2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MoE-Powered Fast VLMs via Curriculum Learning-based Knowledge Distillation: Taming Regular and Corner Cases in Autonomous Driving

Xue Zhao[1*], Zhou Fang[2]

***Abstract*— Autonomous driving has advanced significantly with the integration of large Vision-Language Models (VLMs), which excel in understanding and analyzing driving data. However, existing VLMs face challenges, particularly in terms of latency, which is crucial for real-time driving tasks. While shrinking the model size can reduce latency, it also limits the model's ability to handle both regular and corner cases effectively. To address this challenge, we propose the Curriculum Learning-based Knowledge Distillation (CLKD) framework. CLKD enhances student model performance through three key innovations: (1) integration of a Mixture-of-Experts (MoE) architecture to preserve model expressiveness; (2) Hardness-explored at Two Granularities (H2G), which dynamically identifies easy and difficult samples at both instance and feature levels; and (3) Progressive Release Distillation strategy that gradually reduces reliance on the teacher model, thereby fostering the student's autonomy and improving its generalization capability in complex driving scenarios. In real-world data experiments, CLKD has achieved a twofold increase in speed compared to existing approaches while maintaining comparable performance.**

I. Introduction

Autonomous driving has undergone rapid advancements in recent years. In particular, large Vision-Language Models (VLMs)[1], [2], [3], [4], [5], [6], [7], [8], [9] have introduced novel transformations within the realm of autonomous driving. VLMs excel at deeply understanding and analyzing vast amounts of driving data. The VLMs-based autonomous driving system effectively integrates data from various sensors to build a comprehensive understanding of the surrounding environment. This capability significantly improves the accuracy of the detection. In addition, VLMs improve autonomous vehicles with more intelligent and adaptable decision-making and planning capabilities, enabling them to respond dynamically to changing conditions.

Typically, VLMs serve as general purpose end-to-end models, performing tasks such as planning and perception through question-and-answer interactions [9]. Another more straightforward approach is to use a large language model to directly predict future trajectories or control signals for vehicles [10]. However, since VLMs are not ideally suited for precise numerical predictions [11], an alternative approach is to design a dual system, such as DriveVLM [7] and Senna [11], which combine the strengths of VLMs with traditional end-to-end models [12].

[1]Xue Zhao is with School of Computer Science, Shanghai Jiao Tong University, Dongchuan Road, Shanghai 200240, China. xuezhao0106@gmail.com
[2]Zhou Fang is with Shanghai University.
∗ Corresponding author.

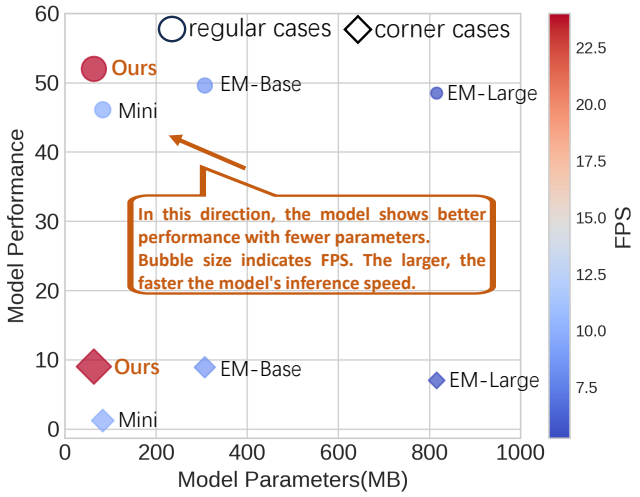

Fig. 1. The comparison results of the performance (on regular cases and corner cases), parameters, and FPS (Frames Per Second) of different models. The size of the bubbles represents different FPS. It can be seen that our model achieves the best balance between performance and efficiency.

Existing VLMs for autonomous driving, have demonstrated potential capabilities in handling autonomous driving tasks, but also exposed some problems that urgently need to be solved. Among them, the latency issue for the applications in the real world is particularly prominent. In particular, for the dual-system architecture, if there is a significant delay on the side of the large language model, it can reduce the overall efficiency of the system. This is particularly problematic for autonomous driving tasks with high real-time requirements. Therefore, **accelerating the VLMs** is crucial for improving the reliability and safety of the autonomous driving system and is a key step in the widespread application of autonomous driving technology. To address the latency problem of the VLM, an intuitive approach is to reduce the size of the model. Smaller models usually have a lower computational complexity and can complete inference tasks in a shorter time, thus reducing latency. However, this approach brings new problems. When the model size is reduced, its learning ability will also be limited. In autonomous driving, the situations to be dealt with can be divided into regular cases and corner cases [13], [14], [15]. Regular cases refer to the scenarios commonly encountered in daily driving, such as driving in accordance with traffic rules under normal road conditions. Corner cases refer to those rare scenarios that may have a significant impact on driving safety, such as road in severe weather or obscured traffic signs. Therefore, while simply shrinking the model can help alleviate the latency issue of the VLM to some extent, it comes at the cost of the model's

learning capacity, limiting its ability to effectively handle both regular cases and corner cases.

