



CoMAL: Collaborative Multi-Agent Large Language Models for Mixed-Autonomy Traffic

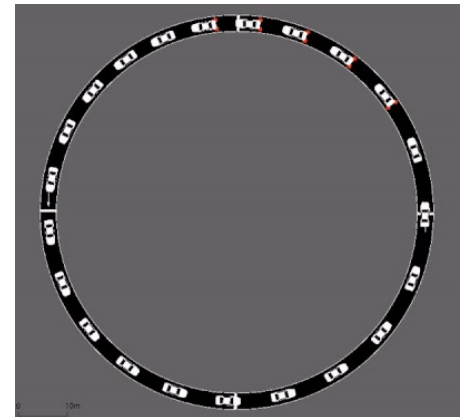
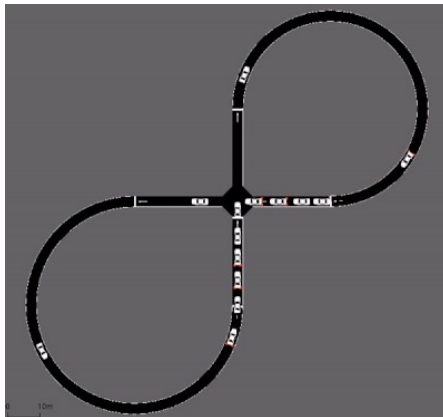
Huaiyuan Yao¹, Longchao Da¹, Vishnu Nandam¹, Justin Turnau¹,
Zhiwei Liu², Linsey Pang², Hua Wei¹

