Label-Efficient Fine-Tuning of VLMs for Interpretable Autonomous Driving via RLOO Algorithm

Kai-Yuan Jeng[*]
National Tsing Hua University
Department of Computer Science
kaiyuanjeng@gapp.nthu.edu.tw

Pei-Hsun Wu[*]
National Yang Ming Chiao Tung University
Graduate Degree Program of Robotics
wn2001.en13@nycu.edu.tw

Chen-Fang Hu
National Yang Ming Chiao Tung University
Arête Honors Program
tracy.ls11@nycu.edu.tw

Abstract

This work investigates fine-tuning Vision-Language Models (VLMs) with Reinforcement Learning (RL) for End-to-End Autonomous Driving (E2E-AD) systems. While VLMs provide interpretability benefits through natural language outputs, existing approaches suffer from heavy reliance on labeled data. We develop RL methodologies that enable VLMs to generate both driving actions and scene descriptions with significantly reduced supervision. Specifically, we adapt the REINFORCE Leave-One-Out (RLOO) algorithm for vision-language tasks in autonomous driving contexts. Our approach fine-tunes VLMs to process image frames of traffic scenes and produce text sequences containing scene descriptions and appropriate driving control actions, reducing dependency on labeled datasets while maintaining interpretability advantages.

TL;DR: We adapt REINFORCE Leave-One-Out to fine-tune Vision-Language Models for autonomous driving, reducing labeled data requirements while preserving interpretability benefits.

1 Introduction

Recent advances in Vision-Language Models (VLMs) open new possibilities for End-to-End Autonomous Driving (E2E-AD) systems [Wang et al., 2024, Xu et al., 2024, Jia et al., 2025]. Unlike traditional approaches that often focus solely on outputting low-level control signals (e.g., throttle, steering) and lack interpretability, pretrained VLMs can process raw visual inputs to directly generate both driving commands and human-understandable justifications. This capability paves the way for more interpretable and reactive decision-making in safety-critical scenarios.

However, leveraging these powerful models for E2E-AD is not straightforward. Pretrained VLMs are designed for general vision-language tasks, leading to a significant domain adaptation gap when applied to specific traffic scenarios. Furthermore, current fine-tuning paradigms heavily rely on large, fixed labeled datasets, limiting the scalability and generalizability required for practical E2E-AD applications.

To address these challenges, we adapt reinforcement learning (RL) techniques to fine-tune a VLM for the E2E-AD scenario. Specifically, we adapt the REINFORCE Leave-One-Out (RLOO) algorithm

[*] These authors contributed equally to this work. (See Section 7)

[Kool et al., 2019], recently explored in Reinforcement Learning from Human Feedback (RLHF) scenarios [Ahmadian et al., 2024]. This choice is motivated by its advantages over commonly adopted Proximal Policy Optimization (PPO) methods [Schulman et al., 2017, Ouyang et al., 2022], as it avoids the need for training a separate critic network and simplifies the complex hyperparameter tuning associated with PPO-based frameworks.

Our goal is to enable a tuned VLM to generate both scene descriptions and driving actions with minimal supervision while avoiding the computational overhead of critic network training. The key contributions of this work include:

1. **RLOO Adaptation to Vision-Language Tasks:** We extend RLOO from current RLHF applications to cross-modal vision-language generation in autonomous driving contexts.
2. **Semantic Alignment Reward Framework:** We develop a CLIP-based reward mechanism that enables RL training without human preference annotations, reducing reliance on extensively labeled datasets.
3. **Comprehensive Evaluation:** We demonstrate improved data efficiency compared to supervised approaches while maintaining interpretability benefits through natural language generation.

2 Related Work

VLM-based Interpretable Driving VLM-RL [Huang et al., 2024] employs contrastive language goals (CLG) to derive semantic rewards, enabling improved vision-language alignment. However, the training methodology still requires human-annotated data or pre-collected contrastive examples. 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, achieving strong results on the BDD Object Induced Actions (BDD-OIA) dataset [Xu et al., 2020]. However, this work focuses on introducing new model architectures and remains within the supervised fine-tuning domain, requiring large labeled datasets and thus limiting scalability.

RL-based Autonomous Driving AlphaDrive [Jiang et al., 2025] achieves strong results in multi-modal planning using a two-stage training strategy (SFT + GRPO [Shao et al., 2024]) with tailored reward terms including action-weighted rewards and format consistency. However, it relies heavily on large-scale preference-labeled datasets, limiting scalability. Wang et al. [2019] proposes a sparse attention model over object features to generate actions from visual input, improving robustness in low-data and urban driving settings. However, the proposed attention maps only partially and indirectly support interpretability, and the AD policy generates only low-level control actions rather than explicitly outputting human-readable explanations.

Critic-free Policy Optimization Methods Ren et al. [2017] formulated image captioning as a sequence generation task and proposed a basic REINFORCE algorithm [Williams, 1992] training framework. However, a critic network is still required for baseline estimation. In the same context, Rennie et al. [2017] proposed Self-Critical Sequence Training (SCST), which uses greedy sampling of the sequence model as the REINFORCE baseline, eliminating the need for a separate critic network.

In the context of language model fine-tuning and RLHF, although PPO [Schulman et al., 2017] is commonly adopted [Ouyang et al., 2022], it requires critic training due to the nature of the Actor-Critic framework [Konda and Tsitsiklis, 1999], which is computationally expensive in Large Language Model (LLM) or VLM contexts. As demonstrated by Ahmadian et al. [2024], simple REINFORCE-style policy optimization can surpass the performance of PPO-style optimization, with RLOO [Kool et al., 2019] providing the best results. Compared to SCST, RLOO serves as an improved generalized version, since RLOO leverages multiple samples for baseline estimation while SCST relies on a single greedy sample. Another recently proposed critic-free method is Group Relative Policy Optimization (GRPO) [Shao et al., 2024], which is a variant of the PPO framework. However, PPO-based methods still suffer from the disadvantage of complex hyperparameter tuning, leading to our adaptation of RLOO.

Limitations of Existing Approaches 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 extensive labeled datasets [Jiang et al., 2025]. Additionally, most RL-based approaches rely on computationally expensive critic networks for training [Ren et al., 2017, Ouyang et al., 2022] or complex hyperparameter tuning [Shao et al., 2024], further limiting their scalability. These approaches fail to address the fundamental challenge of reducing data dependency while maintaining interpretability, motivating our critic-free, label-efficient approach.

3 Problem Formulation

As described earlier (see Section 1), our goal is to fine-tune a pretrained VLM for the E2E-AD scenario. Specifically, we aim to enable the model to generate both the corresponding action and explicit justifications in natural language format. This task can be regarded as a *video understanding problem*, which is an extension of *image captioning*, but with two notable distinctions: 1) **Processing a sequence of images** (i.e., continuous frames) to capture temporal relationships within the traffic scene, which can be addressed by directly combining the semantic embeddings of a batch of frames; and 2) **Generating more detailed and well-aligned descriptions of the scene**, which should be informative enough and should serve as the additional information to be leveraged for the subsequent downstream task.

To address the critical challenges identified earlier (see Section 1), we propose a two-stage inference framework for our target VLM. Each stage is designed to address a specific challenge: 1) **Stage 1: Text Sequence Generation via Reinforcement Learning.** The first stage focuses on adapting the VLM to process video image inputs and generate descriptive outputs based on the image information. These outputs will then serve as valid justifications for subsequent AD control and 2) **Stage 2: Action Selection and Formatting Alignment via Token-wise Supervision.** The second stage enables the model to leverage both the image information and the corresponding justifications to generate the appropriate AD control.

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 of Ren et al. [2017], we reformulate video understanding as a decision-making process, where the agent typically consists of two components: 1) a policy network (the VLM in our case) responsible for decision-making, and 2) a value network that evaluates or estimates the state value.

3.1 Model-Free Control

It is worth noting that sequential language modeling can be formulated as a Markov decision process (MDP) [Ranzato et al., 2016], where the state 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 the VLM's auto-regressive generation of tokens begins only after all image frames have been fully incorporated. For the given image frames, our objective is to generate a detailed and well-aligned description of the observed environment (i.e., the images). This description will then serve as a justification and support the generation of the optimal action for AD control.

In general, the image (or video) captioning task can be regarded as model-free control, as it does not leverage or learn the MDP's transition dynamics for planning. Instead, the model auto-regressively generates tokens based on the initial encoded visual information and previously generated tokens, directly controlling the next token selection. The RL algorithm we adopt (see Section 3.3) updates the policy solely based on environment feedback—such as the semantic similarity between the input image and the generated description (see Section 3.4)—without constructing any explicit model of the underlying environment.

3.2 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 be parameterized as M_ϕ, allowing the model to learn an optimal ϕ to better capture the scene's dynamics. The textual information we aim to extract from $\mathbf{I}$ is represented as a sequence of language tokens $\mathbf{T} = (a_1, a_2, \ldots, a_N)$ in the language modeling framework, where N is the total number of generated tokens.

At time step t, the agent has observed the image frames $\mathbf{I}$ and the text sequence $\mathbf{T}_t = (a_1, a_2, \ldots, a_t)$ generated up to step t, where $1 \leq t \leq N$. The state is therefore 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 leads to an updated state consisting of the image frames $\mathbf{I}$ and the extended token sequence $\mathbf{T}_{t+1}$.

Thus, if we treat the image captioning task as a sequential token generation problem, we may define:

- **The State Space** as $\mathcal{S} = \{s_t = (\mathbf{I}, \mathbf{T}_t) \mid 0 \leq t \leq N\}$ (where $\mathbf{T}_0$ represents an empty sequence).
- **The Action Space** $\mathcal{A}$ as the vocabulary $\mathcal{Y}$, from which tokens are drawn, enabling the model to generate justification text for selecting the desired action. At each step t, the next action is $a_{t+1} \in \mathcal{A}$.

