Decision-Guided Adaptive Data Sampling for Autonomous Driving Model Training

Masaki Nambata[1], Shota Yamazaki[1], Jo Nishiyama[1], and Takuya Nanri[1]

Nissan Motor Co., Ltd., Kanagawa, JP

Abstract. Large-scale driving datasets are widely used to expose autonomous driving models to diverse scenes. However, real-world datasets often contains redundant scenes, creating biased distributions in which safety-critical scenarios are highly rare. This limits models' exposure to safety-relevant scenarios and degrades their performance on safety-critical scenarios. Moreover, the optimal balance between the long tail and common driving scenes cannot be determined in advance because it depends on the definition of rarity, training stability, and model characteristics. To address this problem, we propose a training framework that dynamically optimizes model experience through ontology-based data sampling. The ontology represents driving decisions and their causal factors, allowing scenes to be evaluated not only by model performance but also by driving-relevant events. Using this representation, the framework updates the training dataset according to the learning status of the models. Our experiments show that our method improves the data-efficiency of training and performance in rare driving scenarios.

Keywords: Autonomous Driving · Driving Safety · Adaptive Data Sampling

1 Introduction

End-to-End Autonomous Driving (E2E-AD) models that directly predict future trajectories or control commands from sensor inputs have been widely studied [3, 13, 26, 29, 31]. Recent works have reported scaling laws of E2E-AD models similar to those of large language models, showing that performance improves with more training data [22, 27]. This trend has driven the construction of large-scale driving datasets, and training a model on thousands to tens of thousands of hours of driving data has become common, supporting the development of high-performance foundation models [6, 11, 15, 32]. However, real-world driving data are predominantly composed of routine driving scenes and contain many similar situations, such as driving straight and following a preceding vehicle. Consequently, large-scale driving datasets often exhibit substantial redundancy [14]. Due to this data imbalance, the scenes experienced by the model are concentrated on routine driving, whereas Safety-Critical scenes relatively infrequently and are therefore more likely to lie in the long tail of the data distribution. As a result, increasing data volume simply may yield diminishing returns in

training efficiency and performance on long-tail scenes [7]. As performance on long-tail scenes is critical to driving safety, methods have been proposed to sample data before or during training for efficient learning and selection of long-tail scenes [5, 18, 19, 23, 28].

Data curation methods that prune data in advance preferentially sample long-tail scene data [5, 18, 23, 28]. However, because the complexity of the scenes and the required amount of long-tail data are difficult to define, it is difficult to determine the optimal balance between long-tail and common driving scenes in advance [16, 33].

To address this issue, active learning methods that dynamically sample data during training have been proposed [19]. This approach focuses on reducing the enormous cost of annotation in the training of autonomous driving models. It performs inference on unlabeled data during training and samples those for which the model output accuracy is low—that is, data that model struggles with. This means that data can be sampled dynamically according to the model's learning status. However, accuracy alone does not reveal which driving decisions the model fails to make or which scenarios remain challenging. Driving decisions are determined by the interaction of a wide range of factors, such as other vehicles, pedestrians, traffic signals, lane constraints, and navigation instructions. In addition, even if scenes appear visually distinct, the driving decisions and the factors underlying them may be similar. Thus, treating each scene only as a success or failure is insufficient to identify the experience the model lacks. To improve accuracy not only fundamental scenarios but also in Long-tail ones, it is necessary to enhance model robustness by analyzing scenes in terms of decision-making processes and the factors influencing those decisions, thereby optimizing the model's experience.

In this paper, we propose a training framework based on adaptive sampling of driving decisions and causal factors. Inspired by the Chain of Causation (CoC) dataset [31], we extract a driving decision corresponding to future driving behavior and its causal factors for each scene, and structure them in an ontology. This representation analyzes scenes by the decisions they require and the factors that influence those decisions, revealing experiences the model lacks. During each training epoch, the framework evaluates and identifies low-performing data samples, and uses their ontology representations as queries to retrieve scenes with similar ontology structures from a training data pool. The retrieved samples are added to the training set, it is adaptively updated according to the model's learning status. This exposes the model to scenes it has not experienced sufficiently and improves performance and robustness efficiently.

The main contributions are as follows.

- We introduce a scene representation that structures driving decisions and causal factors in an ontology.
- We propose an ontology-based adaptive sampling framework that retrieves similar scenes on which a model has low performance, and optimizes the model's learning experience.

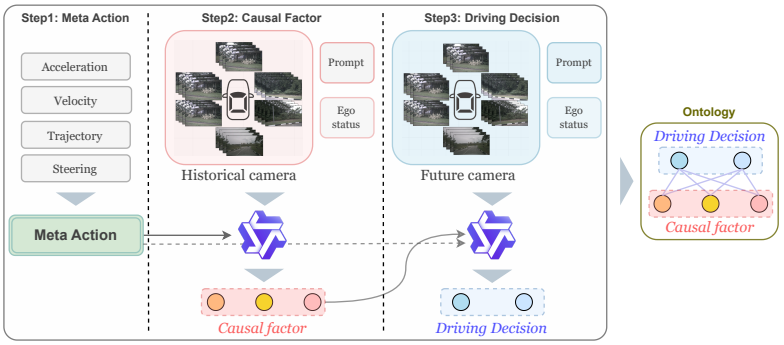

Fig. 1: Ontology annotation pipeline.