¹Arizona State University, ²Salesforce



Presented by:
John Tan Chong Min

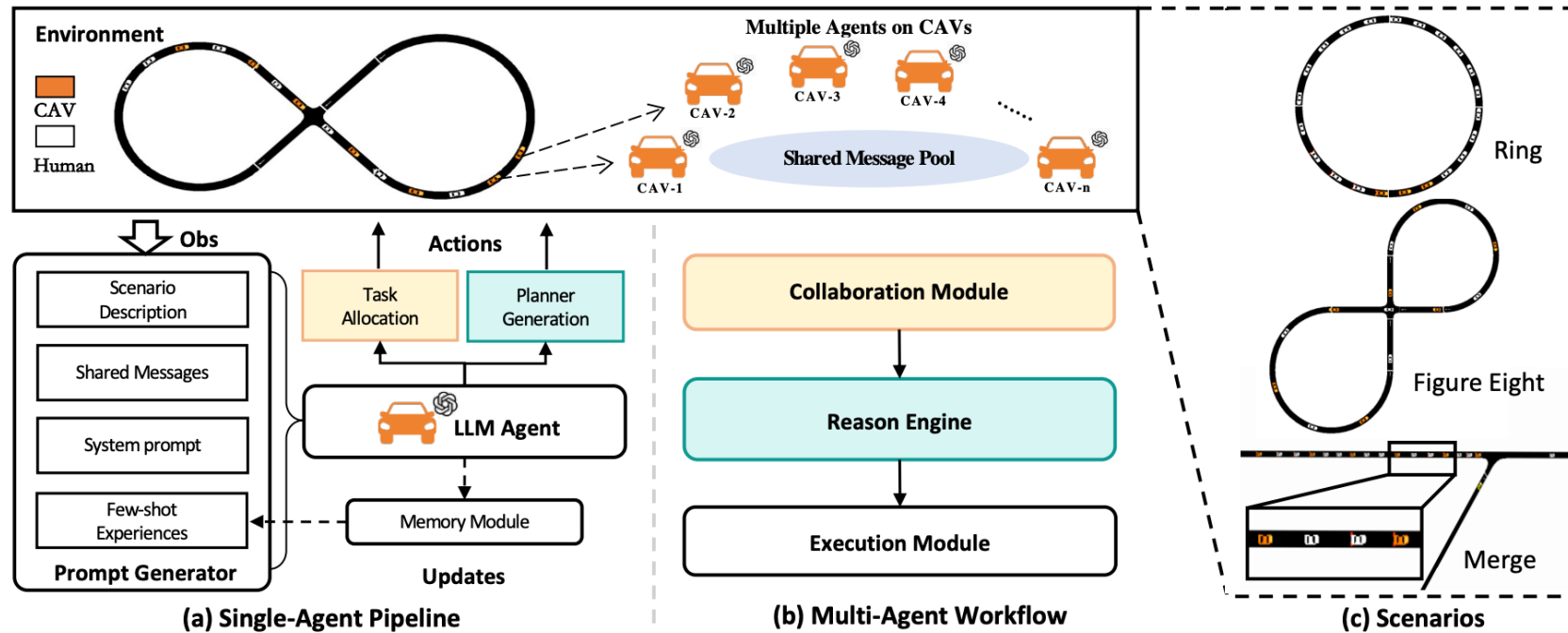
Environments



Key Points

- Deep RL performs well in specific and controlled environments but struggles to generalize across different scenarios due to its reliance on large amounts of specific training data
- Difficult to understand the decision-making process of RL, which limits its interpretability
- (Humans) Adolescents can learn to drive in just 20 hours and handle unfamiliar situations
 - Similar to Yann LeCun's point + Self-supervised Learning

Overall Framework of CoMAL



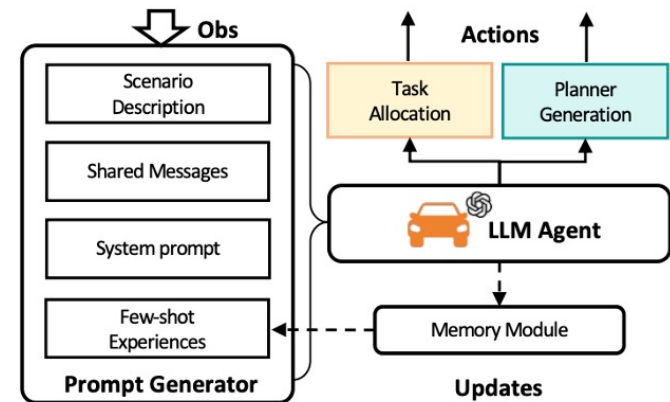
See prompts at: https://github.com/Hyan-Yao/CoMAL/blob/main/flow/controllers/llm_controller.py

Multi-Agent Collaboration: Same same but different

- Agents participate in brainstorming session in **Collaboration Module**
- Each agent develops driving plans according to its designated role through **Reason Engine**
 - Takeaway: Diversity is important
- **Reason Engine** generates **rule-based** driving planner grounded in **Intelligent Driver Model**
 - Takeaway: Rules are important, LLM is too ungrounded

Single-Agent Pipeline

- Encode scenario into textual description
- Add detailed system prompt
- Recall relevant driving experiences (few-shot prompting)
- Receive shared messages from other agents (learn from other agents)
- Get LLM response for task allocation / planning purposes



System Prompt

- Pros: Provides detailed description of the map / task
- Cons:
 - Uses text to encode visual information – LLMs may not understand text-based position that well
 - Uses LLM to encode numbers – LLMs may not have an intuitive understanding of numbers

Driving Task

You are the brain of an autonomous vehicle in the road. Your vehicle ID is **CAV_01**. You can connect all the autonomous vehicles in the scenario. Please make the decision to optimize the average velocity of all the vehicles. Try your best to avoid collisions with other objects.

Map description

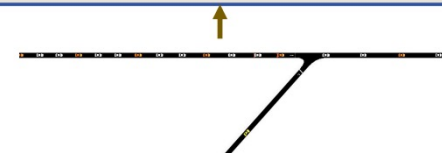
The merge map consists of two straight single lanes. One is the main highway where all vehicles drive, and another is the merge branch to intersect with main highway. The perturbations caused by the vehicles entering through the ramp leads to the formation of congested waves downstream in the main highway.

Motion state

Your speed is 3.80 m/s, and lane position is 11.75 m.