However, as suggested by Ahmadian et al. [2024], it is more appropriate to define the complete generated sequence as a single action and reduce the problem to a bandit setting (see Section 3.3). Under this formulation, we re-define:

- **The State Space** as $\mathcal{S} = \{\mathbf{I} \mid \mathbf{I} \in \mathcal{D}\}$ where $\mathcal{D}$ is the training dataset, and each state represents an input image sequence.
- **The Action Space** conceptually represents all possible complete sequences, thus $\mathcal{A} = \mathcal{Y}^N$. In practice, actions are sampled from the policy distribution over $\mathcal{Y}^N$ rather than enumerating the entire space.

3.3 Policy Optimization

Background and Motivation We adopt policy gradient-based methods for sequence generation, leveraging their suitability for token selection tasks in discrete action spaces (see Section 3.2). While the REINFORCE algorithm [Williams, 1992] serves as the standard baseline and the Advantage Actor-Critic (A2C) framework [Konda and Tsitsiklis, 1999] is commonly employed, Rennie et al. [2017] has highlighted limitations due to inaccurate critic estimation, proposing Self-Critical Sequence Training (SCST) as a remedy.

Although Proximal Policy Optimization (PPO) [Schulman et al., 2017] represents a standard choice in RLHF scenarios, Ahmadian et al. [2024] argues that *"PPO is not the right tool for doing RL in RLHF"* since pretrained models already constitute well-initialized policies, making simpler REINFORCE-style algorithms more appropriate. Furthermore, since reward signals are only obtained at sequence completion, we adopt a bandit formulation where complete sequences are treated as single actions, avoiding the complexity of modeling partial completions.

REINFORCE Algorithm The policy network π_θ is instantiated by the pretrained VLM we aim to fine-tune, parameterized by θ and denoted as $\mathbf{VLM}_\theta$. In our bandit formulation, the policy generates complete sequences $\mathbf{T} \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I})$ given input images $\mathbf{I}$. The REINFORCE objective is:

$$\mathcal{J}_{\text{REINFORCE}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}, \mathbf{T}\sim\mathbf{VLM}_\theta(\cdot|\mathbf{I})} \left[\mathcal{R}(\mathbf{I}, \mathbf{T})\right],$$

with policy gradient:

$$\nabla_\theta \mathcal{J}_{\text{REINFORCE}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}, \mathbf{T}\sim\mathbf{VLM}_\theta(\cdot|\mathbf{I})} \left[\mathcal{R}(\mathbf{I}, \mathbf{T}) \nabla_\theta \log \mathbf{VLM}_\theta(\mathbf{T} \mid \mathbf{I})\right].$$

To reduce gradient variance, REINFORCE with baseline incorporates a state-dependent baseline function:

$$\nabla_\theta \mathcal{J}_{\text{REINFORCE-baseline}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}, \mathbf{T}\sim\mathbf{VLM}_\theta(\cdot|\mathbf{I})} \left[\big(\mathcal{R}(\mathbf{I}, \mathbf{T}) - b(\mathbf{I})\big) \nabla_\theta \log \mathbf{VLM}_\theta(\mathbf{T} \mid \mathbf{I})\right],$$

where $b(\mathbf{I})$ represents the baseline function conditioned on the input state.

Self-Critical Sequence Training SCST [Rennie et al., 2017] circumvents explicit value network training by utilizing the policy network itself as the baseline. This approach employs two decoding strategies:

- **Greedy Decoding:** $\mathbf{T}^{\text{greedy}} = \mathbf{VLM}_\theta^{\text{greedy}}(\mathbf{I})$ - deterministic baseline
- **Sampled Decoding:** $\mathbf{T}^{\text{sample}} \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I})$ - stochastic exploration

The SCST policy gradient is:

$$\nabla_\theta \mathcal{J}_{\text{SCST}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}, \mathbf{T}^{\text{sample}}\sim\mathbf{VLM}_\theta(\cdot|\mathbf{I})} \left[\Big(\mathcal{R}(\mathbf{I}, \mathbf{T}^{\text{sample}}) - \mathcal{R}(\mathbf{I}, \mathbf{T}^{\text{greedy}})\Big) \nabla_\theta \log \mathbf{VLM}_\theta(\mathbf{T}^{\text{sample}} \mid \mathbf{I})\right].$$

REINFORCE Leave-One-Out While SCST relies on a single greedy baseline, RLOO [Kool et al., 2019] constructs more robust baselines from multiple samples, potentially reducing variance further through better baseline estimation. Given k independent samples $\mathbf{T}^1, \mathbf{T}^2, \ldots, \mathbf{T}^k \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I})$, RLOO computes:

$$\nabla_\theta \mathcal{J}_{\text{RLOO}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}}\left[\sum_{i=1}^{k}\Big(\mathcal{R}(\mathbf{I}, \mathbf{T}^i) - \frac{1}{k-1}\sum_{j\neq i}\mathcal{R}(\mathbf{I}, \mathbf{T}^j)\Big)\nabla_\theta \log \mathbf{VLM}_\theta(\mathbf{T}^i \mid \mathbf{I})\right], \quad (1)$$

where each sample $\mathbf{T}^i$ uses the average reward of the remaining $k-1$ samples as its baseline.

The key advantages of RLOO over SCST include: 1) **Improved baseline quality** - the multi-sample average provides a more accurate estimate of expected performance than a single greedy sequence; 2) **Better variance reduction** - by averaging over multiple samples, the baseline becomes more stable and less dependent on individual sequence quality; and 3) **Exploration preservation** - unlike the deterministic greedy baseline in SCST, RLOO's baseline reflects the actual sampling distribution of the policy.

However, RLOO requires generating k samples per training step, increasing computational cost by a factor of k compared to SCST. The choice of k presents a trade-off between baseline quality and computational efficiency, typically ranging from $k = 4$ to $k = 16$ in practice [Kool et al., 2019].

3.4 Reward Signal Modeling

The reward signal $\mathcal{R}$ is designed to evaluate the quality of output sequences generated by our policy (see Section 3). We adopt the *semantic alignment reward* that assesses the semantic correspondence between image frames and generated text sequences, as proposed in [Ren et al., 2017]. To obtain the semantic similarity between images and text, we adapt a pretrained CLIP model [Radford et al., 2021] as our reward function. To further tailor the CLIP model for the end-to-end autonomous driving (E2E-AD) scenario, we perform additional fine-tuning (see Appendix B).

Given a sequence of images $\mathbf{I} = \{I_1, I_2, \ldots, I_M\}$, we extract frame-level image features using CLIP's image encoder. These features are then aggregated through average pooling to obtain a unified image representation:

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

where $\mathbf{CLIP}^{\text{image}}(I_i)$ denotes the normalized embedding of the i-th frame. Correspondingly, the generated text sequence $\mathbf{T}$ is encoded using CLIP's text encoder to obtain the text embedding:

$$\mathbf{t} = \mathbf{CLIP}^{\text{text}}(\mathbf{T}).$$

The semantic alignment reward is computed as the cosine similarity between the aggregated image embedding $\mathbf{v}$ and the text embedding $\mathbf{t}$:

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

where both embeddings are ℓ_2-normalized by their respective CLIP encoders, reducing the cosine similarity computation to a simple dot product. This reward formulation encourages our target VLM to generate text sequences that are semantically coherent with the image frames.

4 Methodology

Our approach addresses the challenge of reducing labeled data dependency in training VLMs for E2E-AD scenarios (see Section 1). Existing datasets such as BDD-OIA [Xu et al., 2020] follow an image-text pair format where image frames depict traffic scenes and text sequences contain both driving actions and corresponding justifications. However, collecting such richly annotated data is expensive and time-consuming.

We propose a two-stage training framework that enables the target VLM to: 1) generate comprehensive text sequences from input image frames, and 2) leverage both visual and textual information to select appropriate driving actions. This approach reduces the reliance on manually crafted justifications while maintaining decision-making capabilities.

Stage 1: Text Sequence Generation via Reinforcement Learning (see Algorithm 2 and Fig. 2) In the first stage, we employ reinforcement learning (RL) (see Section 3.3) to fine-tune the target VLM for generating descriptive text sequences from traffic scenes. The policy network learns to produce semantically rich text sequences that capture relevant visual information without requiring ground-truth textual annotations.

The reward signal is derived from the semantic alignment between input image frames and generated text sequences (see Section 3.4) using a fine-tuned CLIP model (see Algorithm 1), encouraging the model to produce descriptions that accurately reflect the visual content. This stage operates in a self-supervised manner, eliminating the need for human-annotated text sequences.

Stage 2: Action Selection and Formatting Alignment via Token-wise Supervision (see Algorithm 3 and Fig. 3) After RL fine-tuning, the VLM can generate informative text sequences. We hypothesize that combining visual information with generated text sequences provides richer context for driving decisions compared to using image frames alone.

In the second stage, we prompt the target VLM with: 1) the original traffic scene image frames, 2) the generated text sequence from Stage 1, and 3) a structured action prompt "`Based on the caption and image, what is the next action: <forward, stop, turn left, turn right>?`" to guide action generation. This multi-modal approach leverages both image frame perception and text sequence reasoning for decision-making.

The BDD-OIA dataset [Xu et al., 2020] defines driving actions as a four-choice classification problem with specific text sequence formats. Since our VLM generates free-form text sequences, potential formatting misalignment may occur between generated actions and dataset expectations.

To address this issue, we apply token-wise supervised fine-tuning using BLEU score [Papineni et al., 2002] as the optimization objective to ensure text sequence format consistency. This process preserves semantic correctness while aligning output structures with expected patterns.

