# ARDRIVE: ADAPTIVE RETRIEVAL-AUGMENTED GENERATION FOR AUTONOMOUS DRIVING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Retrieval-augmented generation (RAG) has recently been introduced to enhance the reasoning ability of vision-language models (VLMs) for autonomous driving (AD) by incorporating external driving knowledge. However, existing RAG-based AD systems typically perform retrieval indiscriminately, assuming that external knowledge is always beneficial. This assumption does not always hold. RAG can be redundant in simple scenarios and may even introduce misleading information when retrieval is imperfect, leading to extra latency and hallucinated reasoning. To address this limitation, we propose ARDrive, a RAG-aware driving framework that formulates retrieval usage as a learnable, context-dependent decision rather than a fixed inference procedure. ARDrive introduces a learnable `[RAG]` token as a dynamic retrieval gate and trains it through a two-stage RAG-aware pipeline. First, supervised fine-tuning (SFT) establishes initial retrieval awareness using performance-gain-driven pseudo labels generated by comparing RAG and No-RAG reasoning outcomes with an LLM-based semantic scorer. Second, RAG-aware reinforcement learning (RL) further refines the retrieval policy by directly optimizing the trade-off between reasoning accuracy and retrieval cost. Experiments on public AD benchmarks show that ARDrive reduces reasoning latency by 14% and RAG activation by 34%, while improving driving reasoning performance over conventional always-RAG baselines.

## 1 INTRODUCTION

Recent progress in autonomous driving has shifted attention from perception-centric frameworks (Chitta et al., 2023; Hu et al., 2023; Yu et al., 2025; 2024) toward reasoning-oriented agents (Xu et al., 2024; Sima et al., 2024) that interpret complex scenes and make context-aware decisions. With the emergence of large vision-language models (Lin et al., 2024; Chen et al., 2024; Zhai et al., 2023), driving agents are increasingly equipped with multimodal reasoning capabilities that connect visual perception with linguistic understanding. Despite these advances, their reasoning ability remains constrained by the knowledge acquired during training, resulting in limited generalization when encountering previously unseen driving situations (Ding et al., 2023).

To address this problem, recent AD studies (Zhang et al., 2025; Yuan et al., 2024; Ding et al., 2025) have adopted retrieval-augmented generation (RAG) (Lewis et al., 2020) to incorporate external knowledge into model reasoning. For each driving scenario, these methods retrieve relevant external information (*e.g.*, expert demonstrations, previous experiences, or scene priors) based on the current observation and integrate the retrieved content into the reasoning process to enhance generation.

Despite their promising results, existing RAG-based driving frameworks typically retrieve external knowledge at every reasoning step (Zhang et al., 2025; Yuan et al., 2024). This always-RAG paradigm introduces two limitations. First, retrieval can be redundant in simple driving scenarios where the visual observation and ego-state information are already sufficient for decision making (Jeong et al., 2024; Zheng et al., 2025), leading to unnecessary computational overhead and inference latency. Second, retrieved knowledge is not guaranteed to be reliable, since imperfect retrieval engines or databases may return irrelevant or misleading experiences (Zhang et al., 2025; Yuan et al., 2024), which can further induce hallucinated reasoning. As shown in Fig. 1(a), an example from the BDD-X dataset (Kim et al., 2018) illustrates that reasoning without RAG in this



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

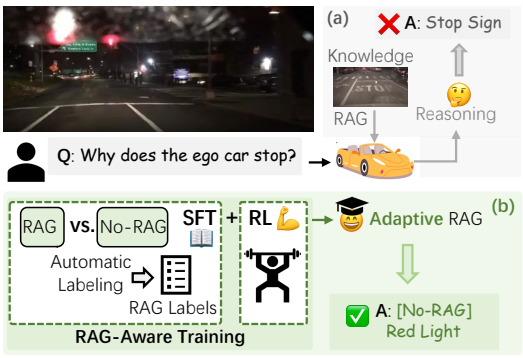


Figure 1: Illustration of our motivation and framework. (a) In autonomous driving, existing RAG-based frameworks perform retrieval at every reasoning step. However, RAG is not always necessary. Unnecessary RAG results in excessive computational overhead and may even mislead the reasoning process. (b) ARDrive learns an adaptive RAG policy through RAG-aware training and dynamically decides when RAG is necessary, achieving a better accuracy-efficiency trade-off.

case not only improves efficiency but also reduces hallucinations introduced by imperfect external knowledge.

To overcome these limitations, we propose ARDrive, a RAG-aware vision-language framework for autonomous driving that learns to decide when retrieval is necessary (Fig. 1(b)). Instead of treating RAG as a fixed inference operation, ARDrive formulates retrieval usage as a context-dependent decision conditioned on the current driving scene, ego-state, and task prompt. Specifically, ARDrive introduces a learnable `[RAG]` token as a dynamic retrieval gate, allowing the model to jointly predict the driving response and whether external knowledge should be invoked. Learning such a retrieval policy is non-trivial because existing AD benchmarks do not provide annotations indicating whether RAG should be used. Moreover, unlike adaptive retrieval in text-only tasks (Asai et al., 2024; Su et al., 2024; Wang et al., 2025), retrieval necessity in multimodal driving cannot be determined solely from the input text. It depends on whether retrieved knowledge actually improves the final reasoning outcome. To address this challenge, we design a two-stage RAG-aware training strategy. In the SFT stage, we construct retrieval supervision through a performance-gain-driven automatic labeling mechanism, where RAG and No-RAG outputs are compared against the ground truth using an LLM-based semantic scorer (Qwen-Max (Yang et al., 2025)). In the RL stage, we further refine the retrieval policy with an explicit retrieval cost, directly optimizing the trade-off between reasoning performance and inference efficiency. Overall, ARDrive provides a principled learning framework for adaptive RAG in autonomous driving.

We summarize our main contributions as follows:

- We make the first attempt to study adaptive retrieval-augmented generation (RAG) for autonomous driving. Different from existing always-RAG frameworks, ARDrive formulates retrieval usage as a learnable, context-dependent decision and enables driving agents to determine when external knowledge is necessary.
- We introduce a learnable `[RAG]` token as a dynamic retrieval gate and design a two-stage RAG-aware training strategy. The SFT stage establishes initial retrieval awareness, while the RL stage further refines the policy by optimizing task performance under retrieval cost.
- We develop a performance-gain-driven automatic labeling mechanism to provide supervision for retrieval necessity without manual RAG annotations. The mechanism compares RAG and No-RAG reasoning outcomes with an LLM-based semantic scorer, which is also used as the reward signal during RL.
- Extensive experiments on public benchmarks (BDD-X (Kim et al., 2018) and DriveLM (Sima et al., 2024)) show that ARDrive reduces reasoning latency by 14% and RAG usage by 34%, while improving driving reasoning performance over conventional always-RAG baselines, achieving a better accuracy-efficiency trade-off.

## 2 Related Work

**VLMs in Autonomous Driving.** The integration of vision-language models (VLMs) into autonomous driving (AD) has enabled interpretable and reasoning-driven decision making. Early end-to-end architectures such as TransFuser (Chitta et al., 2023) and UniAD (Hu et al., 2023) focused on sensor fusion and control prediction without explicit reasoning. With the emergence of large VLMs (Lin et al., 2024; Chen et al., 2024; Zhai et al., 2023), models such as DriveVLM (Tian et al., 2024) and DriveLM (Sima et al., 2024) introduced language-based reasoning for scene understanding, prediction, and planning. DriveGPT4 (Xu et al., 2024) further produced interpretable driving rationales, while DriveAgent-R1 (Zheng et al., 2025) dynamically switched between textual and visual reasoning based on scene complexity. However, most VLM-based systems still rely primarily on internal knowledge, which limits their generalization to unseen or safety-critical driving scenarios. This motivates the use of external knowledge through retrieval-augmented generation (RAG).

**Retrieval-Augmented Generation in Autonomous Driving.** RAG (Asai et al., 2024; Jeong et al., 2024; Su et al., 2024) enhances language models by retrieving external knowledge for context-aware reasoning, and has recently been extended to autonomous driving to improve interpretability and generalization. For example, RAG-Driver (Yuan et al., 2024) aligns visual observations and control signals through in-context retrieval, while RAC3 (Wang et al., 2024) retrieves similar corner cases to mitigate hallucinations. SafeAuto (Zhang et al., 2025) employs safety-oriented memory retrieval for relevant driving priors. Despite these advances, existing RAG-based AD frameworks mainly focus on what knowledge to retrieve, while treating retrieval itself as a fixed operation at every reasoning step. Such an always-RAG design can be redundant in simple scenarios and may even degrade reasoning when the retrieved knowledge is irrelevant. These limitations motivate adaptive RAG mechanisms that allow driving agents to decide when retrieval is truly required.

## 3 Method

### 3.1 Problem Formulation and Framework

In the vision-language reasoning task for autonomous driving, each driving sequence is represented as a set of multimodal observations defined as

$$\mathcal{D} = \{(V_t, s_t, p_t, y_t)\}_{t=1}^{T}, \tag{1}$$

where $V_t$ denotes the visual observations such as multi-view images or videos at time $t$, $s_t$ represents sensor and vehicle state information including speed, curvature, and acceleration, $p_t$ is a language instruction, and $y_t$ is the corresponding reasoning output.

Existing RAG-based driving frameworks (Zhang et al., 2025; Yuan et al., 2024; Ding et al., 2025; Wang et al., 2024; Zuo et al., 2025) assume that at each reasoning step $t$, the model retrieves a set of external memory items $\mathcal{M}_t = \{m_i\}_{i=1}^{k}$, such as relevant driving scenes or expert demonstrations, to enhance response generation:

$$y_t = f_\theta(V_t, s_t, p_t, \mathcal{M}_t), \tag{2}$$

where $f_\theta$ denotes the reasoning function. Although this always-RAG formulation has shown promising results (Yuan et al., 2024; Zhang et al., 2025), it relies on a fixed retrieval assumption that external memory should always be invoked. This assumption increases inference cost and may even mislead the model when the retrieved knowledge is irrelevant or imperfect (Asai et al., 2024).