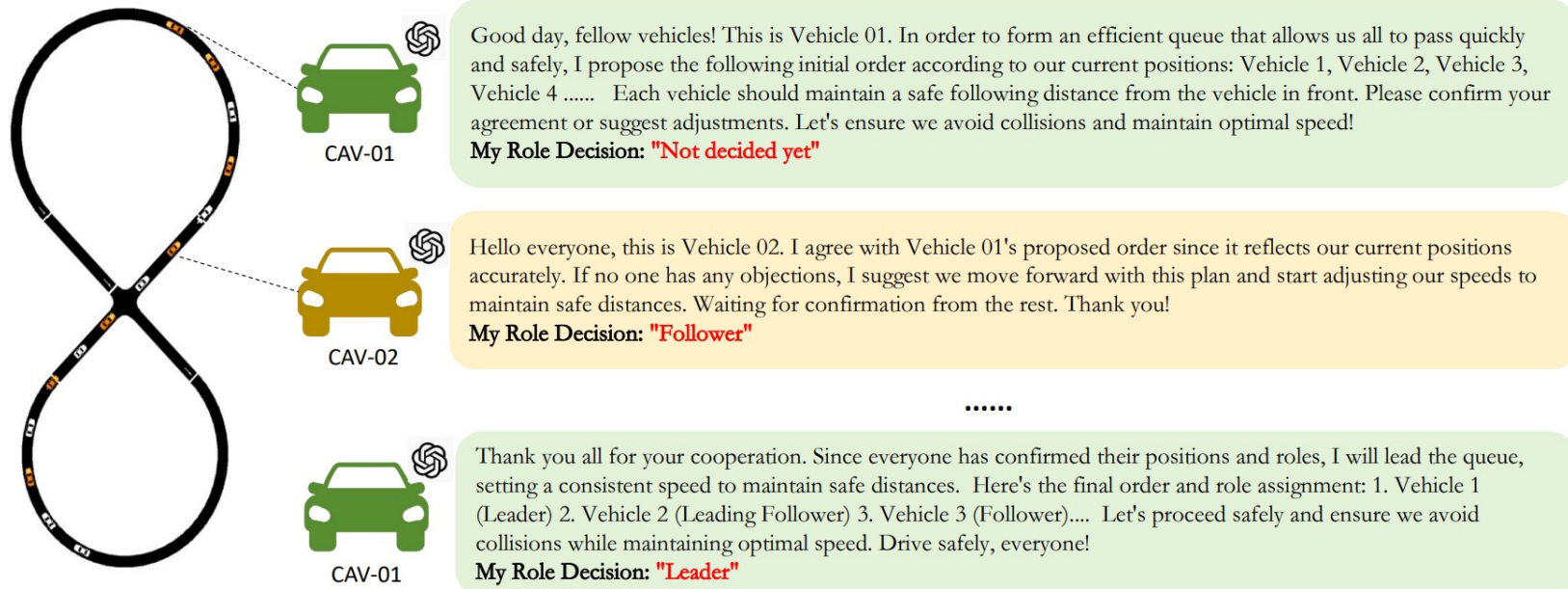
There are other vehicles driving around you, and below is their basic information:

- Vehicle CAV_02 is driving on the same lane as you. The speed of it is 4.51 m/s, and lane position is 15.64 m.



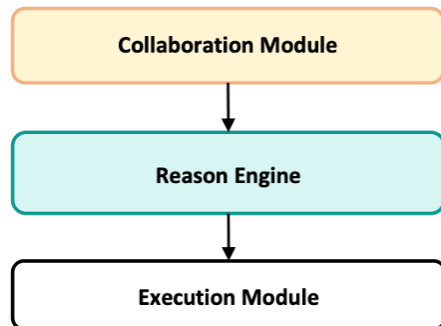
Collaboration between Agents

- Collaboration is done in AutoGen conversation style (AgentJo may be much faster and efficient)
- May be better if there is a clear hierarchy - one leader taking all inputs and making decision for all



Reasoning to Planning

- Reasoner is actually more akin to chain of thought prompting
- Maps to a rule-based planner for greater reliability



Collaboration Module

Thank you all for your cooperation. Since everyone has confirmed their positions and roles, I will lead the queue, setting a consistent speed to maintain safe distances. Let's proceed safely and ensure we avoid collisions while maintaining optimal speed. Drive safely, everyone!

My Role Decision: "Leader"

Reason Engine

Step 1. Role Clarification

After reviewing the positions the consensus shown by other vehicles, I am ready to **take on the role of the leader**.

Step 2. Scene Understanding

My position at 27.26m provides a good overview ahead and control over initiating movement through the intersection.

Step 3. Motion Instruction

I propose we start **moving slowly and steadily** to ensure all vehicles can follow safely.