Finally, we evaluate the overall performance of our fine-tuned VLM using F1-score metrics, measuring the accuracy of action selection.

Training Pipeline Our complete training pipeline consists of:

1. **RL-based Text Sequence Generation:** Fine-tune VLM to generate descriptive text sequences using semantic alignment rewards.
2. **Multi-modal Action Selection:** Prompt the VLM with image frames, generated text sequences, and action cues.
3. **Text Sequence Format Alignment:** Apply supervised fine-tuning to ensure output format consistency.
4. **Performance Evaluation:** Assess action selection accuracy using F1-score metrics.

Algorithm 1 CLIP Fine-tuning for Semantic Alignment Reward

```
Require: BDD-OIA dataset [Xu et al., 2020] $\mathcal{D}$.
Ensure: A fine-tuned CLIP model as the reward function.
Initialize a pretrained CLIP model [Radford et al., 2021].
Split dataset $\mathcal{D}$ into training subset $\mathcal{D}_{\text{train}}$ and validation subset $\mathcal{D}_{\text{val}}$.
Fine-tune the CLIP model on $\mathcal{D}_{\text{train}}$ according to Appendix B.
Evaluate the performance of the fine-tuned model on $\mathcal{D}_{\text{val}}$.
return The best performing model checkpoint.
```

5 Experiment Setting

5.1 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.

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

Algorithm 2 VLM Text Sequence Generation Fine-tuning via RLOO

```
Require: Image frames extracted from $\mathcal{D}$, fine-tuned CLIP reward model.
Ensure: A fine-tuned VLM for generating scene descriptions.
Initialize policy network with the pretrained VLM.
while training epoch not completed do
    for each image frame sequence $\mathbf{I} \in \mathcal{D}$ do
        Sample $k$ independent text sequences: $\mathbf{T}^1, \mathbf{T}^2, \ldots, \mathbf{T}^k \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I})$.
        Compute reward $\mathcal{R}(\mathbf{I}, \mathbf{T}^i)$ for each text sequence using the CLIP reward model.
        Update policy parameters $\theta$ according to RLOO gradient (Equation 1).
    end for
end while
return The fine-tuned VLM.
```

Algorithm 3 VLM Action Format Alignment via Supervised Fine-tuning

```
Require: Full dataset $\mathcal{D}$ with ground-truth actions.
Ensure: A fine-tuned VLM for generating formatted actions.
Initialize with the VLM from Algorithm 2.
while training epoch not completed do
    for each image frame sequence $\mathbf{I}$ with corresponding ground-truth action $\mathbf{A}$ in $\mathcal{D}$ do
        Generate description: $\mathbf{T} \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I})$.
        Concatenate $\mathbf{T}$ with the action prompt to form a query $\mathbf{Q}$.
        Generate predicted action: $\hat{\mathbf{A}} \sim \mathbf{VLM}_\theta(\cdot \mid \mathbf{I}, \mathbf{Q})$.
        Compute BLEU score [Papineni et al., 2002] between $\hat{\mathbf{A}}$ and ground-truth $\mathbf{A}$.
        Update VLM parameters $\theta$ via gradient descent on negative BLEU loss.
    end for
end while
return The fine-tuned VLM.
```

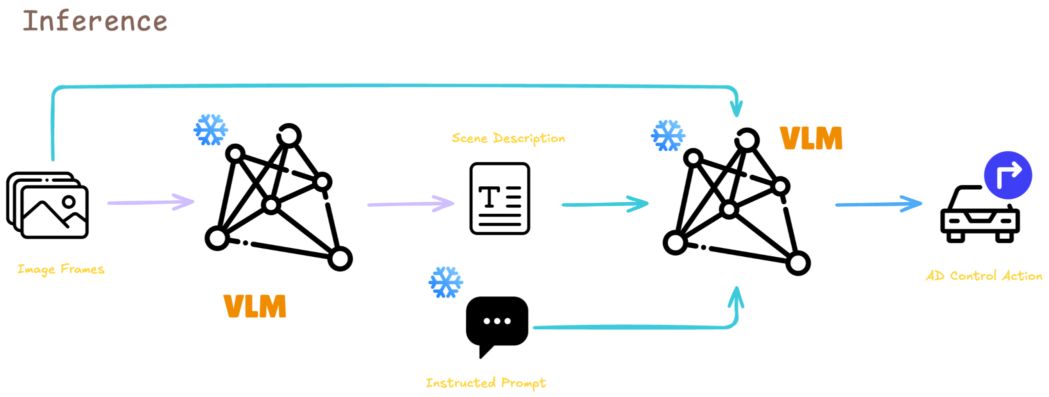


Figure 1: Two-stage inference pipeline. **Stage 1:** The VLM processes image frames to generate scene descriptions. **Stage 2:** Combining the image frames, generated scene descriptions, and an instructed prompt, the VLM produces the AD control action. Note that during inference, both the VLM parameters and the instructed prompt are fixed.

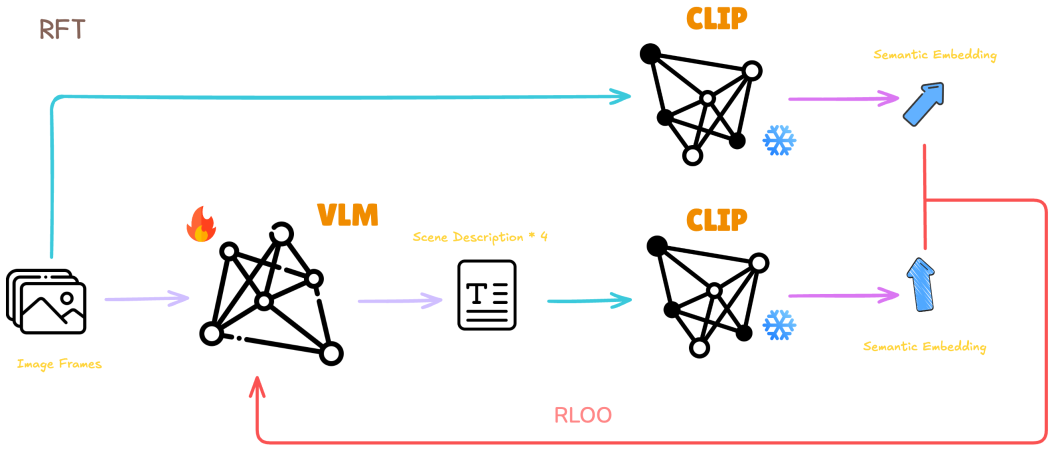


Figure 2: Reinforcement fine-tuning pipeline (Stage 1). Given image frames, the VLM generates multiple candidate descriptions. The CLIP model computes semantic alignment rewards for each sampled sequence. REINFORCE Leave-One-Out (RLOO) policy optimization is then performed using these multi-sample rewards to update the VLM.

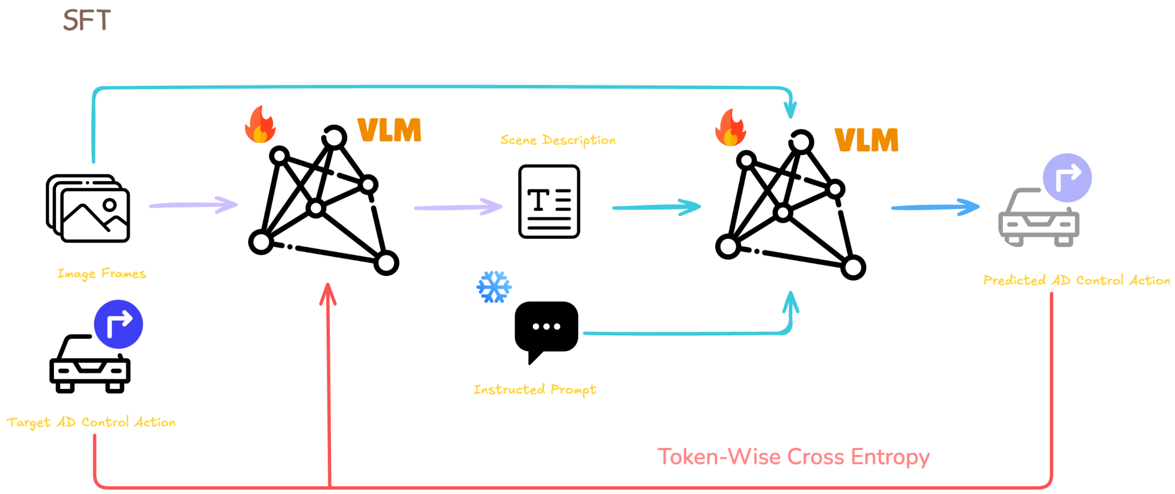


Figure 3: Supervised fine-tuning pipeline (Stage 2). The VLM takes multi-modal inputs: image frames, generated scene descriptions from Stage 1, and an instructed prompt. The predicted actions are compared with ground-truth using BLEU score, and the VLM is updated via supervised learning to ensure proper formatting.

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**: A harmonic mean of precision and recall that accounts for class imbalance in action predictions, following Xu et al. [2020] (Fig. 5).
  - **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.

5.2 Pretraining Limitations and Fine-Tuning for Reward Alignment}

In earlier toy example , the model had not yet undergone fine-tuning, resulting in significantly subpar performance during subsequent training (see Appendix A). For example, when we evaluated the pretrained CLIP model's performance using metrics such as the match percent (shortened as "Match%" in Table 1) and average similarity probabilities (shortened as "Avg. Sim." in Table 1) between images and ground-truth textual explanations, we observed that it struggled to accurately identify the semantic correspondence between the image and the described vehicle action and its underlying reason. Even though both the match percent and average similarity probabilities yielded values of 1 and 100 % respectively (as seen in Table 1) the same results were obtained even when testing with completely irrelevant texts (e.g., simple strings like "123" or "abc"). This indicated that the model was not truly discerning meaningful relationships between images and text, but instead returning uniformly high scores regardless of semantic content.

