Fine-Tuning Vision-Language Models for Action Prediction in Autonomous Driving

Pei-Hsun Wu[1] **Chen-Fang Hu**[2] **Kai-Yuan Jeng**[3]
[1]National Yang Ming Chiao Tung University Graduate Degree Program of Robotics
[2]National Yang Ming Chiao Tung University Arête Honors Program
[3]National Tsing Hua University Department of Computer Science
{[1]wn2001.en13, [2]tracy.ls11}@nycu.edu.tw, [3]kaiyuanjeng@gapp.nthu.edu.tw

1 Project Overview

1.1 Profile

Track: 3. Application

Abstract and project goal: This project investigates fine-tuning Vision-Language Models (VLMs) with Reinforcement Learning (RL) to advance End-to-End Autonomous Driving (E2E-AD) systems. By leveraging RL, we aim to enable VLMs to process sequential driving frames and generate both actionable driving commands and coherent language-based justifications. This approach addresses existing challenges in safety, interpretability, and generalization, fostering reactive and explainable decision-making frameworks for autonomous driving.

The goal is to develop a fine-tuned VLM using RL for E2E-AD. The model will process short video sequences of traffic scenes and produce structured outputs (e.g., JSON) that include:

- Descriptions of traffic scenes.
- Driving actions (control commands).
- Justifications for the decisions made.

TL;DR ("Too Long; Didn't Read"): Fine-tune Vision-Language Models (VLMs) with Reinforcement Learning (RL) for interpretable End-to-End Autonomous Driving (E2E-AD).

1.2 Motivation

- **Why is the problem interesting?** Recent advances in VLMs open new possibilities for E2E-AD by enabling systems to process raw visual inputs and directly generate both driving commands and human-understandable justifications [Wang et al., 2024, Xu et al., 2024, Jia et al., 2025]. Unlike traditional RL methods that focus solely on low-level control, pretrained VLMs naturally capture rich language-vision representations, paving the way for interpretable and reactive decision-making in safety-critical scenarios.
- **Critical challenges.** Adapting VLMs for E2E-AD introduces several core challenges:
  - **Challenge 1.** Most VLMs rely on text prompts to specify task details, which is impractical for real-time AD systems that must operate solely on visual inputs. Achieving direct perception-to-action mapping requires novel architectures or reward designs.
  - **Challenge 2.** Pretrained VLMs are designed for descriptive language-vision alignment tasks, whereas E2E-AD requires grounded, actionable decisions with aligned explanations and without redundant textual contexts.

  - **Challenge 3.** Generating structured outputs (e.g., JSON format) directly from visual inputs without post-processing remains an open challenge, as current methods primarily focus on plain text generation.

- **Justify why the problem remains open or unsolved.** Despite recent advances in integrating RL with VLMs in domains such as robotics and visual navigation, their application to E2E-AD faces significant limitations:

  - **VLM-RL [Huang et al., 2024]:** This method employs Contrastive Language Goals (CLG) to derive semantic rewards, enabling improved vision-language alignment (partially addressing Challenge 2). However, it lacks the granularity needed for precise control and does not support actionable outputs required in AD scenarios.
  - **AlphaDrive [Jiang et al., 2025]:** This framework achieves strong results in multimodal planning using a two-stage RL strategy (SFT + GRPO [Shao et al., 2024]) with tailored reward terms like action-weighted rewards and format consistency (partially addressing Challenge 1). However, it relies heavily on large-scale preference-labeled datasets, limiting scalability, and does not natively address structured output generation (Challenge 3).
  - **XAI-based Transformer Model** [Dong et al., 2023]: Demonstrates a Swin Transformer-based model [Liu et al., 2021] that improves explainability by generating attention-aligned natural language rationales for driving scenes. It achieves strong results on BDD Object Induced Actions (BDD-OIA) [Xu et al., 2020] for explanation quality (Challenge 2). However, it does not resolve Challenge 1 (no real-time visual-to-action mapping) or Challenge 3 (outputs remain unstructured).
  - **Object-Centric Driving Policy** [Wang et al., 2019]: Proposes a sparse attention model over object features to generate actions from visual input. The model improves robustness in low-data and urban driving settings, directly contributing to Challenge 1. Its attention maps also support interpretability (Challenge 2, partial), but it does not output explanations or structured representations (Challenge 3).