– We demonstrate that supplementing lacking experience adaptively enables data-efficient training while improving the performance and robustness of end-to-end autonomous driving models.

2 Related Works

2.1 Autonomous Driving Models

Autonomous driving systems have evolved from early standalone and modular architectures to end-to-end models that integrate perception, prediction, and planning. nuScenes [4], which provides multimodal sensor data such as multi-camera images and LiDAR point clouds, together with high-quality annotations including HD maps and 3D bounding boxes for environment understanding, remains widely used for training and evaluating various autonomous driving models. ST-P3 [12] integrates environmental perception, future prediction, and motion planning using spatiotemporal bird's-eye-view (BEV) features. UniAD [13] proposes a unified framework for jointly learning object tracking, map estimation, motion prediction, and trajectory planning. Furthermore, VAD [3] represents vehicles and map elements as vectors and generates the ego-vehicle trajectory based on the relationships among these elements.

2.2 Data Sampling Methods for Autonomous Driving

Autonomous driving models generally require large-scale training data to achieve high performance, and various driving datasets have therefore been proposed [4, 10, 15, 35]. However, large-scale datasets often contain substantial scene redundancy, which further exacerbates the scarcity of safety-critical scenarios. This redundancy reduces training efficiency and makes it more difficult for models to learn scenarios that are particularly important for driving safety. To address these issues, methods have been proposed to sample informative training

Table 1: Meta action & driving decision definition. Quoted from reference [31].

	Longitudinal		Lateral	
Meta Action	Gentle accelerate	Strong accelerate	Steer left	Steer right
	Gentle decelerate	Strong decelerate	Sharp steer left	Sharp steer right
	Maintain speed	Stop	Reverse left	Reverse right
	Reverse	-	Go straight	-
Driving Decision	Set speed tracking	Lead obstacle following	Lane keeping & centering	Merge / Split (facility change)
	Speed adaptation (road events)	Gap-searching (for LC/merge/zipper)	Out-of-lane nudge (straddle avoidance)	In-lane nudge
	Acceleration for passing/overtaking	Yield (agent right-of-way)	Lane change (lateral push)	Pull-over / curb approach
	Stop for static constraints	-	Turn (intersection/roundabout/U-turn)	Lateral maneuver abort

data either before or during training. These approaches include dataset curation methods for pre-training sampling [5,18,23,28] and active learning methods for sampling during training [19–21]. However, pre-sampling of data may fail to select the data that are most beneficial for a particular model's learning process because it primarily relies on the data distribution. On the other hand, evaluating samples solely on the basis of accuracy, as commonly performed in active learning, may be insufficient to sample data appropriately according to the current state and specific weaknesses of the model because driving decisions in autonomous driving are determined by interactions among diverse factors in the surrounding environment.

3 Method

In this study, we aim to improve model performance and robustness by optimizing model experience. As the optimal data distribution is difficult to determine in advance, training data must be dynamically sampled according to the learning status of the models. We therefore propose a training framework that performs adaptively data sampling through scene retrieval based on driving decisions and their causal factors. First, we construct a pipeline that automatically annotates each scene in an existing dataset with an ontology representation for retrieval. The ontology provides structured information that enables scenes to be analyzed in terms of driving decisions and their related factors, which are intrinsically important for driving. During training, the framework uses the ontology to analyze scenes on which the model performs poorly and retrieves similar scenes for sampling. In this way, it adaptively optimizes model experience according to the model's learning state.

3.1 Dataset

To evaluate and retrieve scenes that the model needs to experience, our method uses an ontology that structures each scene in terms of its Driving Decision and

related Causal Factors. Figure 1 shows an overview of the annotation pipeline. Inspired by the Chain of Causation (CoC) dataset [31], the ontology consists of a Driving Decision and Causal Factors associated with it. The pipeline automatically annotates these elements using a large VLM (Vision Language Model) [2] with camera data, ego-vehicle status, and Meta Actions. The annotation process consists of four stages. First, Meta Actions are extracted from each sequence. Next, Causal Factors are annotated from the scene history, followed by the annotation of Driving Decisions using the previously obtained information. Finally, these elements are structured into an ontology.

Pre-processing Each sample in the database is segmented into 8-second sequences at 2 Hz using a sliding window with a one-frame stride. The frame at time step $t + 4$ is used as the keyframe, with frames t to $t + 3$ as historical information and frames $t + 4$ to $t + 15$ as future information. Camera data, together with ego-vehicle status and trajectories sampled at 10 Hz, are used as inputs. Each sequence is also annotated with longitudinal and lateral Meta Actions. Meta Actions are low-level labels that describe an ego-vehicle motion and are selected from a predefined closed set using rules based on an ego trajectory, steering angles, velocities, and accelerations. The label definitions are listed in Table 1.

Causal Factor Causal Factors are important elements potentially relevant to future driving decisions. They are predicted in an open-ended manner from historical information and Meta Actions. To avoid redundant or irrelevant outputs, we provide representative categories: critical objects, traffic lights, yield/stop controls, road events, lanes/lane lines, routing intents, and operational design domain (ODD) constraints. The VLM is instructed to describe each factor concisely by its name, location, and status. It prioritizes immediate causes over background conditions or general observations, and outputs `None` when no relevant factor exists.