This issue was further confirmed by the actual prediction performance: the pretrained model achieved only 14.03% accuracy in predicting actions, 2.44% accuracy in predicting reasons, and 1.79% accuracy for the full output. These results make it clear that despite the high similarity metrics, the model failed to make meaningful predictions, highlighting a significant misalignment between similarity-based matching and true semantic understanding. This mismatch severely impacted the reliability of reward signals and undermined the overall effectiveness of downstream training. To address the limitations of the pretrained CLIP model, we performed fine-tuning using supervision on either the action or reason annotations. As summarized in Table 1, this fine-tuning process led to more meaningful and discriminative similarity distributions. Specifically, fine-tuning on actions improved the alignment scores for both action and output pairs while simultaneously reducing the irrelevant similarity with reason texts. In contrast, fine-tuning on reasons significantly enhanced the model's ability to align image-reason pairs, achieving a match percent of 94.43% and an average similarity of 0.9289, while effectively suppressing spurious alignment with unrelated actions. These results

Table 1: CLIP Similarity Evaluation: Pretrained vs. Fine-Tuned (Action / Reason)

Image-text Pairing	**Pretrained**		**Action FT**		**Reason FT**	
	Avg. Sim.	**Match %**	**Avg. Sim.**	**Match %**	**Avg. Sim.**	**Match %**
Image vs. Output	1.0000	100.00%	0.8533	87.41%	0.9113	93.33%
Image vs. Action	1.0000	100.00%	0.8682	87.76%	0.5525	56.67%
Image vs. Reason	1.0000	100.00%	0.4007	22.39%	0.9289	94.43%

demonstrate that task-specific fine-tuning enables CLIP to produce more reliable and semantically grounded reward signals for downstream reinforcement learning tasks. Based on the comparative performance shown in Table 1, we then selected the image-reason fine-tuned model as the final version used in the subsequent stages of reinforcement training.

Hence, based on the results shown in Table 1, we selected the image-reason fine-tuned model as the final version to be used in the later stages of the reinforcement learning tasks.

5.3 Baseline Method & Hypermeter Setting

Training Configuration:

- **Pretrained Model:** SmolVLM-Instruct [Face, 2024].
- **Baseline - SFT:** The baseline model is obtained through supervised fine-tuning (SFT) using task-specific action-reason pairs as training data. Each training sample is formatted as: *"Image input $\rightarrow$ Text output: Action: [label]. Reason: [label]."*

  The training process spans 10 epochs with an initial learning rate of 1e-4, a batch size of 4, and a gradient accumulation step of 4 to accommodate GPU memory constraints. Weight decay is set to 0.01 to mitigate overfitting. The optimizer used is `adamw_hf`, and model checkpoints are saved at the end of every epoch.

  The objective is to minimize the cross-entropy loss between the model's generated output and the ground-truth label sequence, encouraging the model to produce coherent and semantically aligned outputs. This baseline serves as a strong reference for evaluating the impact of reinforcement-based strategies in later stages.

  - Training epochs: 10
  - Initial learning rate: 1e-4
  - batch size: 4
  - Gradient accumulation steps: 4
  - Weight decay: 0.01
  - Save strategy: `epoch`
  - logging strategy = "steps"
  - save strategy: "epoch"
  - optimizer: `adamw_hf`

  The example results in Fig. 4.
- **Stage 1 - RLOO:** In Stage 1, we apply reinforcement fine-tuning using reward signals derived from a CLIP-based scoring function. Specifically, the model is prompted to generate multiple candidate outputs per input (using top-k and top-p sampling), and the top-ranked output based on CLIP similarity to the image is selected as pseudo-label supervision.

  This stage is trained for 5 epochs using a fixed learning rate of 1e-6 and a sampling temperature of 1.5 to promote diversity. The training uses a policy-gradient-like objective where the reward corresponds to the CLIP score, guiding the model to favor semantically image-aligned outputs even without direct ground-truth action-reason labels.

  - Training epochs: 5
  - Constant learning rate: 1e-6
  - Top-p sampling: 0.95
  - Top-k sampling: 50
  - Temperature: 1.5
  - Number of return sequences: 8
  - Save strategy: `epoch`
- **Stage 2 - Action SFT:** In Stage 2, we perform a follow-up supervised fine-tuning (Action SFT) to reintroduce structural guidance based on action labels. The model continues training for 3 epochs with the same learning rate (1e-4) and optimizer settings as the baseline. The training objective is again cross-entropy loss, but the supervision only focuses on the action component of the output.

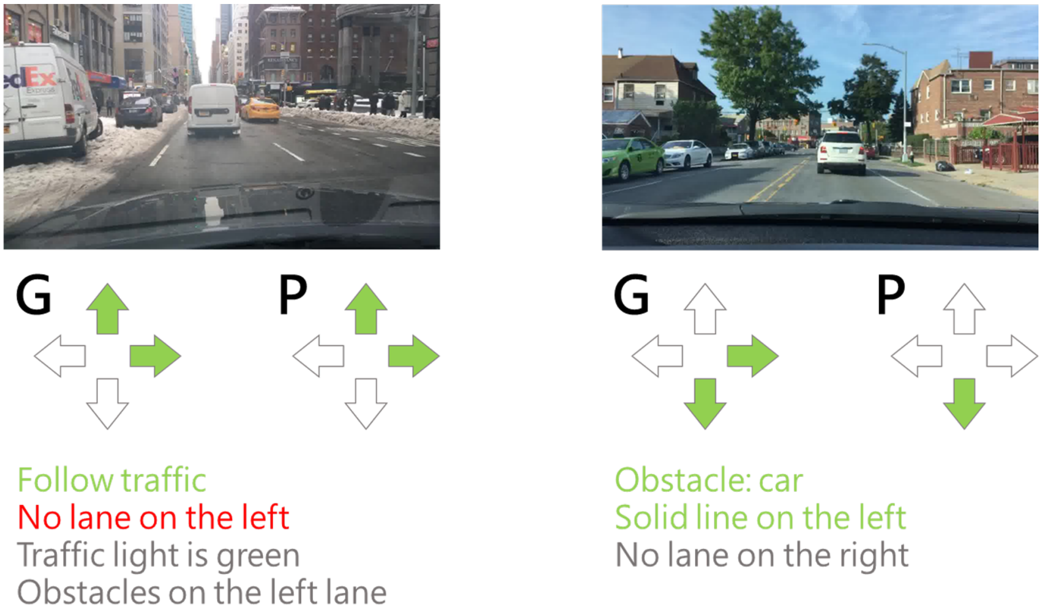


Figure 4: Example results. Figure Legend: **G** denotes the ground-truth and **P** indicates the model prediction. The arrows follow a clockwise order representing the actions: move forward, stop/slow down, turn left, and turn right. Green arrows denote the set of driving actions deemed executable by the model. Explanation texts are color-coded as follows: green for true positives, red for false positives, and gray for false negatives.

This step aims to restore the task-specific structure after the unconstrained learning in Stage 1, combining the semantic richness learned via CLIP-guided reinforcement with the format consistency enforced by supervised learning.

- Training epochs: 3
- Initial learning rate: 1e-4
- Gradient accumulation steps: 4
- Weight decay: 0.01
- Save strategy: `epoch`

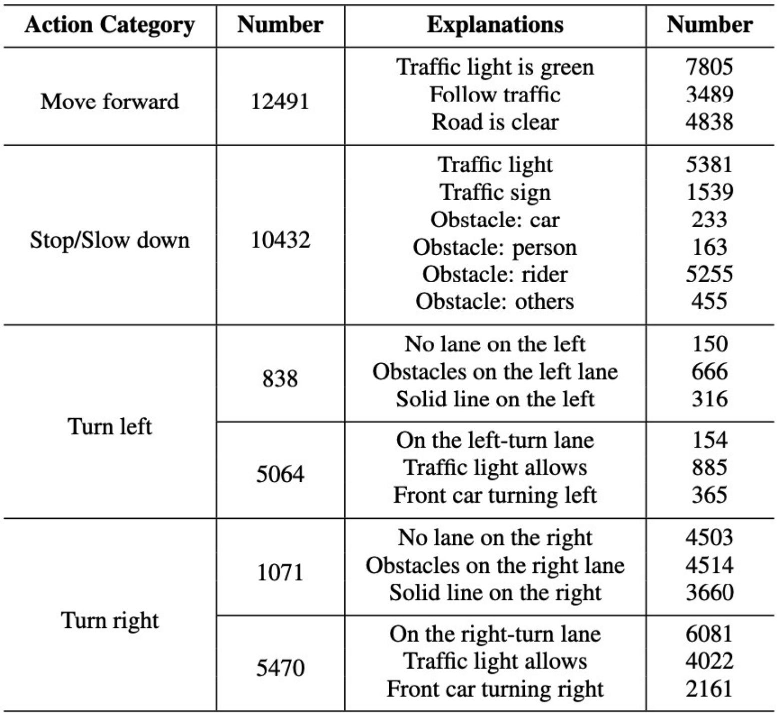
5.4 Benchmark Dataset

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. 5).