  In summary, existing methods either emphasize interpretability through vision-language alignment [Huang et al., 2024, Dong et al., 2023], or focus on action grounding with object-centric perception [Wang et al., 2019], while others depend on costly human feedback and heuristics [Jiang et al., 2025]. However, none of those fully address the challenge of generating interpretable and actionable structured outputs (e.g., JSON) directly from visual inputs. (Challenge 1 and Challenge 3)

- **State-of-the-art methods.** Recent frameworks for fine-tuning VLMs via RL in E2E-AD have achieved notable results:

  - **VLM-RL [Huang et al., 2024]:** Demonstrates significant improvements in semantic alignment metrics through hierarchical reward synthesis, achieving a 15% improvement in safety-related metrics in CARLA simulations.
  - **AlphaDrive [Jiang et al., 2025]:** Sets a benchmark for multimodal planning by achieving human-aligned reasoning abilities through tailored reward terms, with a reported 20% increase in planning accuracy compared to baseline methods.
  - **XAI-based Transformer Model [Dong et al., 2023]:** Outperforms most baselines on BDD-OIA [Xu et al., 2020] by combining Swin Transformer [Liu et al., 2021] and a Transformer decoder [Vaswani et al., 2017], with an F1 score of 0.823 for reasons and 0.913 for actions.

2 Problem Formulation

As described earlier (Section 1), our goal is to fine-tune a pretrained VLM for E2E-AD. Specifically, we aim to enable the model to generate the corresponding action along with explicit justifications in a structured language format (e.g., JSON). This task can be viewed as a video understanding problem—an extension of image captioning—with two additional challenges: 1) processing a sequence of images (i.e., continuous frames) to capture temporal relationships within the traffic scene and 2) generating more detailed and aligned descriptions of the given scene.

To address the critical challenges we identified in adapting VLMs for E2E-AD (Section 1.2), we propose a three-stage framework for fine-tuning our target VLM. Each stage is designed to tackle a specific challenge:

- **Stage 1.** The first stage focuses on adapting the VLM to directly process video inputs and generate descriptive outputs without relying on text prompts.
- **Stage 2.** The second stage enables the model to reason over video descriptions and generate actionable decisions.
- **Stage 3.** The final stage fine-tunes the VLM to produce structured outputs (e.g., JSON format) that encapsulate descriptions, decisions, and reasoning explanations without post-processing.

In a decision-making process, the agent first observes a state s, then takes an action a, and finally transitions to a new state s'. Following the approach in [Ren et al., 2017], we reformulate video understanding as a decision-making process, where the agent comprises two components: 1) a policy network (VLM in our case) responsible for decision-making and 2) a value network that evaluates the state-value. Note that sequential language modeling can be formulated as a Markov decision process (MDP) [Ranzato et al., 2016], where the state typically represents the sequence of tokens generated so far, and the action corresponds to selecting the next token. The key difference in our setting is that VLM's auto-regressive generation of tokens commences only after all image frames have been fully incorporated.

For the given image frames, our objective is to generate structured text information that includes: 1) a detailed and aligned description of the observed environment (images), 2) an optimal action for AD control, and 3) a justification or reasoning supporting the chosen action.

2.1 State and Action Spaces

Let $\mathbf{I} = \{I_1, I_2, \ldots, I_M\}$ denote a sequence of images comprising a predefined number M of frames (where M may also be parameterized as M_ϕ, allowing the model to learn an optimal ϕ to better capture the scene's dynamics). The text information we aim to obtain from $\mathbf{I}$ is expressed as a series of language tokens $\mathbf{T} = (a_1, a_2, \ldots, a_N)$ in the language modeling framework, where N is the total number of tokens generated.

At time step n, the agent has observed the image frames $\mathbf{I}$ together with the text sequence $\mathbf{T}_t = (a_1, a_2, \ldots, a_t)$ generated up to time step t, where $1 \leq t \leq N$. The state is thus defined as $s_t = (\mathbf{I}, \mathbf{T}_t)$. Based on s_t, the agent predicts the next token a_{t+1} as its action, which results in an updated state consisting of the image frames $\mathbf{I}$ and the extended token sequence $\mathbf{T}_{t+1}$. Thus, we define: 1) the state space as $\mathcal{S} = \{s_t = (\mathbf{I}, \mathbf{T}_t) \mid 0 \leq t \leq N\}$ (where $\mathbf{T}_0$ is an empty sequence), and 2) the action space $\mathcal{A}$ as the vocabulary $\mathcal{Y}$, from which tokens are drawn, and also the structured output tokens that correspond to JSON elements, enabling the model to generate interpretable and actionable outputs directly. At each step t, the action taken is $a_{t+1} \in \mathcal{A}$.