In contrast, ARDrive treats retrieval usage as a context-dependent decision conditioned on the current multimodal driving state, as illustrated in Fig. 2. Formally, we introduce a binary RAG decision variable $r_t \in \{0, 1\}$, where $r_t = 1$ indicates that RAG should be performed at step $t$ and $r_t = 0$ otherwise. The reasoning process is defined as

$$y_t = \begin{cases} f_\theta(V_t, s_t, p_t, \mathcal{M}_t), & \text{if } r_t = 1, \\ f_\theta(V_t, s_t, p_t), & \text{if } r_t = 0. \end{cases} \tag{3}$$

The objective is to learn an adaptive RAG policy $\pi_\phi(r_t \mid V_t, s_t, p_t)$ that maximizes reasoning quality while minimizing unnecessary retrieval cost. In other words, the policy should trigger RAG only

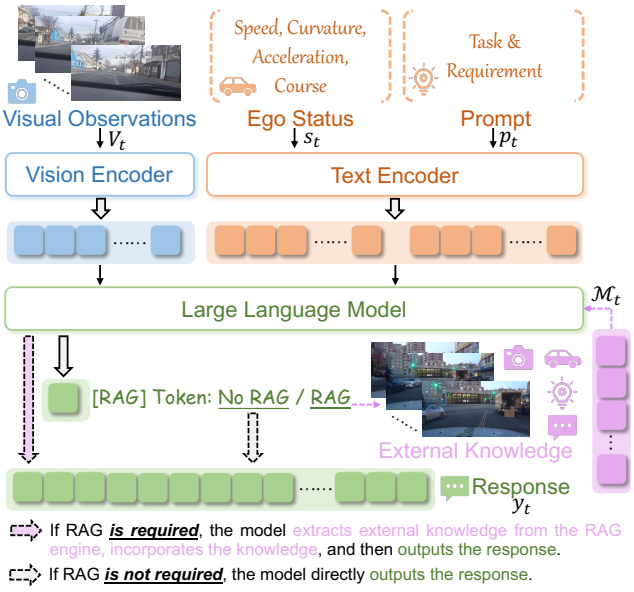


Figure 2: Overall framework of ARDrive. Given multimodal inputs (*e.g.*, visual observations, ego-vehicle status, and textual task prompts), ARDrive first encodes them through a vision and a text encoder. A `[RAG]` token is injected into the large language model to determine whether retrieval should be performed. If RAG is required, the model queries external knowledge and incorporates it into the generation context. Otherwise, the model directly produces the driving reasoning and response.

when the expected benefit of external knowledge outweighs its inference cost. This formulation differs from previous RAG-based driving systems (Zhang et al., 2025; Yuan et al., 2024; Ding et al., 2025; Wang et al., 2024; Zuo et al., 2025), which treat RAG as a fixed operation at every reasoning step.

We realize the adaptive RAG strategy through a learnable `[RAG]` token. This token serves as a dynamic retrieval gate that predicts whether external knowledge should be invoked for the current context. When the `[RAG]` token indicates that retrieval is necessary, the model queries an RAG engine (Zhang et al., 2025) to obtain relevant external knowledge and uses it for response generation. Otherwise, the model proceeds with in-context reasoning directly. The central challenge is therefore how to train this token to capture the actual utility of retrieval, rather than merely imitating a fixed retrieval pattern.

### 3.2 RAG-Aware Training Strategy

To train the RAG-aware ability of the `[RAG]` token, we adopt a two-stage RAG-aware training strategy composed of a supervised fine-tuning (SFT) stage and a reinforcement learning (RL) stage. The SFT stage aims to establish initial RAG awareness by providing the model with supervision signals that indicate when RAG should be performed, enabling the learnable `[RAG]` token to acquire a preliminary decision capability. However, the SFT objective is constrained by sample-level supervision and imperfect labels, which limits the model's ability to fully develop accurate RAG decision-making (Stiennon et al., 2020; Ouyang et al., 2022). To overcome this limitation, the subsequent RL stage refines the RAG policy and further unlocks the potential of the model (Zheng et al., 2025) by directly optimizing the balance between reasoning efficiency and accuracy.

#### 3.2.1 RAG-Aware Supervised Fine-Tuning (SFT)

The supervised fine-tuning (SFT) stage provides a cold start and equips the model with a fundamental understanding of the output format and basic RAG awareness (Zheng et al., 2025; Luo et al., 2025), ensuring that subsequent RL explores within a meaningful policy space rather than perform-

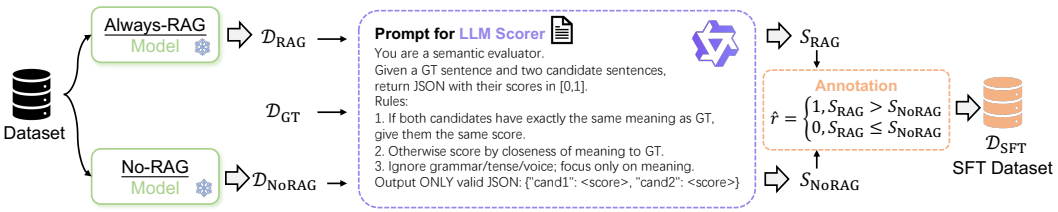


Figure 3: Pipeline of our automatic labeling mechanism for SFT data construction, for example on driving explanation tasks (Kim et al., 2018). We first run an Always-RAG model and a No-RAG model on the same dataset to obtain two sets of reasoning outputs ($\mathcal{D}_{\text{RAG}}$ and $\mathcal{D}_{\text{NoRAG}}$). A LLM-based semantic scorer (Qwen-Max (Yang et al., 2025)) then compares each output with the ground-truth reasoning ($\mathcal{D}_{\text{GT}}$) and assigns normalized scores $S_{\text{RAG}}$ and $S_{\text{NoRAG}}$. Samples for which the RAG-enhanced output achieves a higher score are labeled as requiring RAG ($\hat{r} = 1$), while others are labeled as not requiring RAG ($\hat{r} = 0$). The resulting labeled set $\mathcal{D}_{\text{SFT}}$ is used for SFT.

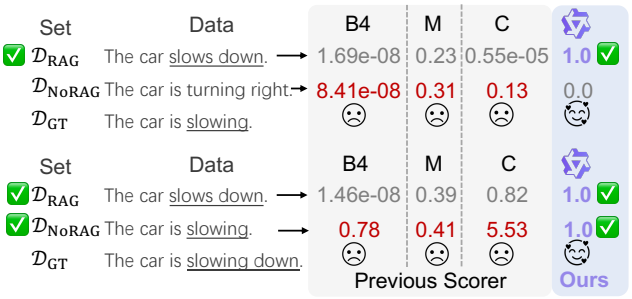


Figure 4: Comparison between traditional n-gram metrics (BLEU (B4), METEOR (M), CIDEr (C)) used in previous works (Yuan et al., 2024; Zhang et al., 2025) and our LLM-based semantic scorer (Qwen-Max (Yang et al., 2025)). In certain cases, our scorer can more accurately identify semantically correct outputs and assign consistent scores to semantically equivalent sentences.

ing random actions (Rafailov et al., 2023; Zhang et al., 2024a). Therefore, we introduce an SFT stage to stabilize the training process and initialize the RAG decision behavior.

**Data Preparation.** Current autonomous driving benchmarks (Kim et al., 2018; Sima et al., 2024) lack annotations that indicate when RAG is needed, making direct supervision infeasible. Unlike natural language processing tasks (Asai et al., 2024; Su et al., 2024; Wang et al., 2025) that can determine the inference mode based on textual analysis, the necessity of RAG in multimodal driving data depends on the potential performance gain obtained from RAG, which cannot be intuitively inferred from the contextual information in the data. This motivates us to develop a labeling mechanism that automatically generates supervision signals by comparing the outputs of No-RAG and Always-RAG models with the ground-truth reasoning.

As illustrated in Fig. 3, we first run two pretrained models on the original driving dataset, including an Always-RAG model that performs RAG at every reasoning step and a No-RAG model that relies solely on in-context information. These two models share the same baseline (Video-LLaVA-7B (Lin et al., 2024)) and differ only in whether RAG is enabled, so any performance gap can be attributed solely to the use of RAG. For each sample, both models generate reasoning results to construct two output sets, denoted as $\mathcal{D}_{\text{RAG}}$ and $\mathcal{D}_{\text{NoRAG}}$, respectively. By measuring which output aligns better to the ground-truth reasoning, we can determine whether RAG is necessary for the current sample. Therefore, a reliable evaluator is required to quantify the performance gap between the model outputs and the ground-truth reasoning. For language-based driving explanation tasks (Kim et al., 2018), previous approaches (Yuan et al., 2024; Zhang et al., 2025) typically rely on n-gram based metrics (*e.g.*, BLEU (Kishore Papineni & Zhu, 2002), METEOR (Banerjee & Lavie, 2004), and CIDEr (Ramakrishna Vedantam & Parikh, 2015)). While these metrics work reasonably well on average (Yuan et al., 2024; Zhang et al., 2025), we observe that in certain cases they may fail to capture semantic correctness, as illustrated by examples in Fig. 4. To reduce such evaluation noise

and obtain more reliable supervision, we employ an LLM-based semantic scorer (Qwen-Max (Yang et al., 2025)) motivated by recent achievements of large language models (LLMs) in semantic comprehension (Touvron et al., 2023; Yang et al., 2025; Asai et al., 2024) to provide semantically faithful scoring, as shown in Fig. 3 and Fig. 4.

We then use the proposed scorer to compute normalized scores $S_{\text{RAG}}$ and $S_{\text{NoRAG}}$ in $[0, 1]$ for each pair of outputs, measuring their semantic alignment with the ground-truth set $\mathcal{D}_{\text{GT}}$. The RAG label $\hat{r}$ is assigned as

$$\hat{r} = \begin{cases} 1, & \text{if } S_{\text{RAG}} > S_{\text{NoRAG}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This means that a sample is labeled as requiring RAG only when the RAG-enhanced output receives a higher score than the No-RAG output. The labeling scheme is consistent with our focus on the accuracy–efficiency trade-off, since a sample is considered to require RAG only when RAG brings a performance gain over the No-RAG output. In addition, for numerical outputs, we convert the root mean squared error (RMSE) into the evaluation score. Finally, we produce a pseudo-labeled dataset $\mathcal{D}_{\text{SFT}}$, which serves as the supervision for fine-tuning the `[RAG]` token introduced in Sec. 3.1.

**Annotation Check.** We manually inspect a random subset of annotations. Since the labels are generated automatically by comparing RAG and No-RAG outputs, they can be noisy due to scene ambiguity and biases in the pretrained models. In practice, we review triplets of {ground truth, RAG output, No-RAG output} and verify that the assigned scores are reasonable (*e.g.*, the output that is closer to the ground truth receives the higher score). A quantitative audit on the subset shows that the automatically assigned labels agree with majority human judgment in 99% of action samples and 92% of justification samples. This confirms that most labels align with human judgment and increases the reliability of the supervision signal used to train the `[RAG]` token in the SFT stage.

Finally, the resulting dataset $\mathcal{D}_{\text{SFT}}$ is used to supervise the training of ARDrive. During this stage, the learnable `[RAG]` token is optimized to predict whether RAG should be activated, while the remaining tokens are trained to generate coherent reasoning conditioned on the selected RAG mode.