To effectively address these challenges, we simulate the gradual process of humans learning to drive and introduce an innovative Curriculum Learning-based Knowledge Distillation (CLKD) framework to achieve fast VLMs. In the pursuit of balancing efficiency and performance, we reduce the model size and utilize the proposed CLKD to achieve this objective. This framework not only accelerates the VLM but also preserves its learning capability, enabling it to accurately handle both regular and corner cases in autonomous driving. CLKD is primarily composed of three key components.

At first, to enable the student model to maintain strong expressive capabilities, we introduce Mixture-of-Experts (MoE) [16] into VLMs. When encountering samples of different difficulty levels, MoE will exhibit different activation states. Specifically, a MoE-adapter processes visual tokens to adapt to varying difficulties and align visual with textual tokens. These fused tokens are then fed into a large language model, where the Feed-Forward Neural Network (FFN) in the Transformer architecture is warped by MoE, allowing the student model to capture diverse multimodal knowledge during distillation [17], [18].

In addition, in the process of CLKD, there are two core issues: how to adaptively distinguish easy and difficult samples and how to achieve progressive learning. To tackle these issues, we first propose an online hardness mining strategy, Hardness-explored at Two Granularities (H2G), which explores the "hardness" of training set at both the token level and the sample level to comprehensively capture the complex information within the data. By leveraging this strategy, the model can automatically recognize hard examples during the training process, gradually improving its ability to handle corner cases. H2G ensures that the student model can adequately learn from both easy and difficult samples when there is an imbalance in the data distribution. Furthermore, we propose Progressive Release Distillation strategy which gradually reduce the teacher model's involvement in the distillation. The existing large-scale teacher models have shown outstanding capabilities in dealing with regular cases. Through distillation, this valuable knowledge can be transferred to the student models. In the initial stage, we leverage the teacher model to help the student model establish a solid knowledge foundation for regular cases. When the student model has gained proficiency with regular cases, we adjust the teaching approach. We gradually reduce the teacher model's influence and shift the focus to challenging samples. This transition enhances the student model's generalization ability to handle complex and difficult scenarios effectively. This distillation framework follows a progressive learning approach, helping the small scale model gradually master driving tasks of varying difficulty levels.

The main contributions of this paper are as follows:

- We introduce the Curriculum Learning-based Knowledge Distillation framework for fast VLMs. This framework strikes a balance between efficiency and performance, and enable the model to handle both regular and corner cases in autonomous driving.
- A MoE-Powered architecture is proposed to maintain the strong expressive capabilities of small scale model.
- Hardness-explored at Two Granularities is introduced to adaptively distinguish easy and difficult samples and ensures that the model can adequately learn from both easy and difficult samples when there is an imbalance in the data distribution.
- We propose Progressive Release Distillation Strategy, which gradually reduces the teacher model's involvement, initially guiding the student model in mastering regular cases and then shifting focus to challenging samples, ultimately enhancing the student model's ability to handle complex driving scenarios.
- Extensive experiments have demonstrated that CLKD has achieved a twofold increase in speed compared to existing approaches while maintaining comparable performance (See Figure 1).

II. Related Work

A. Large Vision-Language Models for Autonomous Driving

Large vision-language models for autonomous driving has attracted much attention recently. Senna [11], mainly focused on generating accurate driving paths in routine scenarios by combining VLMs and end-to-end autonomous driving model. Despite showing excellent performance in handling regular driving cases [19], it has limitations in curved road situations and struggles to adapt to highly complex and rare situations that deviate from typical driving conditions. CODA-LM focuses on evaluating large vision-language models for automated driving in corner cases, providing a new perspective for dealing with uncommon driving situations [20]. However, CODA-LM could not directly improve the model's ability to efficiently handle regular and corner situations at high speeds, which is crucial for real-time autonomous driving operations. By handling both regular and corner cases simultaneously, DriveVLM is introduced to overcome the limitations faced by the previous models. In regular driving situations, it aims to provide a more comprehensive scene understanding and decision-making capability than Senna. However, DriveVLM encountered speed problems during real-time operation, thus limiting its practical application [21]. Thus, DriveVLM is unable to effectively handle both regular situations and corner situations, which highlights a long-standing challenge in the field. In summary, existing large vision-language models designed for autonomous driving have not yet fully met the requirements of real-world driving. They typically exhibit one or more of the following drawbacks: either they are unable to handle both regular and corner situations, or they face significant challenges in terms of computational speed.