2.2 Policy and Value Networks

We adopt an Actor-Critic framework [Konda and Tsitsiklis, 1999] common in RL for sequence generation.

2.2.1 Actor Policy Network

The actor policy network π_θ is directly implemented by the pretrained VLM we aim to fine-tune, parameterized by θ. Its role is to make sequential decisions by generating the structured text output token by token. At each time step t, given the current state $s_t = (\mathbf{I}, \mathbf{T}_t)$, the policy network outputs a probability distribution $\pi_\theta(a_{t+1} \mid s_t) = P(a_{t+1} \mid s_t; \theta)$ over the vocabulary $\mathcal{Y}$ for the next token a_{t+1}, which is then sampled via $a_{t+1} \sim \pi_\theta(\cdot \mid s_t)$.

The parameters θ (potentially all parameters of the VLM) are updated to maximize the expected cumulative reward obtained from the generated sequence. The reward signal typically originates from an independently trained Reward Model (discussed later) evaluating the quality of the complete sequence $\mathbf{T}$.

2.2.2 Critic Value Network

The critic value network V_w is responsible for estimating the state-value function $V^{\pi_\theta}(s_t)$, defined as the expected cumulative (discounted) reward achievable starting from state s_t and subsequently following the policy π_θ:

$$V^{\pi_\theta}(s_t) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{N-t-1} \gamma^k r_{t+k+1} \mid s_t\right],$$

where $\gamma \in [0, 1]$ is the discount factor, and r_{t+k+1} is the future rewards derived from the final sequence reward $\mathcal{R}(\mathbf{T})$ provided by the reward model (Section 2.3).

To approximate this true state-value function, we employ a value network V_w, parameterized by w. The value network V_w shares the core VLM body with the policy network π_θ to leverage the same feature representations. A separate Multi-Layer Perceptron (MLP) head, denoted as $\mathbf{MLP}_w^{head}$, is attached on top of the hidden state representation $h(s_t)$ extracted from the shared VLM body. This MLP head outputs the estimated state value as

$$V_w(s_t) = \mathbf{MLP}_w^{head}(h(s_t)) \approx V^{\pi_\theta}(s_t).$$

The parameters w of the value network (primarily those of the $\mathbf{MLP}_w^{head}$) are updated to minimize the prediction error between the estimated value $V_w(s_t)$ and a target value.

2.3 Reward Signal Modeling

The reward signal $\mathcal{R}$ is designed to evaluate the quality of outputs at each stage (Section 2). The **Semantic Alignment Reward** $\mathcal{R}^1$ assesses the semantic alignment between video inputs and descriptive outputs. The **Action-Reasoning Reward** $\mathcal{R}^2$ evaluates the accuracy and interpretability of actionable decisions based on reasoning. The **Structure-Formatting Reward** $\mathcal{R}^3$ measures the consistency, completeness, and format correctness of structured outputs.

2.3.1 Semantic Alignment Reward

To adapt the pretrained VLM [Face, 2024, Marafioti et al., 2025] for autonomous driving scenarios in Stage 1, we propose a semantic alignment reward based on CLIP [Radford et al., 2021], following the method in [Ren et al., 2017]. Given a sequence of images $\mathbf{I} = \{I_1, I_2, \ldots, I_M\}$, we first extract frame-level image features using CLIP's visual encoder. These features are aggregated via average pooling:

$$\mathbf{v} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{CLIP}^{image}(I_i),$$

where $\mathbf{CLIP}^{image}(I_i)$ is the normalized embedding of the i-th frame. Similarly, the generated description $\mathbf{T}$ is encoded by CLIP's text encoder to obtain the text embedding $\mathbf{t} = \mathbf{CLIP}^{text}(\mathbf{T})$.

The Stage 1 reward is then computed as the cosine similarity between the aggregated video embedding $\mathbf{v}$ and the text embedding $\mathbf{t}$:

$$\mathcal{R}^1(\mathbf{I}, \mathbf{T}) = \mathbf{v} \cdot \mathbf{t},$$

where both embeddings are L2-normalized by the CLIP encoders, simplifying the cosine similarity to a dot product. This reward encourages the model to generate descriptions that are semantically aligned with the video content while eliminating the need for text prompts.