### 3.2.2 RAG-Aware Reinforcement Learning (RL)

While SFT allows the model to imitate RAG behaviors from offline data, it is fundamentally constrained by the quality of its supervision (Stiennon et al., 2020). Since the necessity of RAG is inferred from automatically generated pseudo labels, some label noise is unavoidable. This limits the accuracy ceiling of the learned RAG decisions and prevents SFT from fully capturing the optimal RAG policy. Therefore, we introduce an RAG-aware RL stage that refines the decision policy using outcome-based feedback and directly optimizes the balance between reasoning efficiency and accuracy (Ouyang et al., 2022).

**RAG-Aware GRPO.** The core of our RAG-aware GRPO is to endow the `[RAG]` token with the ability to determine when RAG is necessary. Given a multimodal driving input $x_t = (V_t, s_t, p_t)$, we begin by sampling a set of $n$ candidate outputs $G_t = \{o_t^{(i)}\}_{i=1}^{n}$ from the old policy $\pi_{\text{old}}$. The current policy $\pi_\phi$ is then optimized using the reward signals associated with these candidates. For each output $o_t^{(i)}$, we compute a reward $R_t^{(i)}$ by comparing its prediction $y_t^{(i)}$ against the ground-truth reasoning $y_t^{\text{GT}}$. For semantic tasks (driving explanation (Kim et al., 2018)), our LLM-based scorer (Sec. 3.2.1) produces a quality score $S_t^{(i)} \in [0, 1]$. For numeric outputs, we convert the prediction error (*e.g.*, RMSE) into a bounded score using an exponential kernel, which controls the sensitivity to numeric deviations. Therefore, the reward definition becomes

$$R_t^{(i)} = \begin{cases} S_t^{(i)} - c_k r_t^{(i)}, & \text{Semantic} \\ \exp\big(-\text{RMSE}(y_t^{(i)}, y_t^{\text{GT}})\big) - c_k r_t^{(i)}, & \text{Numeric,} \end{cases} \tag{5}$$

where $r_t^{(i)}$ is an indicator of whether RAG is triggered in the $i$-th candidate and $c_k$ is a scalar cost coefficient. With this design, RAG is encouraged only when its performance gain is large enough to outweigh the RAG cost, which encodes the desired accuracy–efficiency trade-off.

We then convert the per-candidate rewards into group-relative advantages $A_t^{(i)}$. For each candidate $o_t^{(i)}$, both the current policy $\pi_\phi$ and the old policy $\pi_{\text{old}}$ evaluate its log-probability, and the PPO

importance ratio (Schulman et al., 2017) is computed as

$$\rho_t^{(i)} = \frac{\pi_\phi(o_t^{(i)} \mid x_t)}{\pi_{\text{old}}(o_t^{(i)} \mid x_t)}, \tag{6}$$

We adopt the clipped objective used in GRPO (Shao et al., 2024), where the ratio is truncated as

$$\tilde{\rho}_t^{(i)} = \text{clip}\big(\rho_t^{(i)},\, 1-\epsilon,\, 1+\epsilon\big), \tag{7}$$

and the final optimization objective is

$$\mathcal{J} = -\mathbb{E}_t\left[\frac{1}{n}\sum_{i=1}^{n} \min\left(\rho_t^{(i)} A_t^{(i)},\, \tilde{\rho}_t^{(i)} A_t^{(i)}\right)\right], \tag{8}$$

$$\mathcal{L}_{\text{RL}} = \mathcal{J} + \lambda\, \text{KL}(\pi_\phi(\cdot \mid x_t) \,\|\, \pi_{\text{ref}}(\cdot \mid x_t)), \tag{9}$$

where $\lambda$ controls the strength of KL regularization, constraining the updated policy from deviating excessively from the frozen reference policy $\pi_{\text{ref}}$ (Shao et al., 2024).

**How the RAG-aware GRPO works.** Our RAG-aware GRPO refines the SFT prior by directly optimizing RAG decisions based on outcome-level rewards. By generating multiple candidate outputs under the same input state and normalizing their rewards within a group, GRPO (Shao et al., 2024) highlights the relative utility of different RAG behaviors for each context, taking into account both performance gains and RAG cost. The policy is then updated to increase the likelihood of candidates whose RAG decisions achieve higher rewards, reflecting a better accuracy–efficiency trade-off. Through this mechanism, the `[RAG]` token gradually learns a context-dependent RAG policy that corrects SFT noise and aligns RAG behavior with actual task performance.

# 4 Experiment

## 4.1 Dataset

**BDD-X** (Kim et al., 2018) is an explainable autonomous driving benchmark containing 77 hours of driving videos collected under diverse road and weather conditions. We use the processed version from RAG-Driver (Yuan et al., 2024), which takes an input video and control signals from the previous seven frames to perform three tasks, *i.e.*, action prediction, justification generation, and next-frame speed/course prediction. The dataset contains 16,390 training and 2,123 test conversations, following the preprocessing protocol of prior work (Yuan et al., 2024; Zhang et al., 2025).

**DriveLM** (Sima et al., 2024), built on nuScenes (Caesar & Beijbom, 2020), uses six current-frame multi-view images and control signals, including trajectories from the previous three seconds, as input. We adopt the SafeAuto (Zhang et al., 2025) split with 3,447 training and 685 test conversations. Since test annotations for high-level reasoning are unavailable, we evaluate on the low-level motion prediction task (Jin & Liu, 2023; Sima et al., 2024; Hu et al., 2023).

## 4.2 Evaluation Metrics

**BDD-X Dataset.** For the action description and justification tasks, we follow (Yuan et al., 2024; Zhang et al., 2025) and adopt standard metrics including 4-gram BLEU (B4) (Kishore Papineni & Zhu, 2002), METEOR (M) (Banerjee & Lavie, 2004), and CIDEr (C) (Ramakrishna Vedantam & Parikh, 2015). Although these metrics can be noisy on certain examples (Fig. 4), they are widely used and effective overall (Kim et al., 2018; Yuan et al., 2024; Zhang et al., 2025), so we retain them as our evaluation metrics for fair comparison. For the control signal prediction task, we follow (Yuan et al., 2024; Zhang et al., 2025) and report the root mean square error (RMSE) for both course (degrees) and speed (m/s). We also include $A_\sigma$, which represents the prediction accuracy when binarized within a tolerance threshold $\sigma$ from the ground truth.

**DriveLM Dataset.** For the motion prediction task, we use three standard trajectory prediction metrics (Hu et al., 2023; Caesar & Beijbom, 2020), $i.e.$, average displacement error (ADE), final displacement error (FDE), and miss rate at 2 meters (MR@2m). ADE measures the mean L2 distance between the predicted and ground-truth trajectories, while FDE computes the distance at the final prediction step. MR@2m evaluates the proportion of trajectories whose final point deviates from the ground truth by more than 2 meters.

Table 1: Comparison of No-RAG baseline (Base (No-RAG)), Always-RAG baseline (Base (Always-RAG)), and our ARDrive on the BDD-X dataset (Kim et al., 2018) in terms of average inference time (s) and the ratio of using RAG (%).

| **Method** | **Average Inference Time** (s) | **The Ratio of RAG** (%) |
|---|---|---|
| Base (No-RAG) | 1.64 | 0.0 |
| Base (Always-RAG) | 3.32 | 100.0 |
| **ARDrive (Ours)** | 2.85 | 66.0 |

Table 2: Comparison of No-RAG baseline (Base (No-RAG)), Always-RAG baseline (Base (Always-RAG)), and ARDrive on the BDD-X dataset (Kim et al., 2018) and DriveLM dataset (Sima et al., 2024; Zhang et al., 2025). **Bold** and Underline indicate the best and second-best results, respectively.

| **Method** | **BDD-X** | | | | **DriveLM** | | |
|---|---|---|---|---|---|---|---|
| | **Action B4↑** | **Justification B4↑** | **Speed RMSE↓** | **Course RMSE↓** | **Motion ADE↓** | **Motion FDE↓** | **Motion MR@2m↓** |
| Base (No-RAG) | 31.7 | 7.4 | 0.63 | 3.94 | 1.12 | 2.36 | **0.43** |
| Base (Always-RAG) | 38.1 | 9.3 | 0.65 | **3.85** | 1.08 | 2.30 | 0.45 |
| **ARDrive (Ours)** | **38.9** | **10.1** | **0.62** | 4.14 | **1.03** | **2.21** | **0.43** |

### 4.3 Comparison Results

**Quantitative Results.** Table 1 compares the inference efficiency of different RAG strategies. The No-RAG baseline achieves the fastest inference since it never performs retrieval. In contrast, the Always-RAG baseline retrieves at every reasoning step, resulting in the highest latency and full RAG utilization (100%). In comparison, our ARDrive dynamically controls retrieval, activating RAG in only 66.0% of the cases in the BDD-X test set while reducing the average inference time to 2.85 s. These results demonstrate that ARDrive effectively balances accuracy and efficiency by learning when retrieval is truly necessary.

Table 2 presents the comparison among the No-RAG baseline, Always-RAG baseline, and our ARDrive on both BDD-X (Kim et al., 2018) and DriveLM (Sima et al., 2024). Compared with the No-RAG and RAG baselines, ARDrive exhibits strong performance on both language-based driving explanation and numerical prediction tasks. In addition, ARDrive improves the n-gram based metrics, indicating that our LLM scorer enhances general reasoning quality rather than overfitting to a particular evaluator. These results demonstrate that ARDrive effectively unifies high-level semantic reasoning and low-level signal prediction. Combined with the efficiency results in Table 1, ARDrive achieves a better balance between reasoning quality and efficiency, highlighting the advantage of our adaptive RAG.

Table 3 compares ARDrive with representative previous approaches on the BDD-X dataset (Kim et al., 2018). Among No-RAG models, ADAPT (Jin & Liu, 2023) and DriveGPT4 (Xu et al., 2024) obtain competitive results but lack access to external knowledge, which limits their overall performance. The Always-RAG models (RAG-Driver (Yuan et al., 2024) and SafeAuto (Zhang et al., 2025)) improve several driving explanation metrics by adopting RAG at every step. In contrast, ARDrive achieves the best or second-best performance on most metrics and clearly surpasses all previous approaches on speed prediction. For course prediction, ARDrive attains better accuracy on most metrics while remaining competitive in RMSE. Table 4 further reports results on the processed DriveLM dataset (Sima et al., 2024; Zhang et al., 2025) for the motion prediction task. ARDrive achieves the best performance across all metrics. These results jointly confirm that ARDrive is effective on both high-level and low-level tasks.

### 4.4 Ablation Study

**Effects of the two-stage training.** Table 5 shows that adding the SFT stage on top of the Base (No-RAG) model improves both action and justification scores, indicating that SFT provides effective initial RAG awareness. Introducing the RL stage further gives the best B4 and lowers speed errors, which means RL refines RAG decisions and boosts performance. Additionally, SFT improves reasoning but increases inference time due to more frequent RAG calls. RL further improves reasoning under a lower RAG usage ratio, suggesting that it refines the utility of retrieval decisions.