Driving Decision A Driving Decision represents the driver's future maneuver. Given future information, Meta Actions, and Causal Factors, the VLM predicts one longitudinal and one lateral decision from their respective predefined label sets. The label definitions are listed in Table 1. The prediction proceeds in two steps. First, the VLM selects the three most important Causal Factors, referred to as the Top Causes. It then focuses on these factors to predict safe and accurate longitudinal and lateral decisions.

Ontology We define the predicted Driving Decisions as higher-level concepts and the Causal Factors as lower-level concepts, and represent their semantic and structural relationships in an ontology. Each Driving Decision is connected to all of its corresponding Top Causal Factors. The validity of the Driving Decisions and Causal Factors in the constructed Ontology is reviewed by experts, and anomalous annotations are manually corrected. The Ontology represents the event that can be experienced in a scene as a semantic structure, which enables adaptively sampling based on driving-relevant events that are difficult to capture through visual similarity alone.

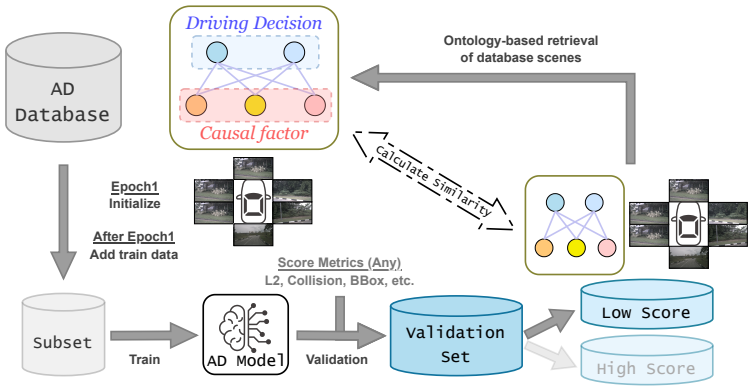

Fig. 2: Proposed method overview.

3.2 Training Framework based on Decision-Guided Adaptively Data-Sampling

We employ adaptively sampling to appropriately optimize the model's experience. Figure 2 illustrates an overview of the proposed method. Our method leverages the Ontology annotated in the dataset, which consists of Driving Decisions and Causal Factors. The Ontology defined in this study enables each scene to be evaluated from the perspective of the events that the model can experience. During the validation phase of each training epoch, we therefore perform scene retrieval for samples on which the model exhibits low performance, using their annotated Ontologies as queries, and sample additional scenes based on their similarity. By adding the scenes required by the model according to its current learning status in each loop, the proposed method optimizes its training exposure and enables efficient performance improvement.

Algorithm The proposed framework follows the cycle described below. First, a data budget B is predefined, and additional samples are progressively added until this budget is reached. For the first epoch, to avoid performance degradation caused by an excessively small initial training set, we construct the initial training subset based on the occurrence frequencies of the Driving Decisions. Specifically, we calculate the occurrence frequency of each Driving Decision in the full dataset, rank the samples in ascending order of their corresponding decision frequencies, and select a predefined proportion of the full dataset.

Next, during the validation phase, a score S is calculated for every sample in the validation dataset. This score is computed as a weighted average of the evaluation metrics associated with multiple model outputs. It should be noted that the definition of this score can be modified according to the outputs provided by the autonomous driving model. Samples whose scores exceed a predefined threshold are regarded as low-accuracy samples, and their annotated Ontologies are used as retrieval queries. Each Ontology is converted into a discrete feature vector, and its cosine similarity to each sample in the training database is calculated.

Algorithm 1 Adaptively sampling using an ontology

Require: Full dataset $\mathcal{D}$, validation set $\mathcal{V}$, untrained autonomous driving model $f_{\boldsymbol{\theta}}$, total number of epochs E, and data budget B
Ensure: Trained model $f_{\boldsymbol{\theta}}$

$\mathcal{S} \leftarrow \emptyset$
for $e = 1$ **to** E **do**
  if $|\mathcal{S}| > B$ **then**
    continue
  end if
  if $e = 1$ **then**
    $\mathcal{S} \leftarrow \text{SelectRareDecisionSamples}\,(\mathcal{D}, 0.1|\mathcal{D}|)$ ▷ Select 10% in ascending order of decision frequency
  else
    $\mathcal{Q} \leftarrow \text{SelectLowAccuracySamples}\,(f_{\boldsymbol{\theta}}, \mathcal{V})$
    $\mathcal{A} \leftarrow \text{OntologySearch}\,(\mathcal{Q}, \mathcal{D} \setminus \mathcal{S})$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{A}$
  end if
  $\boldsymbol{\theta} \leftarrow \text{Train}\,(f_{\boldsymbol{\theta}}, \mathcal{S})$
end for
return $f_{\boldsymbol{\theta}}$

The top-k samples with the highest similarities are then added to the training data for the second and subsequent epochs. The sample-selection process $\mathcal{A}_i^{(e)}$ can be formulated as shown in Eq. 1. Here, f denotes the embedding model, $\mathcal{O}_i$ denotes an Ontology, D denotes a dataset, and x denotes a data sample.

$$\begin{aligned} \mathbf{z}_i &= f_{\text{emb}}(\mathcal{O}_i), \quad \text{sim}(i,j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2}, \\ \mathcal{A}_i^{(e)} &= \text{TopK}_{x_j \in \mathcal{D}_{\text{pool}} \setminus \mathcal{D}_{\text{train}}^{(e-1)}} \left(\text{sim}(i,j),\, k\right), \end{aligned} \tag{1}$$

Through this process, training experiences that are insufficiently represented in the model are retrieved and added to the training set. The complete procedure of the proposed framework is presented in Algorithm 1.