On the contrary, our proposed approach combines distillation-based acceleration techniques with the MoE-Powered method so that both regular and corner cases can be efficiently handled. With this approach, we have taken an important step forward in the field of autonomous driving. We have the potential to revolutionize how self-driving cars

handle diverse driving scenarios, which could significantly enhance their overall performance and practicality.

B. Knowledge Distillation for Large Vision-Language Models

Knowledge distillation (KD) plays a pivotal role in optimizing large Vision-Language Models (VLMs) for deployment in resource-constrained environments, such as autonomous vehicles and mobile devices. As the size of these models increases, their direct deployment becomes increasingly impractical due to significant computational and memory requirements. KD facilitates the transfer of essential knowledge from a large, high-performance teacher model (e.g., large language models, LLMs) to a smaller, more efficient student model, thereby maintaining high performance while reducing the model's size and computational burden. This is particularly critical in autonomous driving, where real-time decision-making depends on efficient scene understanding and the handling of complex interactions.

Recent studies [22] have demonstrated that KD can preserve the semantic capabilities of large models while making them more computationally feasible. Furthermore, methods like Align-KD, introduced in [23], extend KD by focusing not only on distilling individual modalities but also on the crucial cross-modal knowledge that enables effective integration of visual and textual information. This approach guides the training of smaller models to learn cross-modal alignments, significantly enhancing their performance on multiple benchmarks. Such advancements make VLMs more viable for real-world autonomous applications, including pedestrian behavior analysis and interaction management in complex traffic environments.

C. Curriculum Learning

Curriculum Learning [24] is a machine learning strategy in which a model is trained on tasks of increasing difficulty, starting with easier examples and gradually progressing to more complex ones. The key principle behind curriculum learning is that a structured learning process, beginning with simple tasks, allows the model to build useful representations, facilitating better performance on harder tasks. This approach mirrors human learning, where people often master fundamental concepts before tackling more advanced topics. Using this strategy, the model can achieve faster convergence and better generalization, especially when confronted with complex real-world data.

III. Methodology

A. Overview

The overall pipeline of CLKD is illustrated in Figure 2. A MoE-Powered architecture is proposed to maintain the strong expressive capabilities of small scale model. Hardness-explored at Two Granularities is introduced to adaptively distinguish easy and difficult samples and ensures that the model can adequately learn from both easy and difficult samples when there is an imbalance in the data distribution. Progressive Release Distillation Strategy gradually reduces the teacher model's involvement, initially guiding the student model in mastering regular cases and then shifting focus to challenging samples, ultimately enhancing the student model's ability to handle complex driving scenarios.

B. MoE-Powered Fast VLMs

To enable the student model to handle a wide range of scenarios comprehensively and flexibly, we introduce MoE [16] into the VLMs. This module in the VLMs operates through a unique mechanism: for regular cases, the model activates only a small number of experts, ensuring efficient task completion. However, when faced with more complex or corner cases, the system automatically engages additional experts to ensure the model maintains optimal performance.

We first introduce a MoE-adapter in the visual modality to perform adaptive processing on these tokens, since visual tokens span scenes of varying difficulty levels. This approach allows the model to respond efficiently to visual tokens of different complexities while simultaneously ensuring a better alignment between visual and textual tokens. We take the original visual adapter module as experts and set up N experts. We first compute the routing weight W of N experts:

$$W = \text{Softmax}(\text{FC}(x_v)), \tag{1}$$

where x_v is the visual token and FC is a fully connected layer. Then, we calculate the weighted sum of N experts with their corresponding routing values:

$$Output = \sum_{i=1}^{N} W_i \cdot \text{Expert}(x_v). \tag{2}$$

Secondly, after the visual and textual tokens are aligned and fused, they are passed into a large language model for further training. To enable the student model to flexibly handle various scenarios, we replace the Feedforward Neural Network (FFN) layer in the Transformer architecture with MoE [17], [18]. Besides, this design of MoE structure into VLMs also maintain the ability of student model to capture complex multimodal knowledge through sparsely activated experts during distillation [25]. The detailed network structure is presented in Figure 2.

C. Hardness-explored at Two Granularities (H2G)

Given that the proposed CLKD conforms to a progressive learning approach, facilitating the model to gradually acquire the ability to handle samples of varying difficulty levels, this raises a key issue. It involves how the model can adaptively distinguish between simple and difficult samples. Besides, in the existing autonomous driving datasets, there is a severe imbalance in the proportion of easy and difficult samples. It is essential to ensure that the student model can adequately learn from both easy and difficult samples.

