotive control system, driver assistance system, and networked dynamics and control.



Guillaume Sartoretti joined the Mechanical Engineering department at the National University of Singapore as an Assistant Professor in 2019. Before that, he was a Postdoctoral Fellow in the Robotics Institute at Carnegie Mellon University. He received his Ph.D. degree in robotics from EPFL in 2016. He also holds a B.S. and an M.S. degree in Mathematics and Computer Science from the University of Geneva. He is interested in the emergence of collaboration/cooperation in large groups of intelligent agents making individual choices based on their local understanding of the world. Guillaume was a Manufacturing Futures Initiative (MFI) postdoctoral fellow at CMU in 2018–2019, was awarded an Amazon Research Award in 2022, as well as an Outstanding Early Career Award from NUS' College of Design and Engineering in 2023.

Just Accepted