Ontology Retrieval The Ontology represents a scene by hierarchically structuring a Driving Decision as a high-level concept and its associated Causal Factors as lower-level concepts. For retrieval, the Ontology is instantiated as a Knowledge Graph containing entities and relations, which explicitly preserves the hierarchical and semantic relationships among the elements. The Ontology is defined as follows.

$$\mathcal{O} = (\mathcal{C}, \mathcal{R}), \tag{2}$$

Here, $\mathcal{C}$ denotes the set of concepts, including Driving Decisions, Causal Factors, and their attributes, while $\mathcal{R}$ denotes the set of relations defined between these concepts. As the Driving Decision d_i of each scene s_i is selected from a closed-set label space, it can be directly converted into a single entity. In contrast, each

Causal Factor c_i is annotated in an open-ended manner and may contain multiple components, such as its location and status. We therefore use an VLM [2] to extract a normalized name, attributes, and relations from each Causal Factor. The normalized name is represented as a high-level entity, while its attributes are represented as lower-level entities in the Knowledge Graph. Let c_{ij} denote the j-th Causal Factor in scene s_i. The structuring process performed by the LLM is formulated as follows.

$$\Phi_{\mathrm{VLM}}\left(c_{ij}\right) = \left(n_{ij}, \{(r_{ijm}, a_{ijm})\}_{m=1}^{M_{ij}}\right), \tag{3}$$

Here, n_{ij} denotes the normalized name of the Causal Factor, a_{ijm} denotes an attribute value such as its location or status, $r_{ijm} \in \mathcal{R}$ denotes the relation between the Causal Factor and its attribute, and M_{ij} denotes the number of extracted attributes. The Driving Decision entity and each Causal Factor name entity are connected by a "has" relation. The Knowledge Graph $\mathcal{T}_i$ representing scene s_i is defined as follows.

$$\begin{aligned}\mathcal{T}_i = \ & \{(d_i, \mathrm{has}, n_{ij})\}_{j=1}^{N_i} \\ & \cup \{(n_{ij}, r_{ijm}, a_{ijm})\}_{j=1,\ldots,N_i, m=1,\ldots,M_{ij}},\end{aligned}$$

Here, N_i denotes the number of Causal Factors contained in a scene s_i. Furthermore, to ensure data diversity, the Knowledge Graph is divided into Subgraphs for each Driving Decision and used for retrieval. Each scene s_i is assigned two Driving Decisions, which are denoted by

$$\boldsymbol{d}_i = \{d_i^{(Longitudinal)}, d_i^{(Lateral)}\} \tag{4}$$

The Knowledge Graph Subgraph rooted at the p-th Driving Decision in a scene s_i, denoted by $\mathcal{T}_i^{(p)}$, is then defined as follows:

$$\mathcal{T}_i^{(k)} = \left\{\left(d_i^{(k)}, \mathrm{has}, c_{ij}\right)\right\}_{j=1}^{N_i}. \tag{5}$$

4 Experiments

To demonstrate the effectiveness of the proposed method, we conduct experiments using the proposed training framework. For fair evaluation, we adopt the nuScenes dataset which is widely used.

4.1 Baseline Methods

To the best of our knowledge, no existing method performs adaptive data sampling specifically for training end-to-end autonomous driving models in the same manner as our approach. Therefore, we select active learning-based methods as baselines [19, 20, 24, 25, 37]. Following prior work [19], we use ST-P3 [12], UniAD [13], and VAD [3] as the autonomous driving models for comparison. In addition, to ensure fair comparison with prior works, we adopt VAD-Tiny, a lightweight variant of VAD, as the base model for our framework.

Table 2: Quantitative evaluation results

Base Model	Percent	Method	Average L2				Average Collision			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
ST-P3	100%	-	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD		-	0.42	0.64	0.91	0.67	-	-	-	-
VAD-Base		-	0.39	0.66	1.01	0.69	0.08	0.16	0.37	0.20
VAD-Tiny		-	0.38	0.68	1.04	0.70	0.15	0.22	0.39	0.25
VAD-Tiny	10%	Random	0.51	0.83	1.23	0.86	0.40	0.62	0.98	0.67
		ActiveFT [37]	0.54	0.88	1.29	0.90	0.20	0.41	0.81	0.47
		ActiveAD [19]	0.47	0.80	1.21	0.83	**0.13**	**0.35**	0.80	**0.43**
		Ours	**0.45**	**0.75**	**1.13**	**0.78**	0.30	0.45	**0.71**	0.48
VAD-Tiny	20%	Random	0.49	0.80	1.17	0.82	0.36	0.49	0.77	0.54
		Coreset [24]	0.48	0.78	1.16	0.81	0.20	0.40	0.69	0.43
		VAAL [25]	0.54	0.89	1.31	0.91	0.17	0.38	0.66	0.40
		KECOR [20]	0.47	0.82	1.23	0.84	0.23	0.41	0.69	0.44
		ActiveFT [37]	0.50	0.82	1.21	0.84	0.27	0.42	0.63	0.44
		ActiveAD [19]	**0.44**	**0.73**	**1.10**	**0.76**	0.18	**0.36**	**0.62**	**0.39**
		Ours	0.45	0.75	1.13	0.78	**0.16**	0.50	0.67	0.44
VAD-Tiny	30%	Random	0.45	0.76	1.12	0.78	0.17	0.30	0.63	0.37
		Coreset [24]	0.43	0.71	1.06	0.73	0.43	0.51	0.68	0.54
		VAAL [25]	0.46	0.79	1.19	0.81	0.18	0.33	0.54	0.35
		KECOR [20]	0.46	0.78	1.22	0.82	0.22	0.43	0.70	0.45
		ActiveFT [37]	0.46	0.76	1.13	0.78	0.18	0.35	0.63	0.39
		ActiveAD [19]	0.41	0.66	0.97	0.68	**0.10**	**0.18**	**0.36**	**0.21**
		Ours	**0.36**	**0.60**	**0.92**	**0.63**	0.11	0.25	0.46	0.27