6 Evaluation Result

6.1 Ground Truth (GT)

- **Data Format**: `"Action: stop. Reason: Traffic light is not green."`
- **CLIP score**



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

Figure 5: Actions and explanations in BDD-OIA [Xu et al., 2020]

- Average action clip-score: 0.5524
- Match percent: 56.07%
- Average reason clip-score: 0.9293
- Match percent: 94.54%

6.2 Baseline - SFT

- **Output Format**: `"Action: stop. Reason: Traffic light is not green."`
- **Accuracy**:
  - Action accuracy: 90.07%
  - Reason accuracy: 96.89%
- **Macro-F1 (mF1)**:
  - Action macro-F1: 80.15%
  - Reason macro-F1: 66.75%
- **Overall F1 ($\mathbf{F1_{all}}$)**: The overall F1 results in Table 2 and Table 3.
- **CLIP score**:
  - Average action clip-score: 0.5591
  - Match percent: 57.32%
  - Average reason clip-score: 0.9363
  - Match percent: 95.28%

As shown in Table 2 and Table 3, the model performs best in predicting the `Stop` action and its associated reasons, achieving F1 scores of 89.24% and 86.3%, respectively. This strong performance is attributed to the dominance of the `Stop` category in the test set (84.12%) and the presence of visually salient cues such as red traffic lights and vehicle obstacles.

In contrast, the prediction performance significantly degrades for the `Left` and `Right` actions, which only account for 2.98% and 2.73% of the test data, respectively. Notably, the F1 score

Table 2: Test set distribution and F1 score of various actions.

Action	Test GT (%)	Test GT (#)	Test F1 score
Forward	10.17	41	27.16
Stop	84.12	339	89.24
Left	2.98	12	14.29
Right	2.73	11	0.00

Table 3: Test set distribution and F1 score of various reasons.

Reason	Test GT (%)	Test GT (#)	Test F1 score
Follow traffic	3.47	14	16.7
Road is clear	4.96	20	12.5
Traffic light is green	1.99	8	0.0
Obstacle: car	18.11	73	75.2
Obstacle: person	5.71	23	37.0
Obstacle: rider	1.49	6	0.0
Obstacle: others	0.00	0	0.0
Traffic light is not green	51.61	208	86.3
Traffic sign	6.45	26	52.2
Front car turning left	0.25	1	0.0
On the left-turn lane	1.99	8	0.0
Traffic light allows to turn left	0.50	2	0.0
Front car turning right	0.00	0	0.0
On the right-turn lane	0.25	1	0.0
Traffic light allows to turn right	0.00	0	0.0
Obstacles on the left lane	0.25	1	0.0
No lane on the left	1.74	7	0.0
Solid line on the left	0.50	2	0.0
Obstacles on the right lane	0.25	1	0.0
No lane on the right	0.50	2	0.0
Solid line on the right	0.00	0	0.0

for `Right` is 0%, indicating the model's inability to recognize such rare actions. Furthermore, many reason classes have extremely limited samples (e.g., `Obstacle on the left lane`, `On the right-turn lane`), which hinders the model's ability to generalize under infrequent conditions.

6.3 Stage 1 - RLOO

- **Output Format**: `"A truck and a delivery truck are parked in front of an archway gate in an empty street."`

  Since no textual prompt was provided during inference, the generated outputs tend to exhibit a bias toward generic image captioning rather than task-specific descriptions.Since the generated outputs do not follow the expected format specifying both action and reason, accuracy and F1-score cannot be computed.
- **CLIP score**:
  - Average clip-score: 0.4188
  - Match percent: 42.43%

Despite the use of reinforcement learning in Stage 1 (RLOO), the generated outputs remain largely generic and caption-like (e.g., ‘‘`A truck and a delivery truck are parked in front of an archway gate in an empty street.`’’). This is primarily due to the absence of an explicit textual prompt or format constraint during generation. As a result, the model optimizes toward high CLIP similarity scores by producing general image descriptions, rather than structured, task-specific outputs containing both an action and a reason. Consequently, the generated results deviate from the intended format, making it impossible to compute classification metrics such as accuracy and

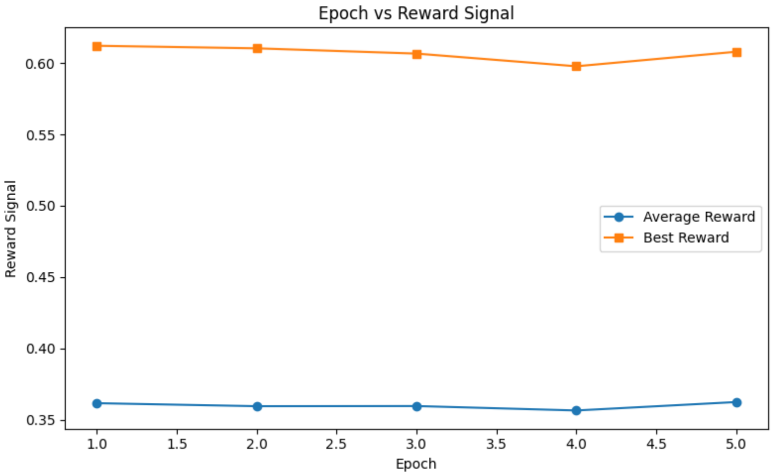


Figure 6: RLOO training epoch vs reward signal. The average reward refers to the mean CLIP-based reward computed across all training samples, where each sample is associated with 8 generated candidate outputs. In contrast, the best reward represents the average of the highest reward selected from each set of 8 candidates per training sample. These metrics respectively reflect the general quality of all generated outputs and the optimal semantic alignment achievable by the model during training.

F1-score. Furthermore, the average CLIP score (0.4188) and match percentage (42.43%) indicate only modest semantic alignment between the outputs and the original images.

(Fig. 6).

6.4 Stage 2 - Action SFT

- **Output Format**: `"A crosswalk is visible in the picture.", "Action: Stop."`
- **Accuracy**:
  - Action accuracy: 84.12%
- **CLIP score**:
  - Average clip-score: 0.4258
  - Match percent: 43.67%

Based on the reward progression shown in the figure, we select the model checkpoint at epoch 5 as the initialization point for Stage 2, where supervised fine-tuning is performed using action-level annotations.

In Stage 2 (Action SFT), supervised fine-tuning reintroduces explicit structure to the outputs by enforcing the format `"Action: [label]"` alongside the generated caption (e.g., `‘‘A crosswalk is visible in the picture.’’, ‘‘Action: Stop.’’`). This structural guidance enables standard evaluation using action labels, yielding a high action classification accuracy of 84.12%. However, the CLIP score only marginally improves (average: 0.4258; match: 43.67%) compared to Stage 1. This modest increase suggests that while Action SFT successfully constrains the output to a task-relevant format, it does not significantly enhance the semantic richness or visual alignment as perceived by the CLIP model.

7 Contributions of Each Member

- **Kai-Yuan Jeng (40%):** *Conceptualization; Methodology; Formal Analysis; Writing – Original Draft; Writing – Review & Editing; Visualization.*
  - Led the development of the theoretical framework and RLOO adaptation design.

  - Wrote the initial drafts for the Abstract, Introduction, Related Work, Problem Formulation, and Methodology sections (Section 1, 2, 3, 4).
  - Co-authored the Toy Example section (Appendix A), focusing on methodology and result analysis.
  - Performed the final manuscript integration, review, and editing.
  - Contributed to the poster design (Appendix C) and led the presentation.

- **Pei-Hsun Wu (40%):** *Software; Investigation; Data Curation; Writing – Original Draft.*
  - Implemented the main experimental pipeline, including toy example and VLM fine-tuning.
  - Assisted with the CLIP fine-tuning experiments.
  - Managed data curation for all experiments.
  - Co-authored the Toy Example section (Appendix A), focusing on implementation and result generation.
  - Wrote the initial drafts of Experiment section (Section 5 and 6).
- **Chen-Fang Hu (20%):** *Software; Investigation; Writing – Original Draft; Visualization.*
  - Managed the integration and organization of the project's source code.
  - Implemented the CLIP fine-tuning experiments (Appendix B).
  - Wrote the initial draft for the CLIP Fine-Tuning section (Section 5 and Appendix B).
  - Assisted with the poster design (Appendix C).

Acknowledgments and Disclosure of Funding

This work was conducted as the final project for the course CS 535514: Reinforcement Learning (Spring 2025) at National Yang Ming Chiao Tung University. We thank the instructor Ping-Chun Hsieh and the teaching assistant Bing-Shu Wu for their guidance. All authors participated in project discussions and have reviewed and agreed upon the contributions statement.

AI Usage Statement: During the preparation of this work, the authors utilized AI language models for mainly the following purposes: 1) to refine the language and improve the clarity of the manuscript and 2) to receive suggestions on LaTeX formatting and academic writing conventions. The authors take full responsibility for all content presented in this report.

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`.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *Deep Reinforcement Learning Meets Structured Prediction, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=r1lgTGL5DE`.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL `https://aclanthology.org/2024.acl-long.662/`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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`.

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.

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`.

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`.

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.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017. doi: 10.1109/CVPR.2017.131.

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.

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.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

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

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Appendix

A Toy Example

We initially explored a simplified approach to validate the feasibility of using RL for VLM fine-tuning in E2E-AD scenarios. This preliminary experiment aimed to establish baseline performance and identify potential challenges before developing the full framework.

A.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 [Xu et al., 2020]. 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.

A.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 (see Section 3). 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. 5), 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.

A.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:

Semantic Alignment Reward $\mathcal{R}^1$ measures the semantic similarity between an image $\mathbf{I}$ and the generated justification text, defined as the cosine similarity between their corresponding embeddings, following the approach of Ren et al. [2017]. We calculated the embedding similarity using a pre-trained CLIP [Radford et al., 2021] model (`openai/clip-vit-base-patch32`).

Action-Reasoning Reward $\mathcal{R}^2$ is a score using LLM-as-a-judge [Zheng et al., 2023], which evaluates the logical consistency between the generated action and justification. To evaluate the action-reasoning reward, we use the BLEU-4 score Papineni et al. [2002] 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 final action-reasoning reward. The prompt we used for the LLM judging is:

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

Overall Fine-Tuning Process The combined reward is

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

and we optimize the target VLM parameters via A2C [Konda and Tsitsiklis, 1999] REINFORCE policy optimization [Williams, 1992], along with a trainable, simple multilayer perceptron (MLP) as the critic network. The overall fine-tuning loop proceeds as Algorithm 4.