2.3.2 Action-Reasoning Reward

For Stage 2, our goal is to enhance the VLM's capability to generate not only scene descriptions but also actionable AD decisions accompanied by coherent justifications. We plan to explore reward mechanisms inspired by recent advances in Chain-of-Thought (CoT) reasoning [Wei et al., 2022, DeepSeek-AI et al., 2025]. However, defining a precise reward function $\mathcal{R}^2$ that effectively balances action correctness with the quality and logical consistency of the reasoning requires further investigation. The specific formulation of this reward is therefore deferred to a later phase of the project.

2.3.3 Structure-Formatting Reward

The objective for Stage 3 is to train the VLM to directly output all relevant information (description, action, reasoning) in a predefined structured format, such as JSON, eliminating the need for post-processing. Designing an appropriate reward function $\mathcal{R}^3$ to evaluate format correctness, completeness, and consistency with the generated content necessitates dedicated research. The specific formulation of this reward is therefore deferred to a later phase of the project.

2.4 Training Objective

2.4.1 Semantic Alignment Optimization

The training objective for Stage 1 involves optimizing two sets of parameters:

- θ: Parameters of the Actor policy network π_θ.
- w: Parameters of the Critic value network V_w.

Actor Objective. The Actor maximizes the expected cumulative reward:

$$\mathcal{J}^1(\theta) = \mathbb{E}_{\mathbf{T}\sim\pi_\theta}\left[\mathcal{R}^1(\mathbf{I}, \mathbf{T})\right],$$

where $\mathcal{R}^1$ is the CLIP-based semantic alignment reward defined in Section 2.3.1.

Critic Objective. The Critic minimizes the prediction error of future rewards:

$$\mathcal{L}^1(w) = \mathbb{E}_{\mathbf{T}\sim\pi_\theta}\left[\sum_{t=0}^{N-1}\left(V_w(s_t) - \sum_{k=t}^{N-1}\gamma^{k-t} r_{k+1}\right)^2\right],$$

where $\gamma \in [0, 1]$ is the discount factor, and $r_{k+1} = \mathcal{R}^1(\mathbf{I}, \mathbf{T}_{k+1}) - \mathcal{R}^1(\mathbf{I}, \mathbf{T}_k)$ represents the incremental reward at step k.

2.4.2 Action-Reasoning Optimization

The training objective for Stage 2 involves extending the VLM's capabilities to generate actionable decisions with coherent reasoning based on scene descriptions. While the general framework follows the Actor-Critic paradigm, the specific design of the reward function $\mathcal{R}^2$ (Section 2.3.2) and the corresponding optimization dynamics require further exploration. Key challenges include:

- Balancing action correctness with reasoning quality, ensuring logical consistency between the two components.
- Incorporating temporal dependencies in reasoning steps to better reflect real-world traffic scenarios.

2.4.3 Structure-Formatting Optimization

The training objective for Stage 3 focuses on formatting the VLM to directly output structured information (e.g., JSON) without relying on post-processing. Achieving this requires designing a reward function $\mathcal{R}^3$ (Section 2.3.3) that evaluates format correctness, completeness, and semantic consistency. However, several open questions remain:

- How to balance structural rigidity (e.g., JSON syntax correctness) with generative flexibility, avoiding over-constrained outputs.
- How to penalize missing or incomplete fields while encouraging semantic alignment with scene descriptions and actions.

3 Empirical Evaluation

3.1 Performance Metrics

- **Accuracy**: The proportion of correctly predicted instances (both positive and negative) to the total number of predictions. It reflects the overall correctness of the model's output.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives, respectively, as defined by the confusion matrix.

- **F1-score** (follow [Xu et al., 2020]): A harmonic mean of precision and recall that accounts for class imbalance in action predictions (Fig. 2).
  - **Macro-F1 (mF1)**: The arithmetic mean of the F1-scores computed independently for each action class. This metric treats all classes equally, regardless of their frequency.

$$\text{mF1} = \frac{1}{C} \sum_{i=1}^{C} \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}$$

where P_i and R_i denote the precision and recall for class i, and C is the total number of action classes. $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$.

  - **Overall F1 ($\mathbf{F1_{all}}$)**: The F1-score computed by globally aggregating true positives, false positives, and false negatives across all samples $|S|$. This metric captures the model's performance in multi-label scenarios where each sample may be associated with multiple actions or explanations.

$$\text{F1}_{all} = \frac{1}{|S|} \sum_{j=1}^{|S|} \text{F1}(\hat{S}_j, S_j)$$

where S_j is the ground-truth label set and $\hat{S}_j$ is the prediction label set for sample j.