4.2 Implementation details

In our experiments, the low-performance score threshold was fixed at 0.5 based on preliminary experiments. To prevent label leakage, the score was calculated using confidence scores associated with the L2, collision, BBox, and map evaluation components. Each confidence score was normalized from min to max and the normalized values were combined into a weighted composite score. During sampling, cases were processed as queries in ascending order of their low-performance scores, beginning with the lowest-scoring case. For each query, candidate samples were added to the training dataset in descending order of similarity. The sampling procedure was terminated once the predefined data budget was reached. To disentangle the effect of representation learning, we represent the subgraphs as discrete feature vectors instead of employing a learned graph embedding model.

4.3 Quantitative Evaluation

The quantitative evaluation results are presented in Table 2. Table 2 reports the performance obtained when the training-data budget is set to 10%, 20%, and 30% of the full dataset. The proposed method achieves performance comparable to that of prior methods across all data-budget settings. In particular, under the 10% and 30% data budgets, our method outperforms the previous methods in terms of L2 error. These results demonstrate the effectiveness of the proposed sampling strategy.

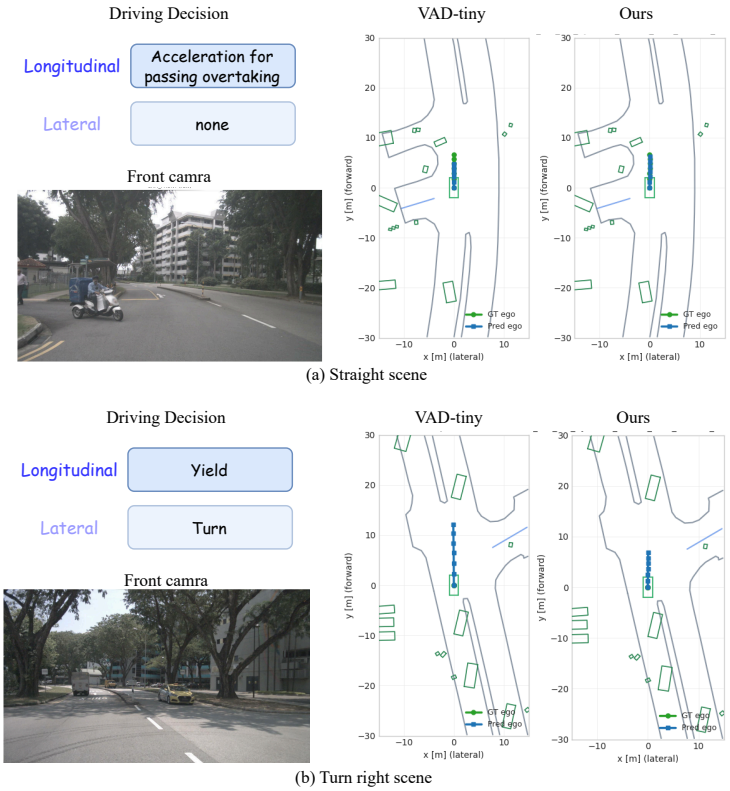

Fig. 3: Qualitative comparison results. For the bird's-eye-view bounding boxes and polylines, we use the ground truth to ensure a fair comparison.

Notably, VAD-Tiny which trained using our proposed framework achieves higher L2 error than VAD-Base trained on 100% of the available data. This result suggests that optimizing the model's training exposure through decision-aware adaptively data sampling can improve training efficiency and enables a lightweight model to achieve competitive or even superior planning performance with substantially less training data.

4.4 Qualitative Evaluation

Figure 3 presents the qualitative evaluation results on long-tail scenes. We compare VAD-Tiny which trained using the full dataset, with the proposed method using a 30% data budget, as shown in Table 2. In the straight-driving scenario shown in (a), the longitudinal driving decision is "Acceleration for passing overtaking". The ego vehicle should accelerate to pass a motorcycle ahead on the left that is yielding. A comparison of the predicted trajectories in the bird's-eye view shows that the standard VAD-Tiny decelerates in response to the motorcycle. In