Table 3: Comparison of representative previous approaches (Jin & Liu, 2023; Jin et al., 2024; Xu et al., 2024; Yuan et al., 2024; Zhang et al., 2025) and ARDrive on the BDD-X dataset (Kim et al., 2018). ADAPT (Jin & Liu, 2023), Time-LLM (Jin et al., 2024), and DriveGPT4 (Xu et al., 2024) are No-RAG models, while RAG-Driver (Yuan et al., 2024) and SafeAuto (Zhang et al., 2025) are Always-RAG models. **Bold** and Underline indicate the best and second-best results, respectively.

| Method | Action | | | Justification | | | Speed | | | | | | Course | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B4↑ | C↑ | M↑ | B4↑ | C↑ | M↑ | RMSE↓ | $A_{0.1}$ ↑ | $A_{0.5}$ ↑ | $A_{1.0}$ ↑ | $A_{5.0}$ ↑ | $A_{10.0}$ ↑ | RMSE↓ | $A_{0.1}$ ↑ | $A_{0.5}$ ↑ | $A_{1.0}$ ↑ | $A_{5.0}$ ↑ | $A_{10.0}$ ↑ |
| ADAPT (Jin & Liu, 2023) | 34.6 | 248 | 30.6 | **11.4** | 103 | **15.2** | 2.68 | 11.8 | 31.8 | 47.4 | 92.8 | 95.9 | 5.87 | 54.5 | 86.4 | 91.1 | 97.4 | 98.2 |
| Time-LLM (Jin et al., 2024) | – | – | – | – | – | – | 1.17 | 21.3 | 53.1 | 74.1 | 99.7 | **99.9** | 4.10 | 65.7 | 83.5 | 89.6 | 97.6 | 98.6 |
| DriveGPT4 (Xu et al., 2024) | 30.0 | 214 | 29.8 | 9.4 | 103 | 14.6 | 1.09 | 56.9 | 77.8 | 88.0 | 99.0 | 99.6 | 4.57 | 69.2 | 79.1 | 84.5 | 95.7 | 96.7 |
| RAG-Driver (Yuan et al., 2024) | 34.3 | 261 | 30.7 | 11.1 | **109** | 14.8 | 0.69 | 51.1 | 85.5 | 94.5 | 99.8 | **99.9** | 4.48 | 74.3 | 88.7 | 93.1 | 98.3 | 99.1 |
| SafeAuto (Zhang et al., 2025) | 38.6 | **337** | **35.5** | 9.4 | 96.0 | 14.0 | 0.65 | 55.5 | 88.8 | 95.3 | 99.8 | **99.9** | **3.85** | **76.3** | **89.7** | **94.1** | 98.3 | **99.3** |
| **ARDrive (Ours)** | **38.9** | 316 | 35.0 | 10.1 | 104 | 14.6 | **0.62** | **58.0** | **89.8** | **96.2** | **99.9** | **99.9** | 4.14 | 75.5 | **89.7** | **94.1** | **98.5** | **99.3** |

Table 4: Comparison of representative previous approaches (Hu et al., 2023; Li et al., 2023; Sima et al., 2024; Zhang et al., 2025) and ARDrive on the processed DriveLM dataset (Sima et al., 2024; Zhang et al., 2025) **Bold** and Underline indicate the best and second-best results, respectively.

| Method | ADE↓ | FDE↓ | MR@2m↓ |
|---|---|---|---|
| UniAD (Hu et al., 2023) | 1.80 | – | – |
| BLIP-RT-2 (Li et al., 2023) | 2.63 | – | – |
| DriveLM-Agent (Sima et al., 2024) | 1.51 | – | – |
| SafeAuto (Zhang et al., 2025) | 1.08 | 2.30 | 0.45 |
| **ARDrive (Ours)** | **1.03** | **2.21** | **0.43** |

Table 5: Ablation study on the progressive training strategies of ARDrive on the BDD-X dataset (Kim et al., 2018). **Bold** and Underline indicate the best and second-best results, respectively.

| Method | Action B4↑ | Justification B4↑ | Speed RMSE↓ | Course RMSE↓ | Average Inference Time (s) | The Ratio of RAG (%) |
|---|---|---|---|---|---|---|
| Base (No-RAG) | 31.7 | 7.4 | 0.63 | **3.94** | 1.64 | 0 |
| Base+SFT | 36.9 | 9.0 | 0.64 | 3.96 | 2.89 | 67.8 |
| Base+SFT+RL | **38.9** | **10.1** | **0.62** | 4.14 | 2.85 | 66.0 |

Table 6: Ablation study on our LLM scorer on the BDD-X dataset (Kim et al., 2018) in terms of the action and justification predictions.

| Method | LLM scorer | Action B4↑ | Justification B4↑ |
|---|---|---|---|
| Base (No-RAG) | | 31.7 | 7.4 |
| Base+SFT | | 36.1 | 8.8 |
| Base+SFT | ✓ | **36.9** | **9.0** |
| Base+SFT+RL | | 37.5 | 9.3 |
| Base+SFT+RL | ✓ | **38.9** | **10.1** |

**Effects of the LLM scorer.** Table 6 evaluates the impact of the LLM-based semantic scorer. We report only action and justification results, as the LLM scorer operates on high-level reasoning outputs. During SFT, the LLM scorer is used for automatic labeling by evaluating reasoning quality. Without it, reasoning scores are obtained by averaging traditional metrics. Incorporating the LLM scorer consistently improves SFT performance. During RL, the LLM scorer serves as the reward model, and replacing it with traditional metrics leads to clear performance degradation. These results demonstrate that the LLM-based reward provides more faithful optimization signals. Moreover, potential bias and noise from the LLM scorer do not introduce noticeable negative effects during training.

## 5 Conclusion

We propose ARDrive, an RAG-aware framework that enables autonomous driving agents to dynamically determine when RAG is necessary through a two-stage training strategy. Extensive experiments demonstrate that ARDrive significantly improves inference efficiency while achieving competitive performance compared to conventional always-RAG approaches.

## AI USAGE DISCLOSURE

Generative AI tools were used only for code checking, manuscript language refinement, and organizational suggestions. They were not involved in the development of the methodology, experimental design, hypothesis formulation, implementation of the proposed method, or interpretation of experimental results. All AI-assisted outputs were manually reviewed and verified by the authors. The authors take full responsibility for the final manuscript, including all text, claims, code, and other AI-assisted content.

## REFERENCES

Anthropic. Claude 4 system card: Claude opus 4 and claude sonnet 4. Technical report, Anthropic, 2025. Technical report.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Int. Conf. Learn. Represent.*, 2024.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2004.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

Bankiti V. Lang A. H. Vora S. Liong V. E. Xu Q. Krishnan A. Pan Y. Baldan G. Caesar, H. and O Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 24185–24198, 2024.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. 2025.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. 2025.

Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6971–6988, 2023.

Wenhao Ding, Sushant Veer, Yuxiao Chen, Yulong Cao, Chaowei Xiao, and Marco Pavone. RealDrive: Retrieval-augmented driving with diffusion models. *arXiv preprint arXiv:2505.24808*, 2025.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics, Volume 1: Long Papers*, pp. 7036–7050, 2024.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

Liu X. Zheng Y. Li P. Zhao H. Zhang T. Zheng Y. Zhou G. Jin, B. and J. Liu. ADAPT: Action-aware drivingcaption transformer. In *2023 IEEE International Conference on Robotics and Automation*, pp. 7554–7561, 2023.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *Int. Conf. Learn. Represent.*, 2024.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. *Proceedings of the Eur. Conf. Comput. Vis.*, 2018.

Todd Ward Kishore Papineni, Salim Roukos and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Adv. Neural Inform. Process. Syst.*, volume 33, pp. 9459–9474, 2020.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVa: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-Xin Yang. AdaThinkDrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

OpenAI. GPT-5 system card. Technical report, OpenAI, 2025. Technical report.

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 *Adv. Neural Inform. Process. Syst.*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Adv. Neural Inform. Process. Syst.*, 2023.

C Lawrence Zitnick Ramakrishna Vedantam and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4566–4575, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:arXiv*, 2017.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *Proceedings of the Eur. Conf. Comput. Vis. 2024*, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Adv. Neural Inform. Process. Syst.*, pp. 3008–3021, 2020.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12991–13013, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In *8th Annual Conference on Robot Learning*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Xi Wang, Procheta Sen, Ruizhe Li, and Emine Yilmaz. Adaptive retrieval-augmented generation for conversational systems. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 491–503, April 2025.

Yujin Wang, Quanfeng Liu, Jiaqi Fan, Jinlong Hong, Hongqing Chu, Mengjian Tian, Bingzhao Gao, and Hong Chen. RAC3: Retrieval-augmented corner case comprehension for autonomous driving with vision-language models. *arXiv preprint arXiv:2412.11050*, 2024.

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*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhu Yu, Runmin Zhang, Jiacheng Ying, Junchen Yu, Xiaohai Hu, Lun Luo, Si-Yuan Cao, and Hui-liang Shen. Context and geometry aware voxel transformer for semantic scene completion. In *Adv. Neural Inform. Process. Syst.*, volume 37, pp. 1531–1555, 2024.

Zhu Yu, Bowen Pang, Lizhe Liu, Runmin Zhang, Qiang Li, Si-Yuan Cao, Maochun Luo, Mingxia Chen, Sheng Yang, and Hui-Liang Shen. Language driven occupancy prediction. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis.*, pp. 7548–7558, October 2025.

Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. RAG-Driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. *arXiv preprint arXiv:2402.10828*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF Int. Conf. Comput. Vis.*, pp. 11941–11952, 2023.

Han Zhang, Yu Lei, Lin Gui, Min Yang, Yulan He, Hui Wang, and Ruifeng Xu. CPPO: Continual learning for reinforcement learning from human feedback. In *Int. Conf. Learn. Represent.*, 2024a.

Jiawei Zhang, Xuan Yang, Taiqi Wang, Yu Yao, Aleksandr Petiushko, and Bo Li. SafeAuto: Knowledge-enhanced safe autonomous driving with multimodal foundation models. In *Forty-second International Conference on Machine Learning*, 2025.

Zihan Zhang, Meng Fang, and Ling Chen. RetrievalQA: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL*, pp. 6963–6975, 2024b.

Weicheng Zheng, Xiaofei Mao, Nanfei Ye, Pengxiang Li, Kun Zhan, XianPeng Lang, and Hang Zhao. DriveAgent-R1: Advancing vlm-based autonomous driving with hybrid thinking and active perception. *arXiv preprint arXiv:2507.20879*, 2025.