To effectively address above challenges, we propose an online hardness mining strategy, namely Hardness-explored at Two Granularities (H2G) in the training process. This strategy delves into the "hardness" of samples at two granularities: the token level and the sample level. Learning from only one dimension might not comprehensively capture the complex information within the data, especially for large-language

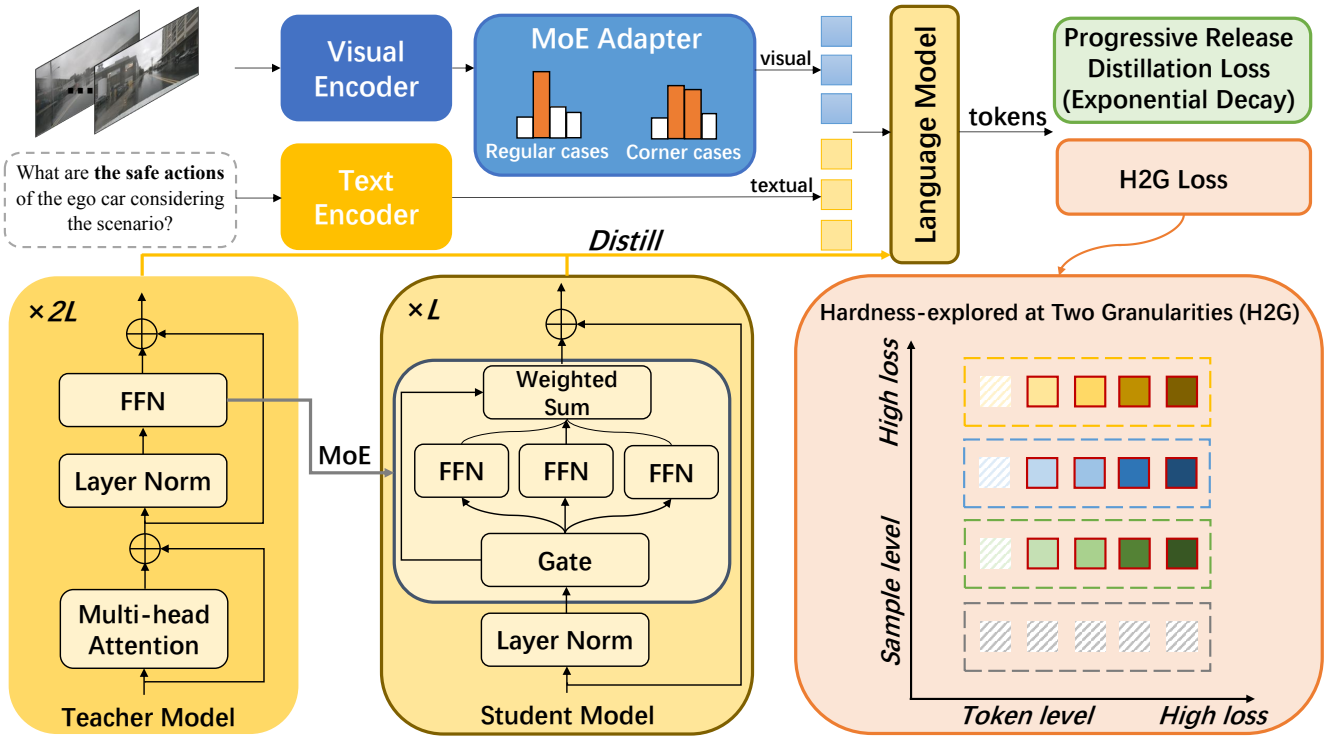

Fig. 2. The Curriculum Learning-based Knowledge Distillation (CLKD) framework for fast VLMs. CLKD integrates Mixture-of-Experts to maintain the student model's expressiveness, introduces Hardness-explored at Two Granularities (H2G) for distinguishing easy and difficult samples, and Progressive Release Distillation strategy to gradually reduce the teacher model's involvement, thus enhancing the student model's generalization ability to handle complex driving scenarios.

models with multi-token inputs. Therefore, by considering both the token and sample levels, we can more comprehensively consider the characteristics of the data and give full play to thoroughly explore the "hardness" in the training process.