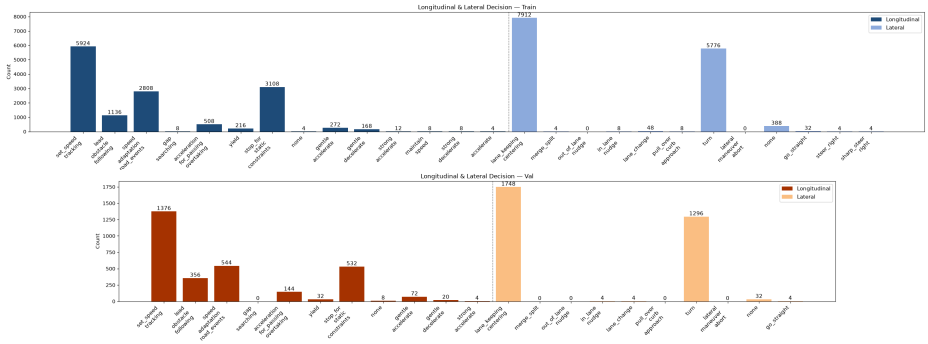

Fig. 4: Distribution of nuScenes. The top row shows the training split, the bottom row shows the validation split, and the left side shows the vertical decision class, while the right side shows the horizontal decision class.

contrast, our method predicts a trajectory that accelerates, passes the motorcycle, and continues straight, consistent with the ground truth. In the right-turn scenario shown in (b), the longitudinal and lateral driving decisions are “Yield" and “Turn", respectively. Although the ego vehicle should yield to an oncoming vehicle, the standard VAD-Tiny predicts a trajectory that accelerates and continues straight. On the oher hand, our method predicts a slowly advancing trajectory that accounts for the oncoming vehicle. In each scenario, it is important to recognize surrounding factors and to make corresponding driving decisions, whereas our model performs appropriate inference. These results suggest that the optimization of model experience by the proposed method contributes to improved robustness.

4.5 Analysis of Data Distribution

In this experiment, we analyze the distribution of nuScenes annotated using our pipeline and that of the final dataset after training. Figure 4 shows the data distribution of nuScenes. The class-wise distribution of the entire dataset clearly exhibits substantial imbalance. In particular, some lateral decision classes contain no samples in either the training or validation splits. These observations demonstrate severe imbalance in nuScenes, scarcity of long-tail scenes, and substantial gaps in class coverage.

Next, Figure 5 presents an analysis of the training data sampled by the proposed framework. The initial training set is constructed by preferentially sampling scenes associated with underrepresented driving decisions. As shown in Figure 5, this initial sampling stage successfully collects rare and long-tail scenes. At the end of training, the framework also samples classes corresponding to common driving scenes that were underrepresented in the initial training set, such as “set speed tracking", “lane keeping centering", and “stop for static constraints". This suggests that training exclusively on long-tail scenes is insufficient

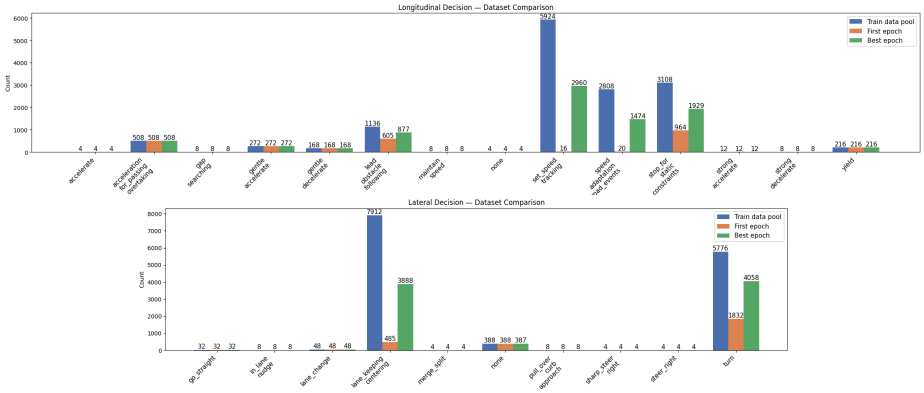

Fig. 5: Distribution of the training data sampled using the proposed method. The top row shows the Longitudinal decision, and the bottom row shows the Lateral decision. For each class, the graphs show, from left to right, the total number of samples in the data pool, the number of samples in the initial stage, and the number of samples after training. This data corresponds to a model trained with the 30% data budget shown in Table 2.

to improve performance on common scenes and that a more comprehensive data distribution is required to improve overall performance.

As shown in Figure 3 and Table 3, our method improves performance on both the overall evaluation set and long-tail scenes in nuScenes. These results demonstrate the effectiveness of optimizing the model's experience through adaptively sampling to improve performance on long-tail scenarios that are important for driving safety.

4.6 Long-Tail Scene Evaluation

Table 3 reports the quantitative evaluation results focusing on long-tail scenes. Based on the distribution shown in Figure 5, we restrict the evaluation to classes that contain no more than 300 samples in the training set and are also present in the validation set. Our method outperforms both VAD-Base and VAD-Tiny in terms of L2 error and collision rate. These results suggest that training on the entire nuScenes dataset without addressing its substantial data imbalance can limit performance on long-tail scenes, further demonstrating the effectiveness of our method.

5 Ablation Study

To evaluate the effectiveness of the proposed method, we conduct ablation experiments in which several components are restricted.

Table 3: Quantitative evaluation results on long-tail scenarios. The evaluation is restricted to classes with 300 or fewer samples in the training set that are also present in the validation set.

	L2			Collision		
	1s	2s	3s	1s	2s	3s
VAD-Base	0.88	1.46	2.08	**0.00**	0.48	1.60
VAD-Tiny	0.80	1.34	1.89	0.32	0.96	1.07
Ours (30%)	**0.54**	**0.92**	**1.35**	**0.00**	**0.47**	**1.06**

Table 4: Results of modifying the retrieval query construction method in the proposed framework. Data budget is 30%.