3.2 Baseline Methods

Given that our ultimate objective is to fine-tune a VLM via RL, we compare our approach against representative baselines that either adopt convolutional neural network (**CNN**) architectures [Lecun et al., 1998] or rely on supervised fine-tuning of VLMs:

- Baseline 1 [Xu et al., 2020]: The proposed method adopts a dual-branch Faster R-CNN framework that integrates global scene context with object-level reasoning via a selector module, enabling joint prediction of driving actions and their explanations to enhance interpretability in autonomous decision-making.
- Baseline 2 [Dong et al., 2023]: The proposed model reformulates the AD decision-making task from a traditional classification problem to an image captioning task guided by language-induced visual attention, so that a fully Transformer-based architecture can jointly generate interpretable text descriptions and driving actions.
- Baseline 3 (Ours): Our proposed approach leverages supervised fine-tuning of the SmolVLM-Instruct model [Face, 2024] for multi-modal action and explanation generation.

  Training Configuration (at present):
  - training epoch = 6
  - batch size = 4
  - initial learning rate = 1e-4
  - learning rate schedule = True
  - gradient accumulation steps = 4
  - gradient checkpointing = True

  The example results (at present) in Fig. 1.

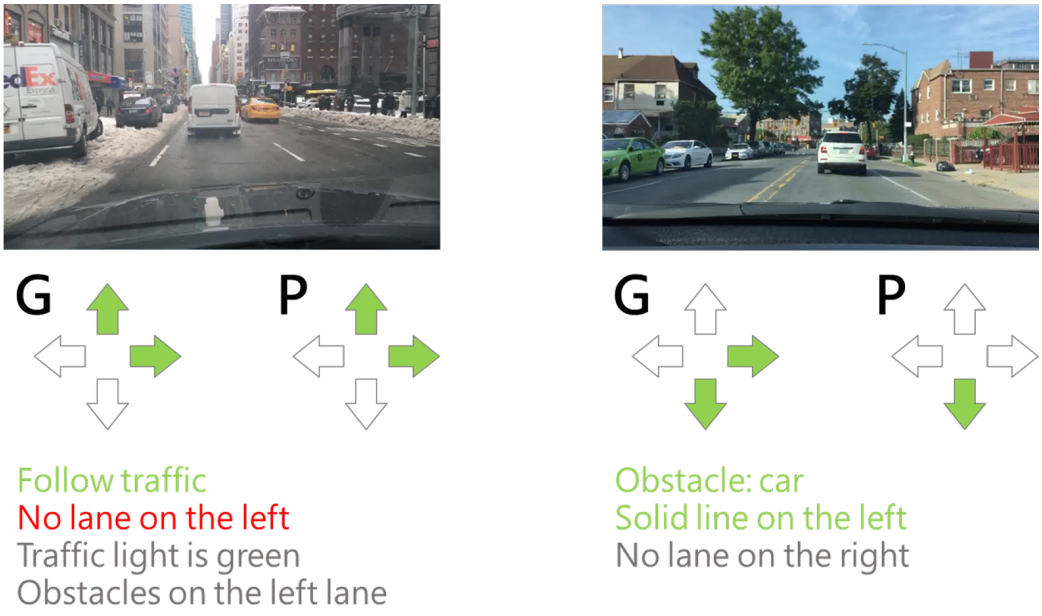

Figure 1: Example results at present. Figure Legend: **G** denotes the ground-truth and **P** indicates the model prediction. The arrows follow a clockwise order representing the actions: move forward, stop/slow down, turn left, and turn right. Green arrows denote the set of driving actions deemed executable by the model. Explanation texts are color-coded as follows: green for true positives, red for false positives, and gray for false negatives.

3.3 Benchmark Tasks or Datasets

We use the BDD-OIA dataset [Xu et al., 2020], which contains object-annotated driving scenes from the BDD100K [Yu et al., 2020] dataset enriched with action decisions and ground-truth object-level influence annotations. Specifically, the BDD-OIA dataset selects complex traffic scenes from BDD100K, defined as those containing more than five pedestrians or more than five vehicles, to emphasize challenging urban decision-making scenarios. In addition to providing labels for four distinct driving actions (e.g., move forward, stop/slow down, turn left, turn right), the dataset also includes 21 fine-grained, human-interpretable explanations corresponding to those actions. This joint annotation of actions and their underlying causal factors facilitates the development and evaluation of autonomous driving models with improved transparency and decision-level interpretability. As such, the dataset is well suited to evaluate both the accuracy of driving decisions and the explainability of autonomous vehicle behavior (Fig. 2).