Specifically, the proposed H2G is illustrated in Figure 2. At the sample level, for $x_i \in \mathcal{X}$, the corresponding loss value $L(x_i)$ is calculated and all the losses are sorted in descending order. The top $m\%$ samples with the highest loss values are selected as hard samples S_m.

$$S_m = \{x_i | L(x_i) \in \text{top } m\% \text{ of losses}\}. \tag{3}$$

Subsequently, within the set of selected hard samples, we further delve into the token level. For each token contained in every hard sample, its loss value is calculated, and then these losses are sorted again in descending order. Tokens whose loss values rank in the top $n\%$ are selected as hard tokens t_n. These tokens carry crucial and complex information and are key elements in the learning process.

$$t_n(x_i) = \{t_j | L(x_i, t_j) \in \text{top } n\% \text{ of losses within } x_i\}. \tag{4}$$

Finally, the model parameters are updated by minimizing the losses of these hard tokens:

$$Loss_{\text{H2G}} = L(x_i, t_n(x_i) | x_i \in S_m). \tag{5}$$

Via H2G, the model is guided to pay more attention to the information contained in the hardness, thereby enhancing the model's learning ability for complex samples and its generalization performance.

D. Progressive Release Distillation Strategy

At present, the existing large-scale teacher models have shown outstanding capabilities in dealing with regular cases. Through the technique of distillation, this valuable knowledge can be transferred to the student models. In this paper, we take these models as teacher models and propose Progressive Release Distillation Strategy. This strategy strengthens the guidance provided by the teacher model in the initial stage. The influence of the teacher model is gradually reduced to allow the student model to explore a more complex knowledge space, ultimately leading to the development of an efficient model capable of handling both regular and corner cases.

In the early phase of CLKD, the teacher model aims to provide the student model with a solid knowledge foundation. The teacher model, having been trained on a large dataset of regular cases, possesses rich knowledge and advanced pattern recognition capabilities. By amplifying the teacher's influence, the student model can quickly acquire accurate and reliable knowledge representations of regular cases. As the student model becomes more proficient in basic knowledge, excessive reliance on the teacher model may limit its generalization capabilities and ability to learn autonomously in complex scenarios. Thus, in the later phase, we reduce the

teacher's influence and allow the student model to navigate a more complex knowledge space independently, enhancing its ability to handle diverse situations. This strategy ultimately fosters better knowledge transfer and improves the model's adaptability to a wide range of samples.

Specifically, we set a hyperparameter α, to control the distillation loss. By annealing the value of α during training process, we can gradually reduce the involvement of the teacher in the distillation and enhance the student's autonomous learning of difficult scenarios. The distillation loss used in this paper is the KL divergence between the student's logits and the teacher's soft label. The overall training losses are as follows:

$$\alpha(t) = \alpha(0) \cdot \gamma^t, \tag{6}$$

$$Loss_{\text{train}} = Loss_{\text{H2G}} + \alpha(t) Loss_{\text{KD}}, \tag{7}$$

where t is the iteration step in the training process and γ is the decay rate.

IV. Experiments

A. Experiments Settings

1) Benchmarks and Datasets: To comprehensively evaluate our method, we first experiment on two existing types of VLM-based tasks in autonomous driving and then perform an accuracy and speed assessment in real-world scenarios.

Firstly, we evaluate on EM-VLM4AD [8] which performs subtasks such as prediction, planning, and perception through question-and-answer interactions. This task is based on the DriveLM dataset [6], which consists of regular cases. To further evaluate the performance on corner cases, we incorporate the CODA-LM dataset [26] that contains only corner cases into DriveLM dataset. EM-VLM4AD adopts T5 [5] large language model, which is encoder-decoder architecture based on the transformers. It combines the advantages of BERT and GPT. The encoder is used to understand the input text, and the decoder is used to generate the output text.

Secondly, we accelerate Senna [11], which is a dual-system structure, including Senna-VLM, a large-scale vision-language model that utilizes multi-image encoding methods and multi-view prompts to efficiently understand scenes and generate high-level planning decisions in natural language, and Senna-E2E, an end-to-end model responsible for predicting precise trajectories based on the high-level decisions generated by Senna-VLM. Senna mainly evaluates on nuScenes [27] dataset. In practice, Vicuna-v1.5-7b [28], a decoder-only architecture, is used as their LLM.

Thirdly, to further verify the practical application value of our acceleration method, we also conduct tests in the real world environment.

2) Evaluation metrics: To fully evaluate the performance of fast VLMs, we evaluate from two aspects. We first evaluate the quality of generated sequence with BLEU-4 [29], ROUGE-L [30], and METEOR [31]. BLEU-4 is a precision-based similarity measurement method for analyzing how many n-grams of words appear in the candidate translation compared to the reference translation. ROUGE-L is a similarity measurement method based on recall, primarily examining the adequacy and faithfulness of the reference translation. METEOR provides a more comprehensive assessment of generated sequence by considering precision, recall, synonym matching, stem matching, and word order. Secondly, we compare the parameter size of different methods and calculate the Frames Per Second (FPS) of different methods to evaluate the inference speed.