Model	Query		L2			Collision		
	Graph	Ontology	1s	2s	3s	1s	2s	3s
VAD-Tiny	Subgraph	Only Driving decision	0.46	0.75	1.11	0.65	0.52	0.90
	Fullgraph	-	0.45	0.76	1.16	0.74	0.94	1.19
	Subgraph	All	**0.36**	**0.60**	**0.92**	**0.11**	**0.25**	**0.46**

5.1 Comparison of Retrieval Methods

We quantitatively compare our full framework with two variants: one that uses the entire ontology graph as a retrieval query without decomposing it into subgraphs, and the other that uses only the driving decision instead of the ontology. The results are reported in Table 4. Both variants perform worse than the proposed framework. Using the entire graph makes the retrieval criteria overly restrictive. Consequently, the diversity of the training data. In contrast, using only the driving decision produces an overly coarse query that cannot sufficiently distinguish relevant scenes, resulting in a more biased training distribution. By constructing subgraphs from the ontology and using them as retrieval queries, the proposed method can sample relevant data while preserving data diversity.

5.2 Comparsion of Hypter-Parameter

Table 5 reports the performance obtained by varying Top-k, the number of samples retrieved for each target scene. The setting used in our experiments, $k = 10$, achieves the best performance, whereas increasing or decreasing k results in a substantial performance drop. This parameter directly affects how quickly the training set reaches the data budget. A larger k adds more samples at each sampling step, causing the data budget to be reached with fewer updates and reducing the diversity of the sampled data. In contrast, a smaller k slows the growth of the training set, resulting in unstable performance even during the middle stages of training.

6 Conclusion & Future Work

Large-scale autonomous driving datasets often contain substantial redundancy, resulting in biased data distributions that make it difficult for models to experi-

Table 5: Results when varying the value of the hyperparameter $TopK$. Data budget is 30%.

Model	Parameter	L2			Collision		
		1s	2s	3s	1s	2s	3s
VAD-Tiny	$k = 5$	0.41	0.68	1.05	0.21	0.35	0.58
	$k = 10$	**0.36**	**0.60**	**0.92**	**0.11**	**0.25**	**0.46**
	$k = 15$	0.45	0.75	1.13	0.30	0.46	0.72

ence and learn from safety-critical scenarios. To address this issue, we proposed a training framework that optimizes the experience of autonomous driving models through ontology-based adaptively sampling. First, each scene is structured as an ontology consisting of a driving decision and the causal factors relevant to that decision. This ontology enables scenes to be analyzed in terms of driving-relevant factors that cannot be adequately captured by visual similarity alone, thereby identifying gaps in the model's experience. The framework then adaptively updates the training data according to the model's learning status, allowing it to acquire the experience required to improve its performance. Our experiments show that models trained using the proposed framework perform comparably to or better than those trained using existing methods. Moreover, the proposed method achieves better L2 error than those that trains models on the entire dataset, demonstrating the effectiveness of optimizing model experience.

Our framework also offers several directions for future work. Although this study is limited to a baseline evaluation on nuScenes, the results suggest that its effectiveness could be further improved by expanding the data pool and designing more appropriate evaluation distributions. In addition, combining our experience-optimization framework with generative models, such as diffusion models [9, 34, 36], could enable the targeted generation of scenes in which the model lacks experience, thereby supporting efficient data augmentation and further improving robustness.

7 Limitation

Nevertheless, the proposed method has several limitations. First, we have not investigated its extension to closed-loop training or reinforcement learning. As optimizing model experience may also be beneficial in these settings, evaluating such extensions remains an important direction for future work. We have also not evaluated the applicability of our method to vision-language-action (VLA) models. Recent VLA models leverage strong scene-understanding capabilities and common sense of vision-language models (VLMs) [1, 2, 30] and have demonstrated strong generalization capabilities [8, 17, 31, 38]. Although these models perform well on some long-tail scenarios, they have not yet achieved sufficiently comprehensive and reliable performance for real-world deployment. In principle, our framework can be adapted to different model architectures by modifying the score used to identify low-performing samples. We therefore plan to evaluate its applicability to VLA models in future work.

References

1. Azzolini, N.A., Bai, J., Brandon, H., Cao, J., Chattopadhyay, P., et al.: Cosmos-reason1: From physical common sense to embodied reasoning. arXiv:2503.15558 (2025)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., et al.: Qwen3-vl technical report. arXiv:2511.21631 (2025)
3. Bo, J., Shaoyu, C., Qing, X., Bencheng, L., Jiajie, C., Helong, Z., Qian, Z., Wenyu, L., Chang, H., Xinggang, W.: Vad: Vectorized scene representation for efficient autonomous driving. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Das, S., Patibandla, H., Bhattacharya, S., Bera, K., Ganguly, N., Bhattacharya, S.: Thresholded multi-criteria online subset selection for data-efficient autonomous driving. International Conference on Computer Vision (ICCV) (2021)
6. Feng, L., Bahari, M., Amor, K.M.B., Éloi Zablocki, Cord, M., Alahi, A.: Unitraj: A unified framework for scalable vehicle trajectory prediction. arXiv:2403.15098 (2024)
7. Feuer, B., Xu, J., Cohen, N., Yubeaton, P., Mittal, G., Hegde, C.: Select: A large-scale benchmark of data curation strategies for image classification. Neural Information Processing Systems (NeurIPS) (2024)
8. Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. International Conference on Computer Vision (ICCV) (2025)
9. Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q.: Magicdrive: Street view generation with diverse 3d geometry control. Conference on Learning Representations (ICLR)
10. Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A.S., Hauswald, L., Pham, V.H., Mühlegg, M., Dorn, S., Fernandez, T., Jänicke, M., Mirashi, S., Savani, C., Sturm, M., Vorobiov, O., Oelker, M., Garreis, S., Schuberth, P.: A2d2: Audi autonomous driving dataset. arXiv:2004.06320 (2020)
11. Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., Co-Reyes, J.D., Agarwal, R., Roelofs, R., Lu, Y., Montali, N., Mougin, P., Yang, Z., White, B., Faust, A., McAllister, R., Anguelov, D., Sapp, B.: Waymax: An accelerated, data-driven simulator forlarge-scale autonomous driving research. arXiv:2310.08710 (2023)
12. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. European Conference on Computer Vision (ECCV) (2022)
13. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., Li, H.: Planning-oriented autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
14. Jaeger, B., Chitta, K., Geiger, A.: Hidden biases of end-to-end driving models. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)