Algorithm 4 Toy Example Fine-Tuning Process

```
Require: BDD-OIA dataset [Xu et al., 2020] $\mathcal{D}$, a pretrained CLIP [Radford et al., 2021] model
Ensure: A fine-tuned VLM.
while training epoch not completed do
    for each image frames $\mathbf{I} \in \mathcal{D}$ do
        Sample
            $\langle \text{action}, \text{justification} \rangle = \hat{\mathbf{T}} \sim \mathbf{VLM}_\theta(\mathbf{I}).$
        Compute $\mathcal{R}^1$ from $\mathbf{I}$ and the justification text $\langle \text{justification} \rangle$, using the pretrained CLIP (openai/clip-vit-base-patch32).
        Compute $\mathcal{R}^2$ by prompting the LLM to judge with the sampled action $\langle \text{action} \rangle$ and the justification $\langle \text{justification} \rangle$.
        Form the total reward $\mathcal{R} = \alpha\,\mathcal{R}^1 + \beta\,\mathcal{R}^2$ and update $\theta$ via REINFORCE policy gradient [Williams, 1992].
        Update the critic network.
    end for
end while
return A tuned VLM $\mathbf{VLM}_\theta$.
```

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

A.4 Evaluation

Based on the experiments conducted on the toy example, we present preliminary results and observations regarding the baseline methods.

We focus on two key evaluation criteria: 1) the semantic alignment reward $\mathcal{R}^1$ derived from CLIP feedback (i.e., the CLIP-score), which serves as the main optimization target for our reinforcement learning (RL) approach; and 2) the task-level correctness measured by the F1-score, as defined in the BDD-OIA dataset [Xu et al., 2020].

Observations. After three epochs of fine-tuning, the baseline SFT model achieves a semantic similarity of CLIP-score $= 0.2576$ and a correctness score of F1-score $= 0.1166$. In contrast, our RL-trained model obtains the same semantic similarity (CLIP-score $= 0.2576$), but a slightly lower correctness score (F1-score $= 0.09935$). Both CLIP-scores remain relatively low, indicating the difficulty of aligning visual and textual semantics in this setting.

Interestingly, as shown in Fig. 8, the output distribution of our RL model more closely matches the ground truth compared to the SFT baseline. In particular, the F1-score of the action component ($\langle$action$\rangle$) improves after A2C fine-tuning. However, Fig. 9 shows that the overall F1-score drops post-RL, likely due to mismatches in full action-justification pairs.

Hypothesis. We hypothesize that this discrepancy arises from several factors:

- The F1-score evaluation relies on exact matching of action-justification pairs (e.g., `("Follow traffic", "Road is clear")`), which penalizes semantically correct but lexically different responses.

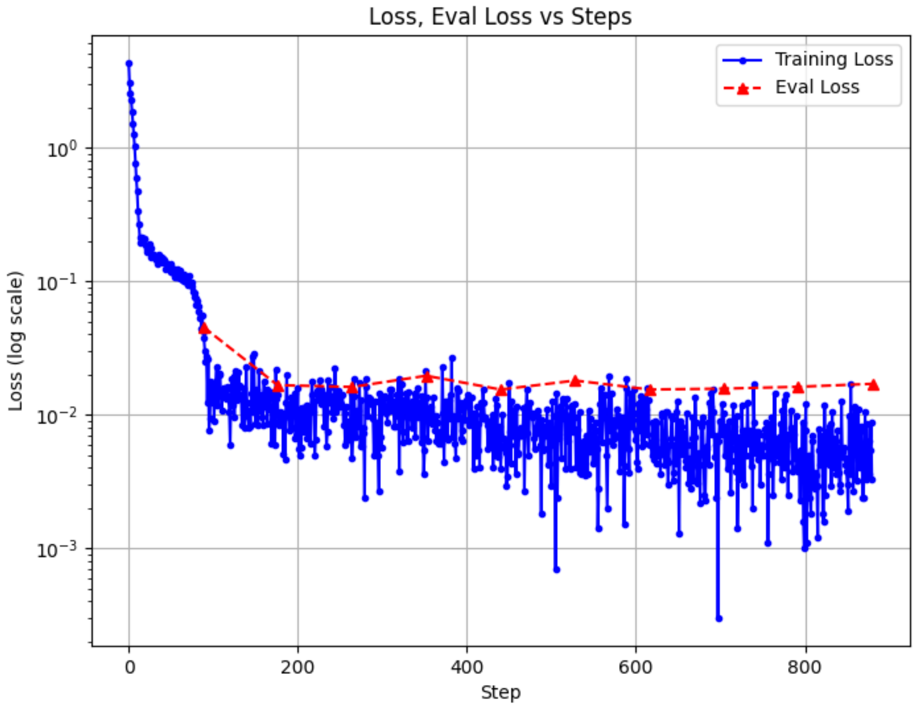


Figure 7: SFT training loss of 10 epoch in toy example. In the following experiment we chose checkpoint-880 as the baseline model.

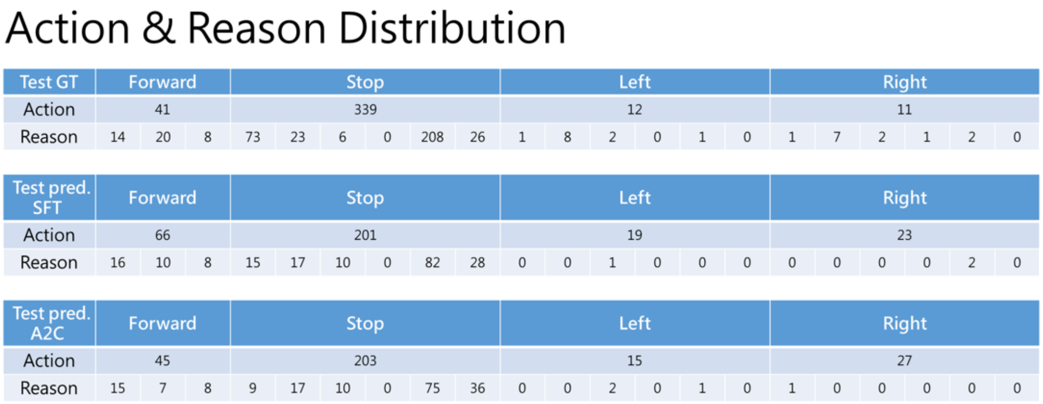
- The use of a general-purpose pretrained CLIP model (`openai/clip-vit-base-patch32`) without task-specific fine-tuning may limit its ability to provide meaningful semantic feedback in the driving context.
- The limited size of the training data may lead the model to produce out-of-domain or unseen responses, especially beyond the predefined sets of four actions and twenty-one justifications.
- The RL pipeline introduces instability via the additional MLP-based critic network, which may hinder effective gradient estimation.
- The action-reasoning reward signal $\mathcal{R}^2$, computed by an external LLM, is binary (0 or 1), providing sparse and coarse feedback that may not be sufficient to guide nuanced policy updates.

Future Work Although these results are derived from a simplified toy setting, our RL model demonstrates promising trends, such as improved output distribution alignment and action selection accuracy, even when trained purely on unlabeled data. This suggests strong potential for further improvements when scaling to the full dataset, especially with better reward signal design and the exploration of more stable, potentially critic-free, policy optimization strategies.

B CLIP Fine-Tuning

B.1 Domain-Specific Challenges

While pre-trained CLIP models provide a powerful and versatile foundation for various vision-language tasks, their direct application within specialized domains like autonomous driving often encounters significant limitations. Preliminary experiments (see Appendix A) and general understanding of such models highlight several key challenges that necessitate a dedicated fine-tuning



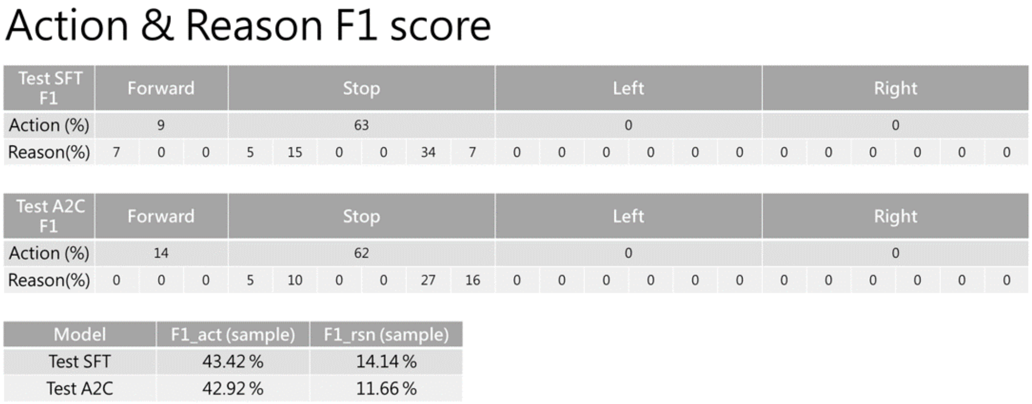
Action & Reason Distribution

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 8: Actions and justifications' **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 9: Actions and justifications' **F1 score** in toy example test dataset. The number in the form means the F1 score percentage of each action or reason.

process. For example, the specific visual lexicon, contextual nuances, and fine-grained details critical in autonomous driving scenarios (e.g., subtle differences in pedestrian posture indicating intent, specific configurations of traffic signals, varied road markings, complex multi-agent interactions) may be underrepresented or abstracted differently in the general pre-training data. For instance, while CLIP might understand the general concept of a "car" or "road," it may lack the specialized sensitivity to judge whether an action like "yield to oncoming traffic" is appropriate given a very specific configuration of vehicles at an unmarked intersection. The model's understanding of an action like "['stop']" might be generic, lacking the refined ability to link it to precise visual cues like a stop sign partially obscured by foliage versus a red traffic light.