3) Implementation Details: The experiments are mainly implemented on RTX 3090 GPU. For the task on DriveLM and CODA-LM Datasets, in the training process, the visual encoder of Swin Transformer Tiny, and text encoding part are frozen. The language model is T5-Small [5]. The model is finetuned with LoRA [32]. Other modules are trained with an initial learning rate of 1e-4 and a weight decay of 0.05. The teacher model is trained for 6 epochs on the training set. The distillation of student model needs 3-4 epochs only. The input size is (224, 224). For the Senna task, the experimental data and training parameters are referenced from [11]. The distillation process only requires half the training time of Senna.

B. Main Results

1) Results on DriveLM and CODA-LM Datasets: For the task on DriveLM and CODA-LM, we compare our method with EM-VLM4AD [8], which employs ViT [33] as image encoder, T5-Base and T5-QLarge as language model, DriveLM-Agent [6] based on LLama-7B [4], and MiniDrive [9] which achieves state-of-the-art performance in terms of parameter size, with the smallest version containing only 83M parameters. We reproduce these methods with training on the datasets consisting of DriveLM and CODA-LM. The results of the comparative results on DriveLM and CODA-LM Datasets are shown in Table I. The experimental results clearly indicate that our model surpasses others in terms of speed, operating at twice the rate. Additionally, it achieves a 25% reduction in the number of parameters. In terms of performance, it delivers optimal results across most metrics, both under regular and corner cases.

2) Evaluation of Senna: Senna [11] is an integrated autonomous driving system that consists of two primary modules: Senna-VLM and Senna-E2E. Senna-VLM processes multiview image sequences and textual inputs, such as user instructions and navigation commands, to generate high-level planning decisions. These inputs are encoded into image and text tokens, which are then processed by a Large Language Model (LLM). The LLM produces high-level decisions that are subsequently encoded into meta-action features via the Meta-action Encoder. These metaaction features, along with the scene information, are passed to Senna-E2E, which generates the precise low-level planning trajectories required for vehicle movement.

The performance of different models in self-driving situations, especially their high-level planning abilities, is measured by several important metrics: Acc. (%) and F1 Score. **Acc. (%)** is used to figure out how accurate a model's high-level planning decisions are. It does this by comparing

TABLE I

COMPARATIVE RESULTS ON DRIVELM (**R**EGULAR CASES) AND CODA-LM DATASETS (**C**ORNER CASES). WE INCORPORATE THE CODA-LM DATASET [26] THAT CONTAINS ONLY CORNER CASES INTO DRIVELM DATASET. ALL THE MODELS ARE TRAINED ON THE MERGED DATASET."DIS." IS THE COMMONLY USED DISTILLATION STRATEGY (SOFT LABEL DISTILLATION).

Dataset	Method	BLEU-4(↑)	METEOR(↑)	ROUGE-L(↑)	FPS (imgs/s)	Params
DriveLM(**R**)	EM-VLM4AD-Base	49.6	36.4	74.1	8.5	307M
	EM-VLM4AD-QLarge	48.5	36.3	71.8	5.25	816M
	MiniDrive$_{224}$	46.1	34.1	71.2	9.75	83M
	Dis.	48.5	35.8	74.3	24	61.3M
	CLKD (Ours)	**52.0**	**38.1**	74.0	**24**	63.5M
CODA-LM(**C**)	EM-VLM4AD-Base	8.9	14.8	24.7	-	-
	EM-VLM4AD-QLarge	7.0	12.5	22.7	-	-
	MiniDrive$_{224}$	1.2	6.7	9.0	-	-
	Dis.	8.1	13.0	24.0	-	-
	CLKD (Ours)	**9.0**	13.9	**25.3**	-	-

TABLE II

ABLATION STUDY RESULTS ON **R**EGULAR AND **C**ORNER CASES. "DIS." IS THE COMMONLY USED DISTILLATION STRATEGY (SOFT LABEL DISTILLATION). "PR-DIS." IS THE PROGRESSIVE RELEASE DISTILLATION STRATEGY WE PROPOSED.

#	Model	Dataset	Dis.	MoE	H2G	PR-Dis.	BLEU-4(↑)	METEOR(↑)	ROUGE-L(↑)	Avg.(↑)
#1	Teacher	R					50.7	37.8	74.5	54.33
#2	Student	R					47.5	35.5	72.6	51.87
		C					7.9	13.7	23.3	14.97
#3	Student	R	✓				48.5	35.8	74.3	52.87
		C					8.1	13.0	24.0	15.03
#4	Student	R	✓	✓			51.8	38.0	74.6	54.80
		C					8.0	13.5	23.6	15.03
#5	Student	R	✓	✓	✓		51.8	38.0	73.6	54.45
		C					9.0	14.2	24.9	16.03
#6	Student	R		✓	✓	✓	52.0	38.1	74.0	54.70
		C					9.0	13.9	25.3	16.07

TABLE III

EVALUATION RESULTS OF SENNA [11] ON THE NUSCENES VALIDATION DATASET. "STRA." DENOTES "STRAIGHT". "STRA.", "KEEP" AND "STOP" ARE MEASURED WITH F1 SCORE. THE SPEED IS TESTED ON 3090 GPU.