15. Karnchanachari, N., Geromichalos, D., Tan, K.S., Li, N., Eriksen, C., Yaghoubi, S., Mehdipour, N., Bernasconi, G., Fong, W.K., Yiluan Guo, H.C.: Towards learning-based planning:the nuplan benchmark for real-world autonomous driving. International Conference on Robotics and Automation (ICRA) (2024)
16. Kishida, I., Nakayama, H.: Empirical study of easy and hard examples in cnn training. International Conference on Neural Information Processing (ICONIP) (2019)
17. Li, Y., Zhou, L., Yan, S., Liao, B., Yan, T., Xiong, K., Chen, L., Xie, H., Wang, B., Chen, G., Ye, H., Liu, W., Sun, H., Wang, X.: Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. arXiv:2604.02190 (2026)
18. Lis, K., Kryjak, T.: Comparative study of subset selection methods for rapid prototyping of 3d object detection algorithms. 27th International Conference on Methods and Models in Automation and Robotics (MMAR) (2023)
19. Lu, H., Jia, X., Xie, Y., Liao, W., Yang, X., Yan, J.: Activead: Planning-oriented active learning for end-to-end autonomous driving. arXiv:2403.02877 (2024)
20. Luo, Y., Chen, Z., Fang, Z., Zhang, Z., Baktashmotlagh, M., Huang, Z.: Kecor: Kernel coding rate maximization for active 3d object detection. In International Conference on Computer Vision (ICCV) (2023)
21. Luo, Y., Chen, Z., Wang, Z., Yu, X., Huang, Z., Baktashmotlagh, M.: Exploring active 3d object detection from a generalization perspective. Conference on Learning Representations (ICLR) (2023)
22. Naumann, A., Gu, X., Dimlioglu, T., Bojarski, M., Degirmenci, A., Popov, A., Bisla, D., Pavone, M., Muller, U., Ivanovic, B.: Data scaling laws for end-to-end autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Autonomous Driving (WAD) (2025)
23. Roque, G., Maquiling, E., Lopez, J.G.T., Greer, R.: Automated data curation using gps & nlp to generate instruction-action pairs for autonomous vehicle vision-language navigation datasets. arXiv:2505.03174 (2025)
24. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In International Conference on Learning Representations (ICLR) (2018)
25. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In International Conference on Computer Vision (ICCV) (2019)
26. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. arXiv:2405.19620 (2024)
27. Tian, H., Li, T., Liu, H., Yang, J., Qiu, Y., Li, G., Wang, J., Gao, Y., Zhang, Z., Wang, L., Ye, H., Tan, T., Chen, L., Li, H.: Simscale: Learning to drive via real-world simulation at scale. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
28. Trinh, L., Anwar, A., Mercelis, S.: Data selection method for assessment of autonomous vehicles. International Conference on Intelligent Transportation Systems (ITSC) (2024)
29. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
30. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., et al.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv:2508.18265 (2025)

31. Wang, Y., Luo, W., Bai, J., Cao, Y., et al.: Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. arXiv:2511.00088 (2025)
32. Wang, Y., Cheng, K., He, J., Wang, Q., Dai, H., Chen, Y., Xia, F., Zhang, Z.: Drivingdojo dataset: Advancing interactive and knowledge-enriched driving world model. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS (2024)
33. Wang, Z., Nichani, E., Bietti, A., Damian, A., Hsu, D., Lee, J.D., Wu, D.: Learning compositional functions with transformers from easy-to-hard data. Conference on Learning Theory (COLT) (2025)
34. Wen1, Y., Zhao, Y., Liu2, Y., Jia, F., Wang, Y., Luo, C., Zhang, C., Wang, T., Sun, X., Zhang, X.: Panacea: Panoramic and controllable video generation for autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
35. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021) (2021)
36. Xiaofan, L., Yifu, Z., Ye, X.: Drivingdiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. European Conference on Computer Vision (ECCV) (2024)
37. Xie, Y., Lu, H., Yan, J., Yang, X., Tomizuka, M., Zhan, W.: Active finetuning: Exploiting annotation budget in the pretraining-finetuning paradigm. In IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) (2023)
38. Zhou, X., Han, X., Yang, F., Ma, Y., Tresp, V., Knoll, A.: Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. arXiv:2503.23463 (2025)