Step 4. Planner Generation

Let's use the planner: **driving_planner(a_max = 0.2, v_target = 6.0, min_headway = 2.0)**

Memory

- Roles and sample IDM parameters are already given for each scenario
- Can be updated based on experiences

```
Memory = {
  "figureeight" : ["The best strategy is to help all vehicles form a queue to pass the intersection quickly, which avoids collisions there. Thus, driver age
    "Below are best IDM planners for different roles: Leader: SlowDown {'a' : 0.2, 'v' : 6.0, 's' : 2.0}, FollowUp {'a' : 1.0, 'v' : 30.0, 's' : 0.6}",
    "While reasoning your role, You can say not decided yet and wait for next round. There is no rush to give a final decision answer unless you are confident",
  "ring": ["If the difference between self headway and lead headway is relatively large (self headway > 1.5 * lead headway), you're in traffic jam. Otherwise
    "If you're in a traffic jam, your role should be Approacher. If not, your role should be Follower",
    "If you're in traffic jam, the best IDM planner for you is Approach {'a' : 1.0, 'v' : lead velocity + 1.0, 's' : 2.0}",
    "If you're not in traffic jam, the best IDM planner for you is FollowUp {'a' : 5.0, 'v' : 150.0, 's' : 0.0}",
    "While reasoning your role, it had better to reason your role directly in the round."],
  "merge": ["If the difference between self headway and lead headway is relatively large (self headway > 1.5 * lead headway), you're in traffic jam. Otherwise
    "If you're in a traffic jam, your role should be Approacher. If not, your role should be Follower",
    "If you're in traffic jam, the best IDM planner for you is Approach {'a' : 1.0, 'v' : lead velocity + 1.0, 's' : 2.0}",
    "If you're not in traffic jam, the best IDM planner for you is FollowUp {'a' : 5.0, 'v' : 150.0, 's' : 0.0}",
    "While reasoning your role, it had better to reason your role directly in the round."],
}
```

Intelligent Driver Model (Grounded in Math)

3.2.3 Execution Module We utilize the rule-based IDM model as a planner to execute driving strategies by adjusting its parameters. IDM is a car-following model to compute longitudinal dynamics. In this model, the acceleration a_k for vehicle k is defined by its bumper-to-bumper headway s_k (distance to preceding vehicle), velocity v_k , and relative velocity Δv_k , via the following equation:

$$(3.1) \quad a_k = \frac{dv_k}{dt} = a_{\max} \left[1 - \left(\frac{v_k}{v_0} \right)^\delta - \left(\frac{s^*(v_k, \Delta v_k)}{s_k} \right)^2 \right]$$

where s^* is the desired headway of the vehicle, denoted by:

$$(3.2) \quad s^*(v_k, \Delta v_k) = s_0 + \max(0, v_k T + \frac{v_k \Delta v_k}{2\sqrt{a_{\max} b}})$$

where $s_0, v_0, T, \delta, a_{\max}, b$ are given parameters. We set the desired time headway T , the comfortable braking deceleration b , and the acceleration exponent δ as constants while adjusting the desired velocity v_0 , the minimum spacing s_0 , and the maximum acceleration a_{\max} to tailor the driving planners. Thus Reason Engine generates a driving planner by customizing IDM's parameters (v_0, a_{\max}, s_0) .

Benchmarks

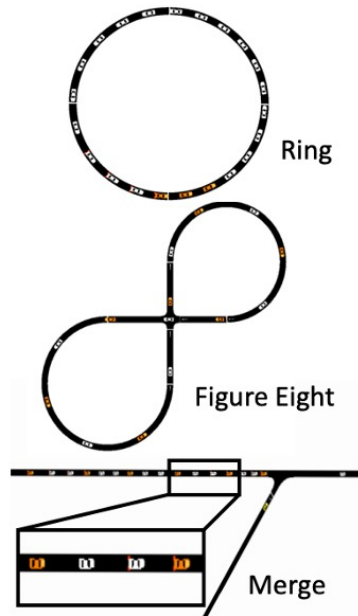


Table 1: Configurations of Benchmarks

| Scenario Name | Time(s) | Vehicles Distribution |
|---------------|---------|----------------------------|
| FE 0 | 150 | 13 humans, 1 CAV |
| FE 1 | 150 | 7 humans, 7 CAVs |
| FE 2 | 150 | 0 humans, 14 CAVs |
| Ring 0 | 150 | 21 humans, 1 CAV |
| Ring 1 | 150 | 19 humans, 3 CAVs |
| Ring 2 | 150 | 11 humans, 11 CAVs |
| Merge 0 | 75 | 10.0% CAV penetration rate |
| Merge 1 | 75 | 25.0% CAV penetration rate |
| Merge 2 | 75 | 33.3% CAV penetration rate |
| Merge 3 | 75 | 50.0% CAV penetration rate |
| Merge 4 | 75 | 90.0% CAV penetration rate |