Method	Acc.	stra.	keep	stop	Latency	Params
Senna	87.7%	98.0	95.2	80.6	0.36s	7B
+CLKD	88.3%	98.0	95.6	80.7	0.25s	3.5B

the model's predicted main actions (like "go straight", "turn left", "turn right", "speed up", "slow down", "stop") with the correct labels. **F1 Score** is used to check a model's skill in predicting the right side-to-side path decisions, such as "go straight", and speed-related decisions, like "maintain speed". The calculation of the $F1$ score for the path measure depends on precision and recall.

The evaluation results of Senna [11] are listed in Table III. We fine-tune Senna-VLM using the nuScenes dataset. Then, we use the trained model as a teacher model. With our proposed CKLD strategy, we obtain a more efficient student model. The model achieves comparable performance, with the number of parameters reduced by 50% and the inference speed increased by 1.44 times.

3) Evaluation in real-world autonomous driving scenarios: To comprehensively assess the performance and efficiency of our proposed method, we carry out in-depth evaluations of our acceleration strategy on real-world data. Drawing on industry practices, in the realm of autonomous driving, it is generally imperative that the entire workflow, spanning from sensor data acquisition to the vehicle's implementation of relevant maneuvers, is within 100 ms.

In light of this requirement, we introduce a novel evaluation metric named "Real-time Successful Decision-making" (RSD). This metric stipulates that the VLM should generate **accurate** decisions within **100 ms**. We gather 200 clips from actual scenarios, inference these samples with model in Sec.IV-B.1, and compute the Real-time Successful Decision-making Rate (RSDR). A more stringent threshold ($<$ 80 ms) is set to better reflect the speeds of different methods. RSD and RSDR are calculated as follows:

$$RSD = \mathbb{I}(t_i \leq 80) \times a_i, \tag{8}$$

$$RSDR = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbb{I}(t_i \leq 80) \times a_i\right], \tag{9}$$

where t_i denotes the time taken by the VLM to process the $i-th$ sample; $a_i = 1$ if the meta-action made by the model is

TABLE IV

EVALUATION RESULTS IN REAL-WORLD AUTONOMOUS DRIVING SCENARIOS. THE SPEED IS TESTED ON 3090 GPU.

Method	Params (MB)	RSDR(%)
EM-VLM4AD-Base	307	16.8
EM-VLM4AD-Small	94	49.1
CLKD (Ours)	63.5	76.9 (↑ 27.8)

accurate; and n is the total number of samples. The evaluation of meta-action is referred to Senna [11].

The performance of our acceleration strategy on real-world data is shown in the Table IV. Since the inference speed of EM-VLM4AD-Base for most samples fails to meet the required time-consumption condition ($<$ 80 ms), RSDR on real-world data test is relatively low. The test results of RSDR indicate that our method has significant advantages in terms of speed and performance, and it holds great potential to support the application of VLMs in real-time applications.

C. Ablation Study

To further explore our distillation framework, we conduct detailed ablation experiments. The specific experimental results are listed in Table II. The first row (#1) presents the results of the existing teacher model, which is trained on regular cases. To obtain the student model, we reduce the number of model parameters by cutting off half of the blocks of the teacher model. The second row (#2) shows the training results of the student model on the merged dataset. By comparing it with the teacher model, we observe that reducing the model parameters leads to a significant decline in the student model's performance on regular cases. When we apply the commonly used distillation strategy (soft label distillation) in experiment (#3), there is an improvement in performance on regular cases, though it still lags behind the teacher model. Building on experiment (#3), we incorporate the MoE structure into the student network. This results (#4) in a notable boost in the model's performance on regular cases, surpassing the teacher model by 0.47%. However, this strategy does not yield a significant improvement on corner cases. The possible cause of this is the imbalance in the ratio of regular cases to corner cases in the training set (17:1). To address this, we further investigate whether the H2G strategy can help alleviate this problem. As is shown in (#5), H2G has indeed effectively improved the performance in corner cases. However, the performance of regular cases shows some fluctuations. How to strike a good balance between the teacher's instruction and the student's exploration of difficult examples is a problem that requires further exploration. To this end, we further validate the Progressive Release Distillation Strategy we proposed. As can be seen from the experimental results (#6), this strategy can help the model better balance the performance of regular cases and corner cases.