Jiacheng Zuo, Haibo Hu, Zikang Zhou, Yufei Cui, Ziquan Liu, Jianping Wang, Nan Guan, Jin Wang, and Chun Jason Xue. RALAD: Bridging the real-to-sim domain gap in autonomous driving with retrieval-augmented learning. *arXiv preprint arXiv:2501.12296*, 2025.

# APPENDIX

## A MORE QUANTITATIVE ANALYSIS OF LABEL QUALITY

We manually audited a random 10% subset of the pseudo-labeled BDD-X training set. Each sample was independently reviewed by 5 annotators, who were shown the gt, RAG output, No-RAG output triplet and asked to decide whether the auto-assigned label was correct. The final agreement between the automatic label and the majority human judgment is 99% for action and 92% for justification.

### A.1 IMPLEMENTATION DETAILS

#### A.1.1 COMPARISON METHODS

On the **BDD-X** dataset (Kim et al., 2018), we compare ARDrive with several representative approaches:

- **ADAPT** (Jin & Liu, 2023), a state-of-the-art video transformer framework that separately models high-level reasoning and low-level control through dual-branch prediction.
- **Time-LLM** (Jin et al., 2024), a reprogrammed large language model for time-series forecasting that converts numerical inputs into text-like patches for unified sequence modeling.
- **DriveGPT4** (Xu et al., 2024), the first multimodal large language model (MLLM) for autonomous driving that jointly generates high-level reasoning and low-level control in an end-to-end manner.
- **RAG-Driver** (Yuan et al., 2024), a RAG-enhanced generation framework that employs triplet loss to align multimodal driving representations and improve reasoning transparency.
- **SafeAuto** (Zhang et al., 2025), a recent RAG-based driving model that introduces a safety-oriented memory retrieval mechanism to provide contextually relevant external knowledge for decision making.

In addition, on the **processed DriveLM** dataset (Sima et al., 2024; Zhang et al., 2025), we compare ARDrive with leading multimodal reasoning and motion prediction methods:

- **UniAD** (Hu et al., 2023), the state-of-the-art unified perception–prediction–planning framework on nuScenes (Caesar & Beijbom, 2020). We adopt a single-frame variant for a fair comparison on the processed DriveLM.
- **BLIP-RT-2**, a BLIP-2 (Li et al., 2023)-based baseline fine-tuned on DriveLM using trajectory tokenization as proposed in RT-2 (Brohan et al., 2023).
- **DriveLM-Agent** (Sima et al., 2024), a graph-based visual question answering model that performs joint high-level reasoning and motion prediction through structured multimodal context modeling.

#### A.1.2 TRAINING DETAILS

ARDrive is built upon Video-LLaVA-7B (Lin et al., 2024), and the RAG engine follows the implementation of SafeAuto (Zhang et al., 2025). All experiments are conducted on two H100 GPUs. In the SFT stage, ARDrive is trained for 2 epochs with a learning rate of $2 \times 10^{-5}$ and a per-GPU batch size of 8. The pretrained weights from this stage are then used to initialize the RL stage. During RL training, we use the AdamW optimizer with a learning rate of $1 \times 10^{-6}$, mixed-precision training, and DeepSpeed ZeRO-2 with CPU offloading. We adopt GRPO with group size $n = 4$ and set the per-GPU batch size to 1. The LLM scorer (Qwen-Max (Yang et al., 2025)) uses greedy decoding with temperature 0 for both SFT labeling and RL reward computation, so that the scores are deterministic. The RL stage is also trained for 2 epochs. The KL regularization weight $\lambda$ and RAG cost $c_k$ are set to $1.5 \times 10^{-1}$ and $5 \times 10^{-2}$, respectively. Ablations on these hyper-parameters are provided in the supplementary material.

For multi-turn interactions in BDD-X (Kim et al., 2018), we train one GRPO (Shao et al., 2024) step per turn. At each turn, we sample a group of candidates from $\pi_{\text{old}}$, compute outcome-based

rewards for each candidate , and perform group-normalized GRPO updates with KL regularization. The highest-scoring candidate at turn $t$ is then committed to the dialog history and becomes input to turn $t+1$. Fig. 5 shows a data sample in BDD-X (Kim et al., 2018). When the RAG triggers at any turn, we inject the corresponding history experience (Fig. 6) obtained from the RAG engine (Zhang et al., 2025) for that turn to obtain the answer. To avoid redundancy and additional latency, once retrieval has been triggered for a sample, later turns do not re-inject the same knowledge. For reporting the RAG usage rate, a dialog is counted as RAG-used if retrieval is triggered at least once. Otherwise it is counted as No-RAG.


```
{
"id": "xxx",
"video": [
"…/video.mp4",
],
"conversations": [
{
"from": "human",
"value": "The current video records driving scenario: <video>\n
Control Signal until current Frame Sequence is: Speed: [14.793, 12.249,
08.650, 06.351, 05.280, 04.924, 04.825]\n
Curvature: [00.000, 00.000, 00.000, 00.000, 00.000, 00.000, 00.000]\n
Acceleration: [-00.737, -02.737, -02.445, -01.421, -00.565, -00.179, -
00.084]\n
Course: [00.000, 00.000, 00.000, 00.000, 00.000, 00.000, 00.000]\n
What is the action of ego car?"
},
{
"from": "gpt",
"value": "The car merges left and comes to a stop"
},
{
"from": "human",
"value": "Why does the ego car doing this?"
},
{
"from": "gpt",
"value": "since the lane is clear but the light turns red."
},
{
"from": "human",
"value": "Predict the control signal for next frame."
},
{
"from": "gpt",
"value": "Speed: 04.632 Course: 00.000"
}
]
},
```


Figure 5: A data sample from the BDD-X dataset (Kim et al., 2018).

### A.1.3 Details of the RAG Engine

Following SafeAuto (Zhang et al., 2025), we adopt a multimodal RAG pipeline that builds a unified embedding over video, control signals, and environmental predicates, and retrieves the top $K$ most similar past driving experiences from a database constructed on the training split. Each modality is projected by a four-layer MLP to a shared dimension, and then linearly fused with learned weights to form the unified query embedding. The MLP projectors are trained with ranking-by-text objective of SafeAuto (Zhang et al., 2025). At inference, we use the unified query embedding to retrieve the top $K$ similar experiences from the database and concatenate the retrieved context with the current prompt for the model to answer. Unless otherwise specified, we match SafeAuto's settings, *i.e.*, $K = 2$ on BDD-X and $K = 1$ on DriveLM. Notably, the RAG engine itself is not the core of our work since we focus on the adaptive RAG strategy.

## A.2 More Ablation Results

**Effect of the SFT+RL pipeline.** The adaptive retrieval strategy in ARDrive is not independent of the SFT+RL pipeline. This pipeline is the mechanism that enables the model to learn the policy, *i.e.*,

```
{
"video": [
"…/history1.mp4",
"…/history2.mp4",
],
"conversations": [
{
"from": "human",
"value": "Here are some historical driving experiences and corresponding
question-and-answer examples:\n\n

Experience 0:\n
Human: The current video records driving scenario: <video>\n Control Signal
until current Frame Sequence is: Speed: [04.840, 00.457, 00.300, 00.345,
00.000, 00.000, 00.000]\n
Curvature: [00.000, 00.000, 00.000, 00.000, 00.000, 00.000, 00.000]\n
Acceleration: [-02.965, -00.671, -00.018, -00.101, 00.000, 00.000,
00.000]\n
Course: [00.000, 00.000, 00.000, 00.000, 00.000, 00.000, 00.000]\n
What is the action of ego car?\n
Assistant: The car stops at the intersection waiting to turn left\n
Human: Why does the ego car doing this?\n
Assistant: because the left turn light is red.\n
Human: Predict the control signal for next frame.\n
Assistant: Speed: 00.040 Course: 00.000\n\n

Experience 1:\n
Human: The current video records driving scenario: <video>\n
Control Signal until current Frame Sequence is: Speed: [11.090, 10.977,
10.885, 11.106, 11.144, 09.613, 06.067]\n
Curvature: [-00.005, -00.008, -00.018, -00.026, -00.041, -00.027, -
00.006]\n
Acceleration: [-00.460, -00.066, 00.016, 00.168, -00.310, -02.123, -
02.925]\n
Course: [00.000, 00.207, 00.444, 00.562, 00.687, 00.909, 00.219]\n
What is the action of ego car?\n
Assistant: The car merges right then comes to a stop\n
Human: Why does the ego car doing this?\n
Assistant: because the light is red.\n
Human: Predict the control signal for next frame.\n
Assistant: Speed: 02.115 Course: 00.000\n\n},
]
},
```


Figure 6: Visualization of the retrieved knowledge from the RAG engine (Zhang et al., 2025) for the data sample in Fig. 5.

the SFT stage provides the initial perception-and-reasoning capability together with RAG-awareness supervision, while the RL stage further refines the retrieval policy and improves robustness to imperfect pseudo labels. To examine whether the gains mainly come from the extra training stages alone, we conducted an additional experiment in which we applied GRPO to the SFT-trained Always-RAG model while keeping retrieval always on. The results on the BDD-X test set are shown in Table 7. Applying GRPO to the Always-RAG baseline brings only limited gains and does not reproduce the improvements of ARDrive. This suggests that the benefit cannot be explained by the additional training pipeline alone. Instead, the gain comes from learning an adaptive retrieval policy through this pipeline.

Table 7: Performance comparison of Base (Always-RAG, SFT only), Base (Always-RAG, SFT+GRPO), and ARDrive on the BDD-X dataset (Kim et al., 2018). We report BLEU-4 for action and justification prediction, and RMSE for speed and course prediction.

| Method | Action B4 ↑ | Justification B4 ↑ | Speed RMSE ↓ | Course RMSE ↓ |
|---|---|---|---|---|
| Base (Always-RAG, SFT only) | 38.1 | 9.3 | 0.65 | 3.85 |
| Base (Always-RAG, SFT+GRPO) | 38.4 | 9.5 | 0.69 | 3.85 |
| **ARDrive (Ours)** | **38.9** | **10.1** | **0.62** | 4.14 |

**High-level performance on the processed DriveLM dataset (Zhang et al., 2025).** The processed DriveLM test split used in prior work does not provide public high-level annotations. Luckily,

the released training split does include such annotations. To address this concern, we additionally used the released high-level annotations from the DriveLM training data to construct a held-out split (3000 train / 447 test) for supplementary evaluation on richer reasoning tasks. On this held-out split, we evaluate Safe Action prediction using BLEU-4, Behavior prediction using accuracy, and also report motion prediction for completeness.The results in Table 8 show that the adaptive-RAG benefit also extends to high-level reasoning tasks on this split. ARDrive outperforms both No-RAG and Always-RAG baselines on Safe Action and Behavior, while also maintaining the best motion-prediction performance. Additionally, the results also suggest that always-on retrieval is not uniformly optimal on this held-out split.