B.1.1 Absence of Explicit Negative Examples for the Target Task:

As CLIP stands for "Contrastive" Language-Image Pre-Training, during its standard pre-training, CLIP learns by contrasting positive pairs against other (randomly sampled or in-batch) negative pairs. Its primary pre-training objective is to maximize the cosine similarity between embeddings of corresponding image-text pairs while minimizing it for non-corresponding pairs within a batch (contrastive learning). However, it is not explicitly trained to differentiate between a correct driving action and a subtly incorrect but plausible driving action for the "exact same visual input". For example, for an image where the correct action is "proceed with caution," the base model might not have been trained to assign a significantly lower score to the action "accelerate" for that same image compared to a completely unrelated action like "park." Fine-tuning with carefully constructed negative examples (e.g., same image, wrong action) is essential to teach the model these fine-grained distinctions crucial for a judgment task.

B.2 Generation of Positive and Negative Training Pairs

In order to fine-tune CLIP, we take the complete dataset of BDDOIA [Xu et al., 2020] and train the CLIP model as a binary classifier for judging image-action similarity, it was necessary to construct a dataset of both positive (matching) and negative (non-matching) pairs.

- **Positive Pairs (Label = 1):** For each of the cleaned entries, a positive pair was created by directly associating the image (identified by file_name) with its corresponding ground-truth action from the JSON annotation. These pairs represent semantically correct and contextually appropriate image-action associations.
- **Negative Pairs (Label = 0):** The generation of negative samples is a critical step, as the original dataset primarily provides examples of correct actions. Without explicit negative examples, the model might learn a trivial mapping or fail to distinguish subtle incorrect pairings. Negative samples were generated using a "same image, different action" strategy:
    1. For each image in the dataset, its true associated action/reason was identified.
    2. A different action/reason was then randomly selected from the complete pool of unique actions/reasons present across the entire dataset subset.
    3. This randomly selected (incorrect) action/reason was paired with the original image to form a negative sample.

  This strategy ensures that the model learns to pay close attention to the specific textual content of the action in relation to the visual details of the image, rather than just learning broad image categories.

B.3 Methodology

We fine-tuned the pre-trained `openai/clip-vit-base-patch32` model using a custom image-text dataset. The data was sourced from a JSON file, containing image filenames and corresponding textual reasons, with images stored in a designated folder.

Preprocessing . Entries with missing image filenames or reasons were removed. For each remaining sample, we generated:

- **Positive pairs** ($L = 1$): the image and its correct reason.
- **Negative pairs** ($L = 0$): the image and a randomly chosen incorrect reason from the set of all unique reasons.

Stratified sampling was used to ensure class balance.

We split the dataset deterministically using a fixed random seed ($s = 42$) into:

- Training: 1,400 samples
- Validation: 200 samples
- Test: 400 samples

Data Augmentation. For training images, we applied `RandomResizedCrop`, `RandomHorizontalFlip`, and `ColorJitter`. Validation and test images used `Resize` and `CenterCrop` to ensure the robustness of the model. All images were normalized using CLIP's precomputed statistics. Textual reasons were tokenized using `CLIPTokenizerFast` with padding and truncation to a maximum length $L_{\text{max}} = 77$.

Model and Training. The image (I_f) and text (T_f) features were extracted via CLIP encoders and L2-normalized. Their dot product was scaled by CLIP's learned logit scale parameter λ_{logit} to compute similarity logits:

$$S_{\text{logits}} = (I_f \cdot T_f) \times \exp(\lambda_{\text{logit}})$$

Binary classification was trained using the `BCEWithLogitsLoss` loss function. We optimized with the AdamW optimizer (learning rate 2×10^{-5}, weight decay 0.01), applying gradient clipping with max norm 1.0.

A `ReduceLROnPlateau` scheduler reduced the learning rate by a factor of 0.2 if validation AUROC did not improve for 3 epochs. Early stopping halted training if no improvement was seen for 7 consecutive epochs. The best model (by validation AUROC) was saved. Batch size was set to 32.

B.4 Results and Discussion

As mentioned previously, since the CLIP is used for generating reward values during the training process in the later stages (e.g., Reinforcement Fine-Tuning such as RFT and RLOO), it is crucial that the CLIP model be properly fine-tuned to adapt to the specific image-text pairing tasks in autonomous driving scenarios. Based on this need, we selected the model fine-tuned on image-reason pairs for subsequent use in the reinforcement learning pipeline. (Section 5.2)

The fine-tuning progression for the image-reason model, illustrated in Fig 10, showed a consistent decrease in training loss from an initial value of approximately 1.45 to below 0.2 by the 11th epoch. Validation AUROC reached 0.90 within the first 5 epochs and stabilized between 0.90 and 0.91. The learning rate decreased after epoch 16, and training terminated after 19 epochs due to early stopping.

Validation Performance. On the validation set (200 samples), the best model achieved a ROC AUC of 0.9118. The confusion matrix at threshold $\tau = 0.5$ (Fig 11a) yielded:

- True Positives: 91
- True Negatives: 76
- False Positives: 24
- False Negatives: 9

From these values:

- Accuracy = 0.8350
- Precision (Class 1) = 0.7913
- Recall (Class 1) = 0.9100
- F1-score (Class 1) = 0.8466

The ROC curve is presented in Fig 11b.

Test Performance. On the test set (407 samples), the model achieved a ROC AUC of 0.9581, demonstrating strong generalization. The test confusion matrix (threshold $\tau = 0.5$, Fig 11c) showed:

- True Positives: 188
- True Negatives: 162
- False Positives: 42
- False Negatives: 15

From these, we computed:

- Accuracy = 0.8599
- Precision (Class 1) = 0.8174
- Recall (Class 1) = 0.9261
- F1-score (Class 1) = 0.8683

The ROC curve is shown in Fig 11d.

C Poster Presentation

We have made slight modifications to our final poster to incorporate the relevant references (Fig. 12).

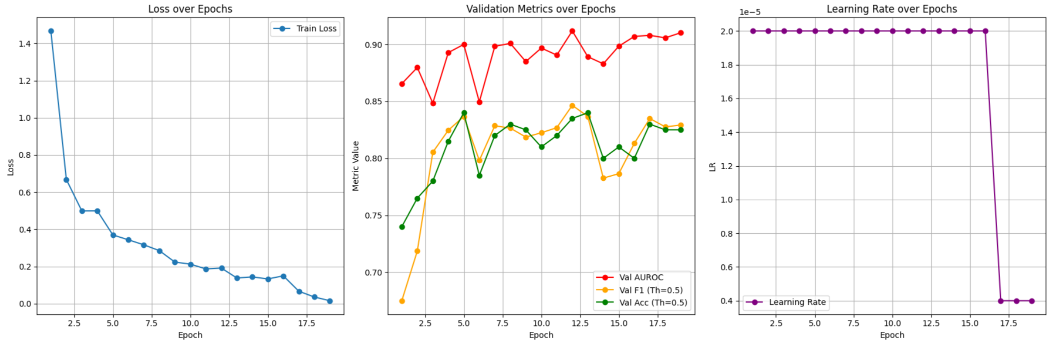
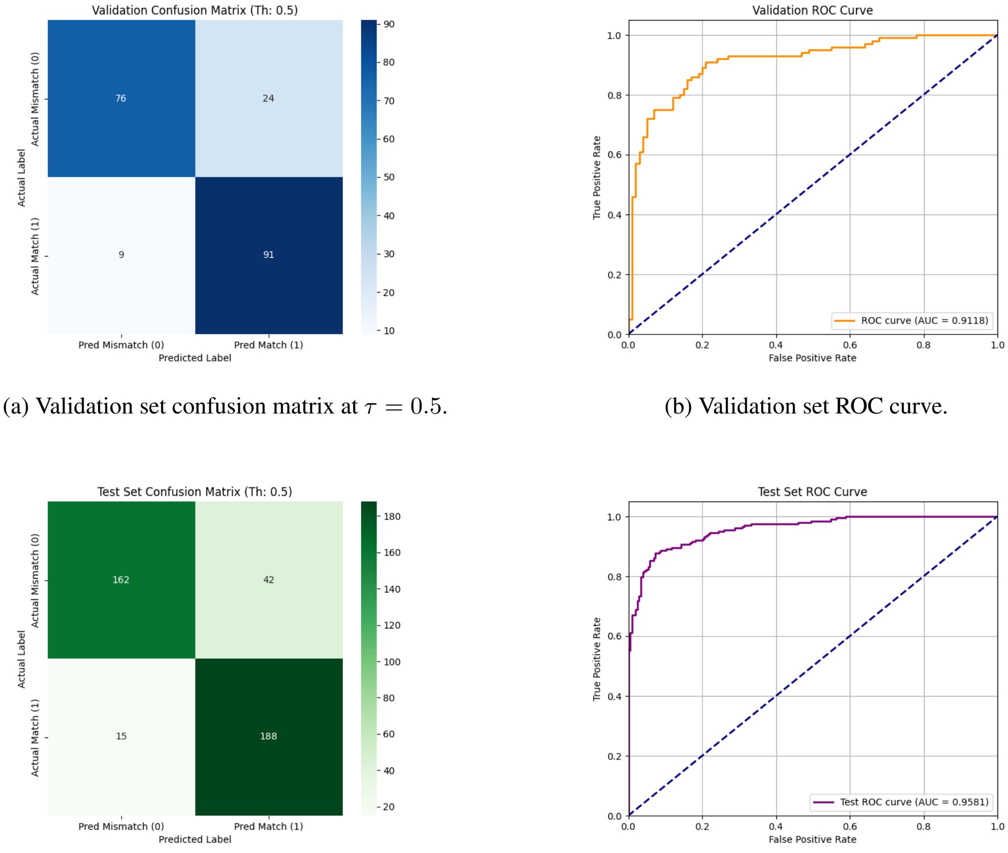


Figure 10: Training and validation loss/AUROC over epochs.