4 Methodology

We propose a three-stage fine-tuning framework for VLMs, designed to address the three critical challenges identified earlier 1.2. In the first stage, we focus on adapting the model architecture to directly process video inputs and generate descriptive outputs, eliminating reliance on text prompts and enabling direct perception-to-description mapping. The second stage extends the model's capabilities by integrating reasoning mechanisms to generate actionable decisions based on the video descriptions, ensuring alignment between observations and decisions. Finally, in the third stage, we aim to fine-tune the model to produce structured outputs (e.g., JSON format) that encapsulate video descriptions, actionable decisions, and reasoning explanations, eliminating the need for post-processing plain text outputs.

Action Category	Number	Explanations	Number
Move forward	12491	Traffic light is green	7805
		Follow traffic	3489
		Road is clear	4838
Stop/Slow down	10432	Traffic light	5381
		Traffic sign	1539
		Obstacle: car	233
		Obstacle: person	163
		Obstacle: rider	5255
		Obstacle: others	455
Turn left	838	No lane on the left	150
		Obstacles on the left lane	666
		Solid line on the left	316
	5064	On the left-turn lane	154
		Traffic light allows	885
		Front car turning left	365
Turn right	1071	No lane on the right	4503
		Obstacles on the right lane	4514
		Solid line on the right	3660
	5470	On the right-turn lane	6081
		Traffic light allows	4022
		Front car turning right	2161

Figure 2: Actions and explanations in BDD-OIA [Xu et al., 2020]

5 Experimental Results

5.1 Example Construction

To validate the effectiveness of our proposed approach, we conducted a toy experiment on a simplified subset of the BDD-OIA dataset. In the original dataset, each input video is paired with multiple candidate $\langle \text{action}, \text{justification} \rangle$ pairs, and the model must select the correct one [Xu et al., 2020]. For clarity, we extract a subset in which each action is associated with exactly one justification. This one-to-one mapping between actions and justifications simplifies the classification task and allows us to more directly assess model performance.

In the experiment, we take the annotation content with only one action and one reason as the toy example. Due to this operation, we still have about 2,000 data points left, which are split into training and test sets in a ratio of 8:2. However, the training time required for full parameter fine-tuning is too long, so we used 1/8 of the training data as a small train dataset. All of the above splits are based on the action distribution of the original toy example.

5.2 Supervised Fine-tuning

As a supervised baseline, we fine-tune our vision–language model (VLM) on the toy dataset. Each example is a pair

$$\langle \mathbf{I},\, \langle \text{action, justification} \rangle \rangle \;\longrightarrow\; \langle \mathbf{I},\, \mathbf{T} \rangle,$$

where $\mathbf{I}$ is the input video or a batch of image frames, and $\mathbf{T}$ is the concatenation text of the action and justification (cf. Section 2). We optimize the token-wise cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = \text{CrossEntropy}\big(\hat{\mathbf{T}},\, \mathbf{T}\big),$$

where $\hat{\mathbf{T}} = \mathbf{VLM}(\mathbf{I})$ is the model prediction. For example (Fig. 2), an incorrect prediction

$$\hat{\mathbf{T}} = \big\langle \text{Action: Move forward; Reason: Traffic light is red} \big\rangle$$

instead of the ground truth

$$\mathbf{T} = \big\langle \text{Action: Move forward; Reason: Traffic light is green} \big\rangle$$

yields a positive cross-entropy loss.

5.3 Reinforcement Fine-tuning

Next, we evaluate RL-based fine-tuning in the E2E-AD scenario, where the VLM must generate an $\langle \text{action}, \text{justification} \rangle$ pair given only the images. We discard the ground-truth label $\mathbf{T}$ and use the two reward functions defined in Section 2.3.

5.3.1 Semantic Alignment Reward

Semantic Alignment Reward $\mathcal{R}^1$ is the cosine similarity between CLIP embeddings of the image $\mathbf{I}$ and the generated justification (see Section 2.3.1). For the pre-trained CLIP model, we use `openai/clip-vit-base-patch32`.

5.3.2 Action-Reasoning Reward

Action-Reasoning Reward $\mathcal{R}^2$ is a score from an LLM-as-judge, which evaluates the logical consistency between the generated action and justification. To evaluate the action-reason reward, we use the BLEU-4 score as the evaluation metric, compare the ground truth and the predicted output, check the accuracy of n-grams (whether a combination of n consecutive words appears, n is from 1 to 4), and use a threshold of 0.5 to distinguish between 0 and 1 as the action-reason reward.