Table 8: Performance comparison of Base (No-RAG), Base (Always-RAG), and ARDrive in terms of safe action prediction, behavior prediction, and motion forecasting on the processed DriveLM (Zhang et al., 2025) dataset. We report BLEU-4 for safe action prediction, accuracy for behavior prediction, and ADE/FDE for motion forecasting.

| Method | Safe Action B4 ↑ | Behavior Acc ↑ | Motion ADE ↓ | Motion FDE ↓ |
|---|---|---|---|---|
| Base (No-RAG) | **0.41** | 0.63 | 1.83 | 3.58 |
| Base (Always-RAG) | 0.39 | 0.62 | 2.39 | 4.45 |
| **ARDrive (Ours)** | **0.41** | **0.65** | **1.56** | **3.16** |

**Effect of the hyper-parameter** $\lambda$. The hyper-parameter $\lambda$ controls the strength of KL regularization toward the SFT reference policy. A very small $\lambda$ allows the RL stage to freely chase the reward model, which may lead to overfitting and unstable behavior, while an excessively large $\lambda$ constrains the policy to stay too close to the SFT model and reduces the effective benefit of RL fine-tuning (Shao et al., 2024). Table 9 reports the effect of $\lambda$. When $\lambda = 0$, the policy is unconstrained by KL and tends to overuse RAG, yielding the highest RAG activation ratio of $72.5\%$ and worse semantic performance compared with $\lambda = 0.15$. In this setting, the course RMSE reaches its lowest value, suggesting that an aggressive RL update may overfit the steering signal while sacrificing high-level reasoning quality and RAG efficiency. Introducing a moderate KL weight $\lambda = 0.15$ improves overall performance and reduces speed RMSE, while also lowering the RAG ratio to $66.0\%$, which is below the SFT RAG ratio of $67.8\%$. Here the course RMSE slightly increases to $4.14$, reflecting a small trade-off on steering accuracy in exchange for more stable and efficient reasoning. When $\lambda$ is further increased to $0.30$, the behavior is pulled back closer to the SFT policy, with a RAG ratio of $68.3\%$ and degraded semantic metrics. Overall, $\lambda = 0.15$ provides a good trade-off between leveraging RL improvements, controlling RAG usage, and staying sufficiently close to the SFT reference.

Table 9: Ablation study on the hyper-parameter $\lambda$ on the BDD-X dataset.

| $\lambda$ | Action B4↑ | Justification B4↑ | Speed RMSE↓ | Course RMSE↓ | RAG ratio % |
|---|---|---|---|---|---|
| 0 | 38.4 | 9.7 | 0.64 | 3.85 | 72.5 |
| 0.15 | 38.9 | 10.1 | 0.62 | 4.14 | 66.0 |
| 0.30 | 38.3 | 9.6 | 0.63 | 4.01 | 68.3 |

**Effect of the hyper-parameter** $c_k$. The hyper-parameter $c_k$ controls the cost for invoking RAG in the reward function. A larger $c_k$ makes each RAG call more expensive, which in principle encourages the policy to invoke RAG only when the expected performance improvement justifies the additional cost and to skip RAG in simpler cases. Table 10 reports the effect of the RAG cost $c_k$. When $c_k$ increases from $0$ to $0.05$, the main task metrics remain essentially unchanged, while the RAG activation ratio slightly decreases from $67.0\%$ to $66.0\%$. This indicates that a small cost mainly acts as a mild regularizer that discourages unnecessary RAG calls without altering the overall behavior of the policy. When $c_k$ is further increased to $0.50$, the RAG ratio is reduced more noticeably to $63.7\%$, but we also observe a slight degradation in justification. These results suggest that overly penalizing RAG can make the agent overly conservative and start to harm reasoning quality in complex situations. We therefore adopt $c_k = 0.05$ as a balanced choice between performance and RAG efficiency.

Table 10: Ablation study on the hyper-parameter $c_k$ on the BDD-X dataset.

| $c_k$ | Action B4↑ | Justification B4↑ | Speed RMSE↓ | Course RMSE↓ | RAG ratio % |
|---|---|---|---|---|---|
| 0 | 38.8 | 9.8 | 0.62 | 4.13 | 67.0 |
| 0.05 | 38.9 | 10.1 | 0.62 | 4.14 | 66.0 |
| 0.50 | 38.4 | 9.7 | 0.62 | 4.18 | 63.7 |

**Effect of the different LLM scorer**. Table 11 compares ARDrive and the No-RAG baseline under five different LLM scorers. For each scorer, we feed the model prediction and the corresponding ground truth for 1,000 randomly selected BDD-X test instances and obtain a normalized semantic similarity score, which is then averaged over these 1,000 instances. Across Qwen-Max (Yang et al., 2025), Claude Sonnet 4 (Anthropic, 2025), DeepSeek-R1 (DeepSeek-AI, 2025), Gemini 2.5 Pro (Comanici et al., 2025) and GPT-5 (OpenAI, 2025), ARDrive consistently achieves higher scores than the No-RAG baseline. Additionally, scores from different LLM remains relatively consistent. These results indicate that different LLM scorers provide consistent assessments, which suggests that our framework is not tailored to a specific LLM but remains effective under different LLM evaluators.

Table 11: Ablation study on different LLM scorers. Each scorer evaluates the semantic similarity between the model prediction and the ground truth on 1,000 randomly selected BDD-X test instances.

| LLM Scorer | ARDrive | | Base (No-RAG) | |
|---|---|---|---|---|
| | Action | Justification | Action | Justification |
| Qwen-Max (Yang et al., 2025) | 0.82 | 0.40 | 0.66 | 0.38 |
| Claude Sonnet 4 (Anthropic, 2025) | 0.80 | 0.41 | 0.63 | 0.37 |
| DeepSeek-R1 (DeepSeek-AI, 2025) | 0.80 | 0.45 | 0.60 | 0.40 |
| Gemini 2.5 Pro (Comanici et al., 2025) | 0.83 | 0.44 | 0.64 | 0.37 |
| GPT-5 (OpenAI, 2025) | 0.82 | 0.40 | 0.63 | 0.35 |

Notably, we adopt conventional metrics (BLEU-4 (Kishore Papineni & Zhu, 2002), METEOR (Banerjee & Lavie, 2004), and CIDEr (Ramakrishna Vedantam & Parikh, 2015)) as the main evaluation criteria in our main experiments. These metrics are widely adopted in previous work (Zhang et al., 2025; Yuan et al., 2024; Kim et al., 2018) and have been shown to be effective on evaluating high-level predictions, even though they can be imperfect in certain corner cases. Moreover, since ARDrive leverages an LLM-based scorer during training, using an LLM-based evaluation could introduce unfair bias in favor of our method and blur the boundary between training signal and evaluation metric, especially given that different LLM scorers exhibit differences in their score scales. Therefore, we treat LLM-based scores as a complementary robustness analysis that corroborates the advantage of ARDrive, while using standard n-gram based metrics as the main evaluation metrics for quantitative comparison in the main manuscript.

**Effect of the scale of the knowledge base in the RAG engine.** We evaluated ARDrive under different knowledge base (KB) sizes from 1k to 16k entries in Table 12. On BDD-X (Kim et al., 2018), we find that around 13k entries are already sufficient, and increasing the KB size to 16k brings only marginal benefit. We used 16k in the paper to follow previous works (Yuan et al., 2024; Zhang et al., 2025). When the KB is too small (1k entries), RAG quality is poor, so the learned policy almost never activates RAG (1.7%), and the performance remains close to the No-RAG baseline. With 8k entries, RAG becomes more useful, and the model correspondingly starts to invoke RAG much more often (36.5%), together with improved performance. At 13k entries, the results are already very close to those with 16k entries, both in task metrics and in RAG ratio, suggesting that the retrieval pool is already sufficiently rich for this dataset. This is also consistent with the design logic of ARDrive. The gating policy is not learned as a function of KB size itself, but as a function of the retrieval benefit for the current context. Therefore, we do not expect a simple monotonic scaling law with the number of entries alone. A larger KB would increase RAG usage only if it actually improves retrieval quality and makes retrieval more beneficial for the current query.

Moreover, BDD-X (Kim et al., 2018) is a 3-turn multi-turn dataset with action, justification, and signal prediction turns. Meanwhile, our paper reports RAG usage at the dialog level, not the turn

Table 12: Ablation study of ARDrive with different numbers of training entries on the BDD-X dataset (Kim et al., 2018). We report BLEU-4 for action and justification prediction, RMSE for speed and course prediction, and the ratio of using RAG.

| Method | Action B4 ↑ | Justification B4 ↑ | Speed RMSE ↓ | Course RMSE ↓ | RAG Ratio |
|---|---|---|---|---|---|
| Base (No-RAG) | 31.7 | 7.4 | 0.63 | 3.94 | 0.0% |
| ARDrive (1k entries) | 31.9 | 7.7 | 0.64 | **3.89** | 1.7% |
| ARDrive (8k entries) | 32.1 | 8.0 | 0.63 | 4.06 | 36.5% |
| ARDrive (13k entries) | **39.0** | 9.7 | **0.62** | 4.18 | 65.8% |
| **ARDrive (16k entries)** | 38.9 | **10.1** | **0.62** | 4.14 | 66.0% |

level. To avoid redundancy and additional latency, once retrieval has been triggered for a sample, later turns do not re-inject the same knowledge. For reporting the RAG usage rate, a dialog is counted as RAG-used if retrieval is triggered at least once. Otherwise it is counted as No-RAG. This means that a dialog is already counted as "RAG-used" once retrieval is triggered at least once, even if the other turns still behave like No-RAG. Therefore, to further understand how KB size affects performance on BDD-X (Kim et al., 2018), it is necessary to go beyond the dialog-level statistic and examine which turns are actually benefiting from retrieval. Under this turn-level view, the effect from 1k to 8k becomes much clearer. The gains in this range are not uniform across turns. On the justification turn, Justification B4 increases from 7.7 to 8.0 (about 3.75% relative improvement), while the correct RAG triggering rate rises sharply from 1.5% to 36.7%. By contrast, on the action turn, Action B4 increases only from 31.9 to 32.1 (about 0.6% relative improvement), and the correct RAG triggering rate also rises only marginally from 0.0% to 5.7%. This alignment suggests that when the KB grows from 1k to 8k, the benefit of retrieval appears first and much more clearly on the justification turn, while the action turn still receives only very limited retrieval support. Therefore, at this stage, the effect of a larger KB is mainly reflected in justification, rather than in a broad improvement across all major metrics. By contrast, from 8k to 13k, the correct RAG triggering rate rises sharply on both major high-level turns, *i.e.*, from 5.7% to 75.3% on action, and from 36.7% to 56.6% on justification. This matches the corresponding metric changes, where Action B4 improves from 32.1 to 39.0 and Justification B4 improves from 8.0 to 9.7. In other words, once retrieval becomes strong enough to benefit not only justification but also action, the main performance gains become clearly visible.