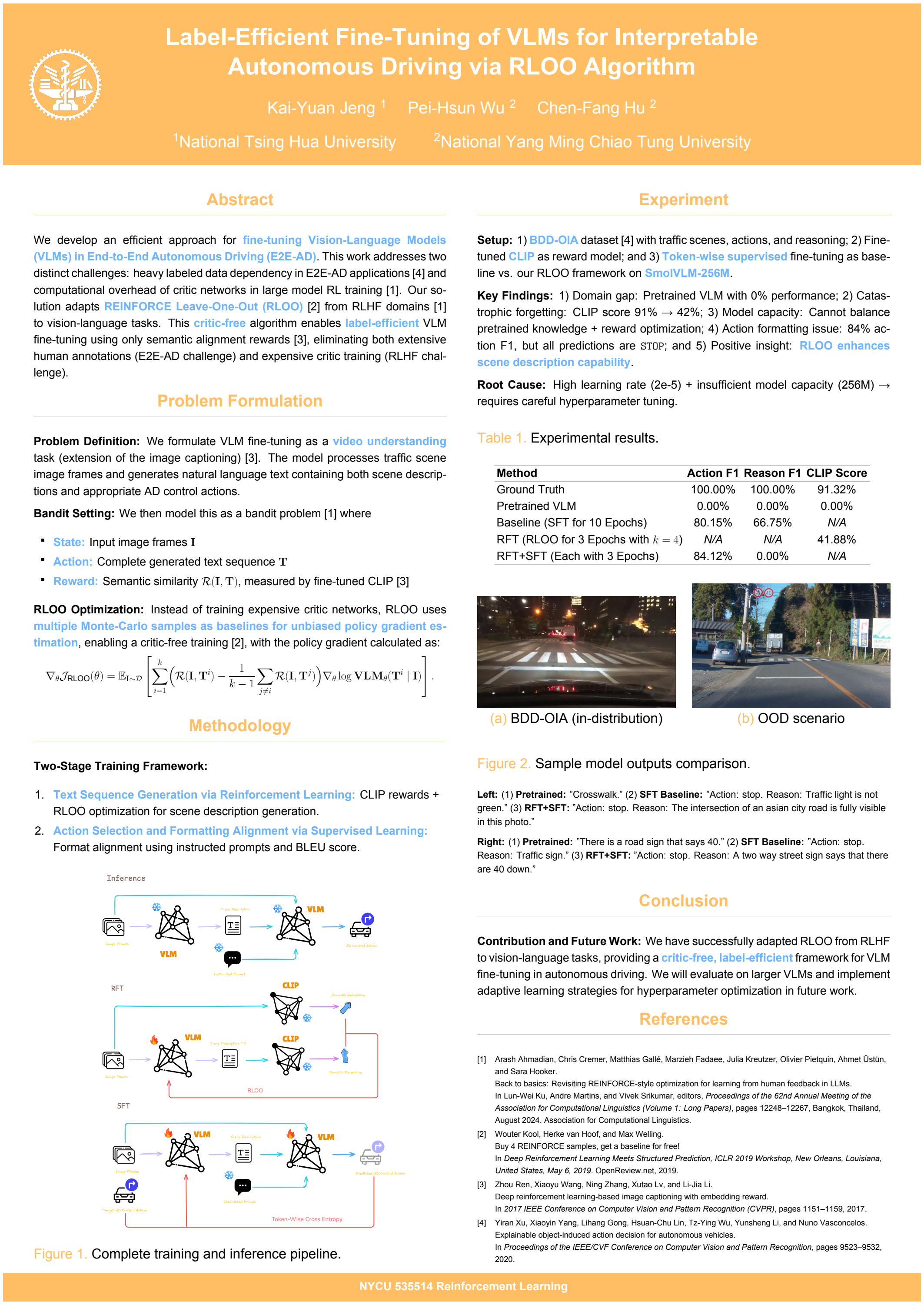
(a) Validation set confusion matrix at $\tau = 0.5$.

(b) Validation set ROC curve.

(c) Test set confusion matrix at $\tau = 0.5$.

(d) Test set ROC curve.

Figure 11: Confusion matrices and ROC curves for validation and test sets.



Label-Efficient Fine-Tuning of VLMs for Interpretable Autonomous Driving via RLOO Algorithm

Kai-Yuan Jeng [1] Pei-Hsun Wu [2] Chen-Fang Hu [2]

[1]National Tsing Hua University [2]National Yang Ming Chiao Tung University

Abstract

We develop an efficient approach for **fine-tuning Vision-Language Models (VLMs) in End-to-End Autonomous Driving (E2E-AD)**. This work addresses two distinct challenges: heavy labeled data dependency in E2E-AD applications [4] and computational overhead of critic networks in large model RL training [1]. Our solution adapts **REINFORCE Leave-One-Out (RLOO)** [2] from RLHF domains [1] to vision-language tasks. This **critic-free** algorithm enables **label-efficient** VLM fine-tuning using only semantic alignment rewards [3], eliminating both extensive human annotations (E2E-AD challenge) and expensive critic training (RLHF challenge).

Problem Formulation

Problem Definition: We formulate VLM fine-tuning as a **video understanding** task (extension of the image captioning) [3]. The model processes traffic scene image frames and generates natural language text containing both scene descriptions and appropriate AD control actions.

Bandit Setting: We then model this as a bandit problem [1] where

- **State:** Input image frames $\mathbf{I}$
- **Action:** Complete generated text sequence $\mathbf{T}$
- **Reward:** Semantic similarity $\mathcal{R}(\mathbf{I}, \mathbf{T})$, measured by fine-tuned CLIP [3]

RLOO Optimization: Instead of training expensive critic networks, RLOO uses **multiple Monte-Carlo samples as baselines for unbiased policy gradient estimation**, enabling a critic-free training [2], with the policy gradient calculated as:

$$\nabla_\theta \mathcal{J}_{\text{RLOO}}(\theta) = \mathbb{E}_{\mathbf{I}\sim\mathcal{D}}\left[\sum_{i=1}^{k}\left(\mathcal{R}(\mathbf{I}, \mathbf{T}^i) - \frac{1}{k-1}\sum_{j\neq i}\mathcal{R}(\mathbf{I}, \mathbf{T}^j)\right)\nabla_\theta \log \mathbf{VLM}_\theta(\mathbf{T}^i \mid \mathbf{I})\right].$$

Methodology

Two-Stage Training Framework:

1. **Text Sequence Generation via Reinforcement Learning:** CLIP rewards + RLOO optimization for scene description generation.
2. **Action Selection and Formatting Alignment via Supervised Learning:** Format alignment using instructed prompts and BLEU score.

Figure 1. Complete training and inference pipeline.

Experiment

Setup: 1) **BDD-OIA** dataset [4] with traffic scenes, actions, and reasoning; 2) Fine-tuned **CLIP** as reward model; and 3) **Token-wise supervised** fine-tuning as baseline vs. our RLOO framework on **SmolVLM-256M**.

Key Findings: 1) Domain gap: Pretrained VLM with 0% performance; 2) Catastrophic forgetting: CLIP score 91% → 42%; 3) Model capacity: Cannot balance pretrained knowledge + reward optimization; 4) Action formatting issue: 84% action F1, but all predictions are `STOP`; and 5) Positive insight: **RLOO enhances scene description capability**.

Root Cause: High learning rate (2e-5) + insufficient model capacity (256M) → requires careful hyperparameter tuning.

Table 1. Experimental results.

Method	Action F1	Reason F1	CLIP Score
Ground Truth	100.00%	100.00%	91.32%
Pretrained VLM	0.00%	0.00%	0.00%
Baseline (SFT for 10 Epochs)	80.15%	66.75%	*N/A*
RFT (RLOO for 3 Epochs with $k = 4$)	*N/A*	*N/A*	41.88%
RFT+SFT (Each with 3 Epochs)	84.12%	0.00%	*N/A*

(a) BDD-OIA (in-distribution) (b) OOD scenario

Figure 2. Sample model outputs comparison.

Left: (1) **Pretrained:** "Crosswalk." (2) **SFT Baseline:** "Action: stop. Reason: Traffic light is not green." (3) **RFT+SFT:** "Action: stop. Reason: The intersection of an asian city road is fully visible in this photo."

Right: (1) **Pretrained:** "There is a road sign that says 40." (2) **SFT Baseline:** "Action: stop. Reason: Traffic sign." (3) **RFT+SFT:** "Action: stop. Reason: A two way street sign says that there are 40 down."

Conclusion

Contribution and Future Work: We have successfully adapted RLOO from RLHF to vision-language tasks, providing a **critic-free, label-efficient** framework for VLM fine-tuning in autonomous driving. We will evaluate on larger VLMs and implement adaptive learning strategies for hyperparameter optimization in future work.

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker.
Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs.
In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[2] Wouter Kool, Herke van Hoof, and Max Welling.
Buy 4 REINFORCE samples, get a baseline for free!
In *Deep Reinforcement Learning Meets Structured Prediction, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019.

[3] 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.

[4] 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.

NYCU 535514 Reinforcement Learning

Figure 12: Final Poster Presentation