D. Visualization of MoE

We have visualized the weights of MoE-adapter. As shown in the Figure 3, the activation status of MoE is presented when the model encounters regular cases and corner cases. It can be observed that the activation of experts aligns with expectations. When encountering corner cases, more experts are activated. The cooperation between the MoE module and the hardness-explored strategy H2G can adaptively activate the corresponding experts.

Fig. 3. The activation status of MoE is presented when the model encounters regular cases and corner cases. When encountering regular cases, the activation pattern of experts is relatively single. When encountering corner cases, more experts are activated.

V. CONCLUSION

In conclusion, we propose Curriculum Learning-based Knowledge Distillation (CLKD) framework to address the dilemma of the elusive equilibrium between performance and efficiency. It combines MoE for student model expressiveness, uses Hardness-explored at Two Granularities (H2G) to differentiate sample difficulty, and the Progressive Release Distillation strategy to gradually decrease teacher model influence. This boosts the student model's generalization in complex driving scenarios. Extensive experiments have demonstrated the strength of CLKD in both performance and efficiency.

ACKNOWLEDGMENT

We sincerely thank the anonymous reviewers for their valuable comments and constructive suggestions. We also acknowledge the computing resources provided by our institution.

REFERENCES

[1] Yan Zeng, Hanbo Zhang, Jiani Zheng, Jiangnan Xia, Guoqiang Wei, Yang Wei, Yuchen Zhang, Tao Kong, and Ruihua Song. What matters in training a gpt4-style language model with multimodal inputs? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7930–7957, 2024.

[2] 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.

[3] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[6] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

[7] 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. *arXiv preprint arXiv:2402.12289*, 2024.

[8] Akshay Gopalkrishnan, Ross Greer, and Mohan Trivedi. Multi-frame, lightweight & efficient vision-language models for question answering in autonomous driving. *arXiv preprint arXiv:2403.19838*, 2024.

[9] Enming Zhang, Xingyuan Dai, Yisheng Lv, and Qianghai Miao. Minidrive: More efficient vision-language models with multi-level 2d features as text tokens for autonomous driving. *arXiv preprint arXiv:2409.07267*, 2024.

[10] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

[11] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

[12] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[13] Daniel Bogdoll, Stefani Gunesh Ka, Florian Rößner, Werner Ritter, and Jürgen Beyerer. One ontology to rule them all: Corner case scenarios for autonomous driving. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1501–1508, 2023.

[14] Tianqi Liu, Yanjun Qin, Shanghang Zhang, and Xiaoming Tao. Empowering corner case detection in autonomous vehicles with multimodal large language models. *IEEE Signal Processing Letters*, 2024.

[15] Jingxing Zhou and Jürgen Beyerer. Corner cases in data-driven automated driving: Definitions, properties and solutions. *IEEE Xplore*, 2023.

[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[17] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[18] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[19] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging Large Vision-Language Models and End-to-End Autonomous Driving. *arXiv e-prints*, page arXiv:2410.22313, October 2024.

[20] Kai Chen, Yanze Li, Wenhua Zhang, Yanxin Liu, Pengxiang Li, Ruiyuan Gao, Lanqing Hong, Meng Tian, Xinhai Zhao, Zhenguo Li, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. Automated Evaluation of Large Vision-Language Models on Self-driving Corner Cases. *arXiv e-prints*, page arXiv:2404.10595, April 2024.

[21] 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. *arXiv e-prints*, page arXiv:2402.12289, February 2024.

[22] Haoxiang Gao and Yu Zhao. Application of vision-language model to pedestrians behavior and scene understanding in autonomous driving, 2025.

[23] Qianhan Feng, Wenshuo Li, Tong Lin, and Xinghao Chen. Align-kd: Distilling cross-modal alignment knowledge for mobile vision-language model, 2024.

[24] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[25] Fangxun Shu, Yue Liao, Le Zhuo, Chenning Xu, Lei Zhang, Guanghao Zhang, Haonan Shi, Long Chen, Tao Zhong, Wanggui He, et al. Llava-mod: Making llava tiny via moe knowledge distillation. *arXiv preprint arXiv:2408.15881*, 2024.

[26] Kai Chen, Yanze Li, Wenhua Zhang, Yanxin Liu, Pengxiang Li, Ruiyuan Gao, Lanqing Hong, Meng Tian, Xinhai Zhao, Zhenguo Li, et al. Automated evaluation of large vision-language models on self-driving corner cases. *arXiv preprint arXiv:2404.10595*, 2024.

[27] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[28] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023.

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

[30] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[31] 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*, pages 65–72, 2005.

[32] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[33] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.