Therefore, our current evidence suggests a more layered interpretation of KB size on BDD-X (Kim et al., 2018). Because BDD-X (Kim et al., 2018) is a multi-turn benchmark, the dialog-level RAG ratio alone is insufficient to explain performance changes. As the KB grows, retrieval benefit becomes visible earlier on justification in our analysis, while the main overall gains emerge only when retrieval quality becomes sufficiently strong to also substantially improve action-turn triggering.

### A.3 Qualitative Results

Fig. 7 presents examples that illustrate the adaptive RAG behavior of ARDrive. When the current observation already provides sufficient visual cues, ARDrive learns to suppress RAG and reason directly in the No-RAG mode (Fig. 7(a)). In contrast, when the scene involves greater contextual uncertainty, ARDrive selectively activates the RAG mode to enhance reasoning (Fig. 7(b)). These qualitative results demonstrate that ARDrive achieves a more favorable balance between accuracy and efficiency.

Notably, in Fig. 7(a), the justification predicted by ARDrive outperforms both the No-RAG and Always-RAG baselines, and in Fig. 7(b), ARDrive produces more accurate signal and justification outputs than either baseline. We attribute this to the adaptive RAG training process, which encourages the model to more deeply understand the current observation (Zhang et al., 2024b) in order to decide whether RAG is necessary. As a result, the model develops stronger intrinsic reasoning capability, allowing it to outperform both static strategies in some cases.

We provide expanded visualizations in Fig. 8 and Fig. 9 for the cases in Fig. 7. Notably, in Fig. 7, the two retrieved experiences both describe left turns under a green signal, which guides ARDrive to correctly answer the green light. Moreover, ARDrive does not merely repeat the retrieved content but also analyzes the current observation to identify a lead vehicle initiating a left turn. This shows

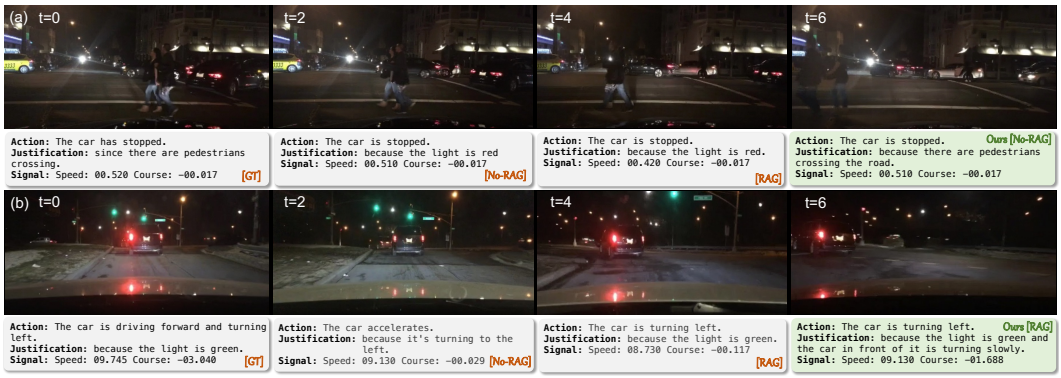


Figure 7: Visualization of the video, corresponding ground truth ([GT]), and representative results produced by Base (No-RAG) ([No-RAG]), Base (RAG) ([RAG]), and our **ARDrive** in the BDD-X dataset (Kim et al., 2018). Please refer to the zoomed-in view for better visualization.




Figure 8: Detailed visualization of the case in Fig. 7 where ARDrive decides to use RAG

that in certain cases ARDrive can integrate retrieved context with the present scene to produce a more informative and accurate output.

## A.4 Discussion

### A.4.1 Originality of Our ARDrive

Prior adaptive-RAG studies establish adaptive retrieval as a general concept, while also showing that its effective formulation for a specific task can constitute a meaningful contribution. Following this view, ARDrive develops an autonomous-driving-specific adaptive RAG process. ARDrive shows empirically that this problem matters in this domain and establishes adaptive retrieval as a meaningful and learnable problem in autonomous driving.

ARDrive challenges the assumption that retrieval is always beneficial for AD. Prior AD-RAG methods such as RAG-Driver and SafeAuto are built on the implicit assumption that retrieval is uniformly beneficial and therefore apply retrieval in an always-on manner. ARDrive shows that Always-RAG is not optimal in AD, *i.e.*, retrieval should be used selectively, and learning when to retrieve improves both efficiency and performance. We believe this is a meaningful domain-grounded contribution.

ARDrive makes RAG necessity learnable in autonomous driving, where such supervision is not directly available. The key challenge in autonomous driving is that whether retrieval is beneficial

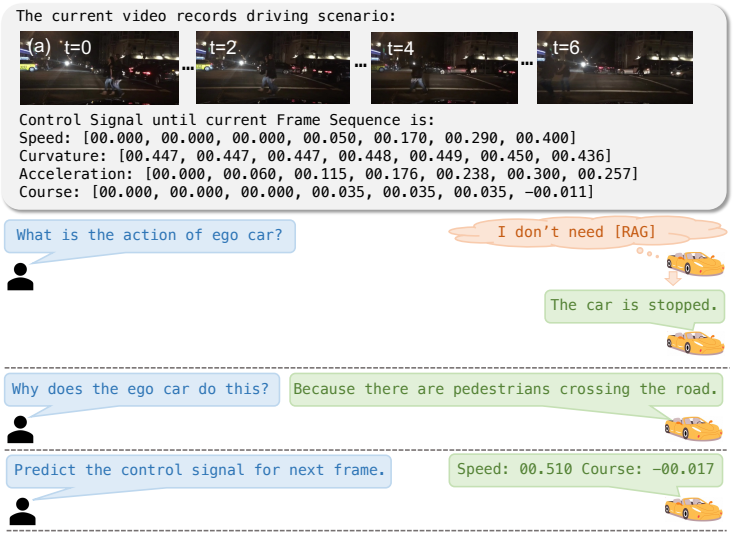


Figure 9: Detailed visualization of the case in Fig. 7 where ARDrive decides not to use RAG

is determined by the actual reasoning gain that RAG provides under the current context. As a result, the retrieval-needed supervision is not directly available. Therefore, ARDrive formulates retrieval necessity as a comparative utility problem between RAG and No-RAG, and learns this decision through a performance-driven learning pipeline tailored to AD. In particular, we design an automatic labeling strategy based on RAG-vs.-No-RAG comparison, and further use a scorer to provide supervision for training the policy.

### A.4.2 Reliability of the LLM Scorer

The reliability of the LLM Scorer used in ARDrive is supported by validations at multiple stages.

(1) *We have conducted additional human evaluation to validate the final explanations and RAG decisions.* Specifically, we conducted a blinded evaluation on the full BDD-X test set (N=2,123) with five annotators. Each sample was independently evaluated by five annotators, and all reported labels were determined by majority vote. For retrieval helpfulness, annotators compared the RAG and No-RAG outputs under the same input and judged whether retrieval improved the final response. Gate agreement measures whether ARDrive's RAG/No-RAG decision matches this majority judgment after excluding neutral cases. ARDrive achieved 83.6% factual correctness and 82.9% safety relevance. Among activated retrievals, 57.4% were judged helpful, 29.1% neutral, and 13.5% harmful. The gate agreed with the majority human judgment in 64.2% of non-neutral cases.

(2) *We have validated the scorer-generated supervision.* Five annotators audited a random 10% of the 32,780 Action and Justification pseudo-labels generated from the BDD-X training set (N=3,278). The automatic labels agreed with the majority human judgment in 99% of Action samples and 92% of Justification samples (Sec. 3.2.1).

(3) *Table 6 ablates the scorer in both training stages.* Replacing the LLM scorer with traditional metrics reduces the B4 scores of Action and Justification. Thus, the scorer provides a more effective training signal.

(4) *Table 11 evaluates 1,000 test samples using Qwen-Max (Yang et al., 2025), Claude Sonnet 4 (Anthropic, 2025), DeepSeek-R1 (DeepSeek-AI, 2025), Gemini 2.5 Pro (Comanici et al., 2025), and GPT-5 (OpenAI, 2025).* All five scorers consistently rank ARDrive above No-RAG.

(5) *The main results themselves are evaluated using conventional metrics, including BLEU-4, METEOR, and CIDEr.* Our main evaluation uses conventional language metrics. The improvements on conventional metrics validate the effectiveness of the LLM scorer.

Collectively, the human assessment, human pseudo-label audit, metric-based ablation, cross-scorer evaluation, and independent task metrics provide evidence that the reported improvements are not specific to a single LLM evaluator.

### A.4.3 System Bottleneck of the Latency

The main system bottleneck is the VLM reasoning cost of processing the additional retrieved context. Specifically, we performed an online retrieval benchmark on the same device used in our paper and found that the retrieval stage (query embedding and nearest-neighbor search) is about 1 ms/query on average. This indicates that the dominant overhead comes from the model-side computation needed to consume the extra retrieved context, which explains why the latency reduction is smaller than the reduction in RAG frequency (Table 1). ARDrive provides a way to reduce unnecessary RAG-induced overhead. Other methods such as backbone optimization, context compression, and quantization are complementary and can be combined with ARDrive.

### A.4.4 Comparison with Adaptive Retrieval Baselines

To examine whether the gains of ARDrive can be reproduced by simpler adaptive retrieval strategies, we compare it with three baselines covering retrieval similarity, generation uncertainty, and learned multimodal routing.

Table 13: Comparison with adaptive retrieval baselines on the BDD-X dataset (Kim et al., 2018). We report BLEU-4 for action and justification prediction, RMSE for speed and course prediction, and the dialogue-level RAG ratio.

| Method | Action B4 ↑ | Justification B4 ↑ | Speed RMSE ↓ | Course RMSE ↓ | RAG Ratio ↓ |
|---|---|---|---|---|---|
| Base (No-RAG) | 31.7 | 7.4 | 0.63 | 3.94 | 0.0% |
| Retrieval-similarity threshold | 35.0 | 9.2 | 0.66 | 3.96 | 52.7% |
| Entropy threshold | 36.3 | 7.5 | 0.64 | 4.14 | 45.9% |
| Multimodal learned classifier | 37.7 | 8.4 | 0.63 | 4.64 | 100.0% |
| Base (Always-RAG) | 38.1 | 9.3 | 0.65 | **3.85** | 100.0% |
| **ARDrive** | **38.9** | **10.1** | **0.62** | 4.14 | 66.0% |

For all methods, the dialogue-level RAG ratio counts a dialogue as using RAG if retrieval is activated in any of its three task turns.