CoMAL is more consistent

- Metrics:
 - Average vehicle speed in the network (m/s). Higher values indicate better overall traffic flow.
 - Standard deviation of vehicle speed (m/s). The smaller is more stable. Lower values reflect greater stability and consistency in traffic movement.

Table 2: Quantitative Evaluation of CoMAL on Flow Benchmarks

| Metric | Model | FE 0 | FE 1 | FE 2 | Ring 0 | Ring 1 | Ring 2 | Merge 0 | Merge 1 | Merge 2 | Merge 3 | Merge 4 |
|--------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Avg | Human Driver | 5.61 | 5.61 | 5.61 | 2.88 | 2.88 | 2.88 | 6.40 | 6.40 | 6.40 | 6.40 | 6.40 |
| | CoMAL | 6.40 | 6.47 | 6.29 | 2.86 | 2.85 | 2.87 | 6.59 | 7.40 | 7.42 | 7.86 | 8.83 |
| Std | Human Driver | 4.55 | 4.55 | 4.55 | 0.79 | 0.79 | 0.79 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 |
| | CoMAL | 1.74 | 1.77 | 2.24 | 0.29 | 0.26 | 0.31 | 2.88 | 2.91 | 2.61 | 2.47 | 2.70 |

Perception, Memory and Collaboration are all important

Table 3: Ablation Study

| No. | Perception | Memory | Collaboration | FE 1 | Merge 1 |
|-----|------------|--------|---------------|-------------|-------------|
| 1 | × | × | × | 5.61 | 6.40 |
| 2 | × | ✓ | ✓ | 5.81 | 6.51 |
| 3 | ✓ | × | ✓ | 5.17 | 6.72 |
| 4 | ✓ | ✓ | × | 5.18 | 6.88 |
| 5 | ✓ | ✓ | ✓ | 6.47 | 7.40 |

Agentic Approaches win RL in some instances

- IDM: Intelligent Driver Model (Just the planner done for each car without any collaboration)
- RL Frameworks: Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Evolutionary Strategies (ES), Augmented Random Search (ARS)
- CoMAL: IDM + Perception + Memory + Collaboration

Table 5: Comparison to RL Benchmark

| Model | FE 0 | FE 1 | FE 2 | Merge 0 | Merge 1 | Merge 2 |
|-------|-------------|-------------|-------------|--------------|--------------|--------------|
| IDM | 5.61 | 5.61 | 5.61 | 6.40 | 6.40 | 6.40 |
| ARS | 7.31 | 6.43 | 5.70 | 11.30 | 11.06 | 11.50 |
| ES | 6.87 | - | 5.96 | 13.31 | 17.29 | 17.36 |
| TRPO | 8.26 | 5.61 | 5.03 | 14.95 | 13.74 | 14.14 |
| PPO | 8.26 | 5.61 | 5.03 | 13.66 | 14.61 | 14.54 |
| CoMAL | 6.40 | 6.47 | 6.29 | 6.59 | 7.40 | 7.42 |

Questions to Ponder (1/2)

- Scenarios are relatively simple with toy examples, will this scale to the real world?
- Could there be a more efficient way of discussion rather than discussing with all agents? Localised discussion possible? How do human drivers discuss without talking?
- How can collaboration be done better without having “hanging” roles but fixed roles? Will a central planner be better?

Questions to Ponder (2/2)

- How can we improve the perception module? Is there a way to bring in multiple abstraction spaces for perception / memory?
- How can reasoning be done better? Do LLMs actually understand numbers well?
- Why does the LLM Agentic Framework perform poorer than RL frameworks for the Merge scenario? Would the LLM framework adapt fast in real time?