Explanation: {explanation}
Reason: {reason}
Score:

5.3.3 Overall Fine-Tuning Loop

The combined reward is

$$\mathcal{R} = \alpha\, \mathcal{R}^1 + \beta\, \mathcal{R}^2$$

and we optimize the VLM parameters via policy gradient.

The fine-tuning loop proceeds as follows:

1. For each image frames $\mathbf{I}$, sample

   $$\langle \text{action}, \text{justification} \rangle \sim \pi_\theta(\cdot \mid \mathbf{I}),$$

   that is,

   $$\langle \text{action}, \text{justification} \rangle = \hat{\mathbf{T}} = \mathbf{VLM}(\mathbf{I}).$$

2. Compute $\mathcal{R}^1$ by embedding $\mathbf{I}$ and the justification with CLIP.
3. Compute $\mathcal{R}^2$ by prompting the LLM judge with the sampled action and justification.
4. Form the total reward $\mathcal{R} = \alpha\, \mathcal{R}^1 + \beta\, \mathcal{R}^2$ and update θ via policy gradient.
5. Repeat for subsequent episodes.

And finally, we use the **F1-score** in BDD-OIA [Xu et al., 2020] to compare the two approaches.

5.4 Evaluation of Baseline Methods

Based on the above experiments on the toy example, we present the following preliminary results.

Reward Signal Our RL approach is guided by two reward signals: *similarity* and *correctness* scores. After 3 epochs of fine-tuning, the SFT model achieves a similarity of similarity $= 0.2576$ and a correctness score of correctness $= 0.1166$, while our RL model achieves the same similarity score (similarity $= 0.2576$) but a slightly lower correctness score of correctness $= 0.09935$. We hypothesize that this may be attributed to the fact that CLIP, as a general-purpose vision-language encoder, lacks task-specific fine-tuning, which potentially limits its ability to accurately align images and text in the context of driving scene understanding.

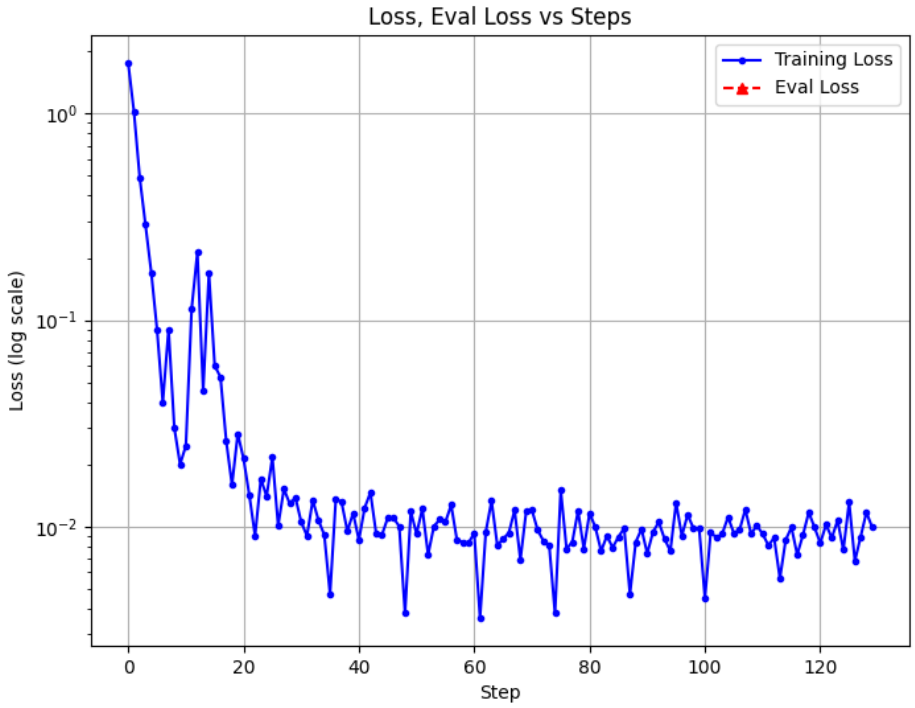

Figure 3: Training Loss of 10 epoch in toy example. In the following experiment we chose checkpoint-120 as the model.

As shown in Figure 5, the overall F1 score decreases after A2C fine-tuning. However, from the distribution shown in Figure 4, we observe that the output distribution after A2C training is closer to the ground truth distribution. Moreover, the F1 score for the *action* component improves.