(1) *Both threshold-based baselines perform worse than ARDrive.* The retrieval-similarity baseline activates RAG when the top-1 cosine similarity returned by the retrieval encoder exceeds the mean similarity computed on the training set. The entropy baseline first generates a short No-RAG draft and activates RAG when its mean token entropy exceeds the corresponding training-set mean. Both thresholds remain fixed during testing, and the latency of the entropy baseline includes the preliminary draft generation. These results show that retrieval similarity or generation uncertainty alone cannot reliably estimate the actual utility of retrieval in multimodal driving.

(2) *Under the dialogue-level metric, the classifier fails to provide an effective trade-off.* Its three task-specific classifiers activate RAG for 74.9% of Action samples, 46.5% of Justification samples, and 36.5% of Signal samples. Following our dialog-level evaluation protocol described in Appendix B.2, a sample is counted as using RAG if retrieval is triggered in any of its three turns. Under this protocol, every test sample is assigned RAG by at least one classifier head, resulting in a 100% dialogue-level RAG ratio. Therefore, the classifier provides no dialogue-level retrieval saving and performs worse than ARDrive on all four metrics. It also underperforms Always-RAG on Action, Justification, and Course.

### A.4.5 When does ARDrive choose to use RAG?

To better understand the behavior of the learned adaptive RAG strategy, we further analyze the textual summaries produced by ARDrive on the BDD-X test set (Kim et al., 2018). For each sample, we examine the model's final decision description and manually categorize the underlying driving situation. Among the samples where ARDrive selects the No-RAG mode, we find that roughly 45% correspond to simple stopped scenarios (for example, "The car is stopped") and about 30% to

straightforward cruising ("The car is driving forward" with no notable interactions). In contrast, the samples where ARDrive chooses the RAG mode are dominated by more complex situations, such as "The car is merging into the lane to its right", "The car accelerates and turns right", or "The car is moving slowly forward in traffic." This distribution suggests that the learned RAG gate is not triggering randomly or uniformly, but has aligned with scenario complexity, *i.e.*, ARDrive tends to skip RAG in low-uncertainty steady states, while reserving RAG for decision-critical transitions and semantically ambiguous situations where additional contextual reasoning is beneficial.

Moreover, We constructed a safety-critical subset from the BDD-X (Kim et al., 2018) test set to capture scenarios involving pedestrian interaction, turning/intersection maneuvers, braking/stopping, and lane-change/merge behaviors, since these cases are generally more safety-sensitive than simple cruising. This subset covers 49% of the BDD-X test set. We found that the overall RAG activation rate is 66%, while the activation rate on this safety-critical subset is 69%, compared with 63% on the remaining samples. This suggests that the learned policy tends to activate retrieval more often in safety-critical scenarios, and ARDrive correspondingly activates RAG more frequently in these cases. Importantly, this is an empirical behavior of the learned policy, not because ARDrive is explicitly triggered by safety labels, but because its activation strategy is learned based on whether retrieval improves the current reasoning

Besides, ARDrive remains effective on this safety-critical subset and activates RAG more frequently for them. On the safety-critical subset defined above, the RAG activation rate is 69%, compared with 63% on the remaining test samples. As shown in Table 14, ARDrive achieves the best Action, Justification, and Speed performance on this subset, supporting the effectiveness of its learned retrieval policy in safety-sensitive scenarios.

Table 14: Performance comparison on the safety-critical subset of the BDD-X test set (Kim et al., 2018). We report BLEU-4 for action and justification prediction and RMSE for speed and course prediction.

| Method | Action B4 ↑ | Justification B4 ↑ | Speed RMSE ↓ | Course RMSE ↓ |
|---|---|---|---|---|
| Base (No-RAG) | 34.4 | 10.0 | 0.56 | **2.26** |
| Base (Always-RAG) | 39.2 | 9.2 | 0.57 | 2.44 |
| **ARDrive** | **42.9** | **10.7** | **0.53** | 3.19 |

### A.4.6 Detailed Discussion of the Course prediction

We observe that course prediction seems to benefit less from our adaptive RAG strategy, as shown in Tables 2, 3, and 5. Course prediction consistently improves with RAG, while speed prediction performs better without it. This discrepancy arises from the BDD-X dataset (Kim et al., 2018), where speed and course are generated in the same reasoning round but exhibit different sensitivities to retrieval. Their conflicting RAG preferences introduce noisy gradients during joint optimization, slightly degrading course accuracy (Shao et al., 2024).

Moreover, the under-activation of retrieval in course-sensitive cases causes the degradation in Table 2. Course prediction is more retrieval-sensitive than speed prediction. Our further case analysis shows that the cases where speed improves but course degrades are associated with lower RAG usage and simpler scenarios. Their RAG activation rate is 6% lower, and turning scenarios are 10% less frequent than in the opposite cases (where speed degrades but course improves). This suggests that course degradation mainly occurs when ARDrive suppresses RAG in cases that appear simple. In such cases, the model behaves more like No-RAG, which tends to help speed but hurt course. Conversely, in more complex cases, retrieval is triggered more often and is more favorable to course.

To verify that this limitation is due to dataset design rather than model capacity, we evaluate ARDrive on the motion prediction task of the processed DriveLM dataset (Sima et al., 2024; Zhang et al., 2025) (Table 4). ARDrive significantly outperforms prior methods, demonstrating its effectiveness for low-level reasoning.

### A.4.7 CLOSED-LOOP DRIVING EVALUATION

In Table 15, we further evaluate ARDrive on Bench2Drive (Jia et al., 2024) (220 routes), a CARLA-based benchmark for closed-loop autonomous driving. Compared with open-loop evaluation on fixed observations, closed-loop driving continuously feeds previous decisions back into the environment and is therefore more sensitive to accumulated prediction errors and policy instability. This experiment evaluates whether ARDrive's adaptive retrieval policy can transfer from open-loop training to sequential closed-loop driving, where retrieval decisions directly affect subsequent vehicle behavior and observations.

Table 15: Closed-loop driving performance of ARDrive and RAG baselines on Bench2Drive (Jia et al., 2024). We report Driving Score (DS), Route Completion (RC), Infraction Penalty (IP), Success Rate (SR), and the decision-step RAG ratio.

| Method | Driving Score ↑ | Route Completion ↑ | Infraction Penalty ↑ | Success Rate ↑ | Decision-step RAG Ratio |
|---|---|---|---|---|---|
| Base (No-RAG) | 31.015 | 43.704% | 0.769 | 3.182% | 0.0% |
| Base (Always-RAG) | 27.191 | 36.286% | 0.797 | 3.636% | 100.0% |
| **ARDrive** | **35.795** | **44.108%** | **0.841** | **5.455%** | **3.806%** |

(1) *We retrain ARDrive on the open-loop Bench2Drive (Jia et al., 2024) training data and evaluate its open-loop-to-closed-loop transfer without any simulator rollouts or closed-loop policy optimization.* Specifically, we rebuild the retrieval engine using the Bench2Drive (Jia et al., 2024) training set, reconstruct the RAG/No-RAG supervision labels, and repeat the complete ARDrive training pipeline from SFT to GRPO. No closed-loop trajectories, simulator rollouts, or policy-level rewards are used during training. During closed-loop evaluation, the model receives the current visual observation and ego state and predicts the target Speed and Course at each decision step. A fixed low-level controller, shared by No-RAG, Always-RAG, and ARDrive, converts these predictions into CARLA control commands. The longitudinal controller maps the error between the predicted and current speed to throttle or brake, while the lateral controller converts the predicted course into steering. The resulting throttle, brake, and steering commands are then executed in CARLA. Therefore, the only difference among the three settings is their retrieval policy.

(2) *ARDrive's closed-loop results are within the expected range.* Without using closed-loop trajectories or policy optimization, ARDrive achieves a Driving Score of 35.795 and a strict Success Rate of 5.455%. As contextual references, open-loop-trained AD-MLP and TCP-ctrl report Driving Scores of 18.05 and 30.47 and Success Rates of 0% and 7.27%, respectively, under the official Bench2Drive (Jia et al., 2024) setting. They provide useful context showing that ARDrive's scores fall within a reasonable and non-trivial range for open-loop-trained models evaluated in closed loop.

(3) *The comparison among retrieval strategies shows that ARDrive learns an effective retrieval policy for closed-loop driving.* On the complete 220-route Bench2Drive (Jia et al., 2024) evaluation, Always-RAG achieves a slightly higher Infraction Penalty and Success Rate than No-RAG, but substantially reduces Route Completion and the overall Driving Score, indicating that continuously injecting retrieved context can interfere with sustained driving control. ARDrive avoids this trade-off by activating RAG on only 3.806% of decision steps, while achieving the best Driving Score, Route Completion, Infraction Penalty, and Success Rate among all three strategies. These results demonstrate that ARDrive selectively invokes retrieval when it is beneficial to closed-loop decision making.

## A.5 THE TEMPORAL CONSISTENCY OF THE ARDRIVE

We observed that the [RAG] gate is relatively stable in continuous video streams. Specifically, we evaluated 100 real driving videos using a 3-second sliding window. Across all adjacent-window pairs, the proportion of unreasonable [RAG] gate switches is about 2.0%. Separately, the proportion of pairs in which a change in retrieved content leads to a change in model output is about 1.8%. These results suggest that both the retrieval gate and the resulting model behavior are relatively stable in continuous video streams. However, we do not exclude the possibility of temporal fluctuation in real-world deployment, since this is a broader challenge in sequential autonomous driving systems rather than an issue unique to ARDrive. Our paper mainly focuses on adaptive retrieval learning. The issue raised here is therefore an important extension setting beyond the main research focus of

the paper. In practice, this can be handled with strategies such as temporal smoothing, hysteresis with separate on/off thresholds, or minimum hold time before switching. These refinements are orthogonal to the core ARDrive framework and can be naturally added in real deployment.

### A.5.1 Limitations and Future Works

Our contribution is a general training framework for learning a domain-matched retrieval policy for autonomous driving. We apply the same pseudo-labeling, SFT, and RL pipeline separately to BDD-X Kim et al. (2018) and DriveLM Sima et al. (2024), allowing each model to learn when retrieval is useful under its own data distribution and task definition. ARDrive consistently outperforms No-RAG and Always-RAG.

However, the trained gate is specific to the target dataset and retrieval engine. Once training is completed, changing the retrieval engine at test time does not update the frozen gate, because the retrieval decision is made before the retrieved content is observed. Therefore, ARDrive does not aim to learn a single universal retrieval policy that transfers across domains without adaptation. Extending a trained gate to unseen domains without retraining remains an important direction for future work.