We hypothesize that this discrepancy is due to the limited size of the training data, which may lead to generated outputs that do not belong to the predefined 4 actions or 21 justifications. Additionally, our F1 score calculation compares the entire pair of action and justification, such as `("Follow traffic", "Road is clear")`. As a result, even if the semantic meaning is correct, any variation in wording will lead to a mismatch and the pair will not be counted as correct in the F1 score. This also explains why the total sample count in the evaluation table does not always match the ground truth.

RL Training Loss The training dynamics are plotted in Fig. 3.

RL Agent's Action Distribution The actions and reasons' distribution is shown in Fig. 4.

F1 Score Comparison The F1 score comparison between SFT and RL approaches is shown in Fig. 5.

Future Work While these results are currently limited to a simplified setting for a toy example, we already obtain an improvement with only the unlabeled data. Thus, we expect an overall improvement when training on the full dataset.

6 Expected Contributions of Each Team Member

- Pei-Hsun Wu: Responsible for empirical evaluation, including dataset selection, experimental setup, and parameter tuning for model fine-tuning.

Action & Reason Distribution

Test GT	Forward			Stop						Left						Right					
Action	41			339						12						11					
Reason	14	20	8	73	23	6	0	208	26	1	8	2	0	1	0	1	7	2	1	2	0

Test pred. SFT	Forward			Stop						Left						Right					
Action	66			201						19						23					
Reason	16	10	8	15	17	10	0	82	28	0	0	1	0	0	0	0	0	0	0	2	0

Test pred. A2C	Forward			Stop						Left						Right					
Action	45			203						15						27					
Reason	15	7	8	9	17	10	0	75	36	0	0	2	0	1	0	1	0	0	0	0	0

Figure 4: Actions and explanations distribution in toy example test dataset. The number in the form means the number of samples.

Action & Reason F1 score

Test SFT F1	Forward			Stop						Left						Right					
Action (%)	9			63						0						0					
Reason(%)	7	0	0	5	15	0	0	34	7	0	0	0	0	0	0	0	0	0	0	0	0

Test A2C F1	Forward			Stop						Left						Right					
Action (%)	14			62						0						0					
Reason(%)	0	0	0	5	10	0	0	27	16	0	0	0	0	0	0	0	0	0	0	0	0

Model	F1_act (sample)	F1_rsn (sample)
Test SFT	43.42 %	14.14 %
Test A2C	42.92 %	11.66 %

Figure 5: Actions and explanations F1 score in toy example test dataset. The number in the form means the F1 score percentage of each action or reason.

- Chen-Fang Hu: Focuses on result presentation, including data visualization, poster design, and delivering the final presentation.
- Kai-Yuan Jeng: Leads the problem formulation and methodology design, providing theoretical foundations and ensuring alignment with project goals.

References

Pengqin Wang, Meixin Zhu, Xinhu Zheng, Hongliang Lu, Hui Zhong, Xianda Chen, Shaojie Shen, Xuesong Wang, Yinhai Wang, and Fei-Yue Wang. Bevgpt: Generative pre-trained foundation model for autonomous driving prediction, decision-making, and planning. *IEEE Transactions on Intelligent Vehicles*, pages 1–13, 2024. doi: 10.1109/TIV.2024.3449278.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024. doi: 10.1109/LRA.2024.3440097.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M42KR4W9P5.

Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving, 2024. URL https://arxiv.org/abs/2412.15544.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning, 2025. URL https://arxiv.org/abs/2503.07608.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Jiqian Dong, Sikai Chen, Mohammad Miralinaghi, Tiantian Chen, Pei Li, and Samuel Labi. Why did the ai make that decision? towards an explainable artificial intelligence (xai) for autonomous driving systems. *Transportation Research Part C: Emerging Technologies*, 156:104358, 2023. doi: 10.1016/j.trc.2023.104358. URL https://doi.org/10.1016/j.trc.2023.104358.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. Explainable object-induced action decision for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9523–9532, 2020.

Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving, 2019. URL https://arxiv.org/abs/1811.05432.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1151–1159, 2017. doi: 10.1109/CVPR.2017.128.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks, 2016. URL https://arxiv.org/abs/1511.06732.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Hugging Face. Smolvlm-instruct. https://huggingface.co/HuggingFaceTB/SmolVLM-Instruct, 2024. Accessed: 2025-04-11.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. URL https://arxiv.org/abs/2504.05299.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.