DriveAdapter: Breaking the Coupling Barrier of Perception and Planning in End-to-End Autonomous Driving

Xiaosong Jia[1,2], Yulu Gao[2,3], Li Chen[2], Junchi Yan[1,2†], Patrick Langechuan Liu[4], Hongyang Li[2,1†]

[1] MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
[2] OpenDriveLab, Shanghai AI Lab   [3] Beihang University   [4] Anker Innovations
[†]Correspondence authors
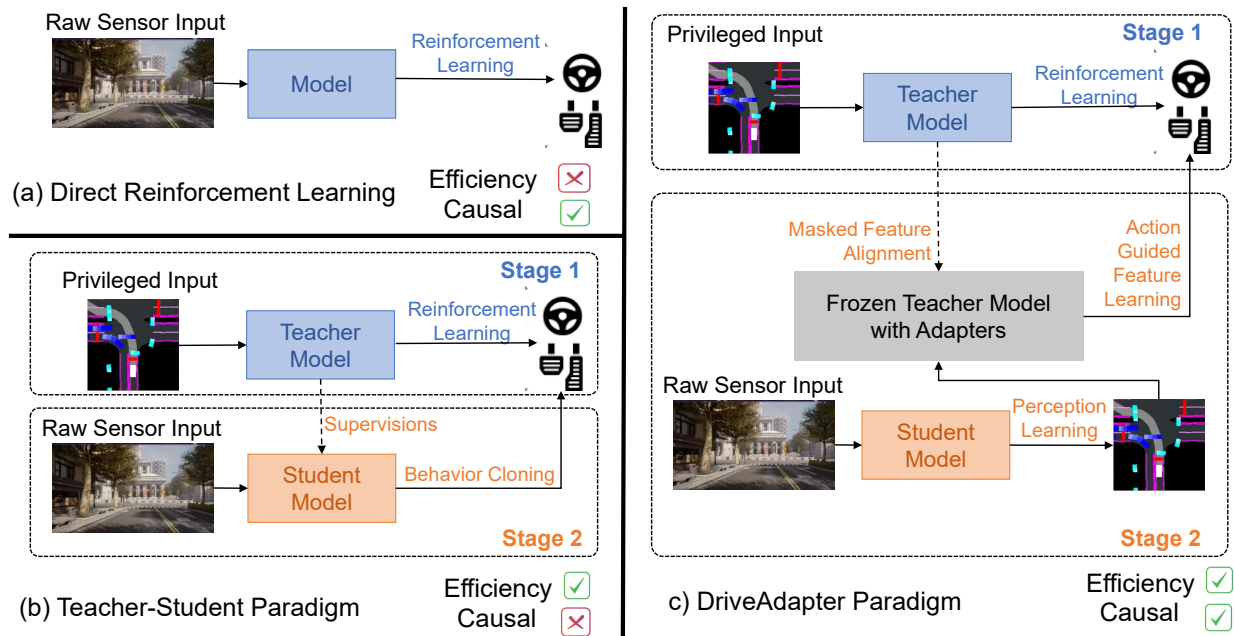https://github.com/OpenDriveLab/DriveAdapter

Figure 1: **Comparison of different paradigms for end-to-end autonomous driving.** (a) Though directly conducting reinforcement learning (RL) with raw sensor inputs [35] enables the inference model to learn the causal relationship of driving through rewards, it requires tens of days of training with the simulator rendering raw sensor inputs, which is of low efficiency. (b) State-of-the-art works [4, 3, 40, 32, 14] usually adopt the teacher-student paradigm to enable efficient policy learning by providing privileged inputs (perception ground-truth) to the RL model. However, their student model suffers from the causal confusion issues [37] due to behavior cloning. (c) In the proposed DriveAdapter paradigm, the model still enjoys high RL training efficiency while the usage of the frozen teacher model empowers the inference process with its driving knowledge. The student model could focus on perception learning, and the proposed adapter module with its masked feature alignment objective functions deals with the distribution gap between the perception results and the privileged inputs.

Abstract

End-to-end autonomous driving aims to build a fully differentiable system that takes raw sensor data as inputs and directly outputs the planned trajectory or control signals of the ego vehicle. State-of-the-art methods usually follow the 'Teacher-Student' paradigm. The Teacher model uses privileged information (ground-truth states of surrounding agents and map elements) to learn the driving strategy. The student model only has access to raw sensor data and conducts behavior cloning on the data collected by the teacher model. By eliminating the noise of the perception part dur-

ing planning learning, state-of-the-art works could achieve better performance with significantly less data compared to those coupled ones.

However, under the current Teacher-Student paradigm, the student model still needs to learn a planning head from scratch, which could be challenging due to the redundant and noisy nature of raw sensor inputs and the casual confusion issue of behavior cloning. In this work, we aim to explore the possibility of directly adopting the strong teacher model to conduct planning while letting the student model focus more on the perception part. We find that even equipped with a SOTA perception model, directly letting the student model learn the required inputs of the teacher model leads to poor driving performance, which comes from the large distribution gap between predicted privileged inputs and the ground-truth.

To this end, we propose DriveAdapter*, which employs adapters with the feature alignment objective function between the student (perception) and teacher (planning) modules. Additionally, since the pure learning-based teacher model itself is imperfect and occasionally breaks safety rules, we propose a method of action-guided feature learning with a mask for those imperfect teacher features to further inject the priors of hand-crafted rules into the learning process. DriveAdapter achieves SOTA performance on multiple closed-loop simulation-based benchmarks of CARLA.*

1. Introduction

In recent years, autonomous driving has become an active research topic due to the enormous progress of deep learning. A traditional pipeline of an autonomous driving system is usually composed of object detection [23], motion prediction [18, 17], trajectory planning [32], *etc*. To fully unleash the power of deep learning and big data and avoid cumulative errors, the concept of end-to-end autonomous driving is proposed [29, 26, 15, 5] which aims to build a fully differentiable model directly mapping the raw sensor data into planned trajectories or control signals.

One difficulty of end-to-end autonomous driving is that the noisy and redundant raw sensor inputs make it hard to directly learn a good policy. For example, raw sensor inputs based reinforcement learning (RL) agent MaRLn [35] requires 20 million steps (around 20 days) to converge even equipped with their pretraining techniques. To this end, in Roach [45], they decouple the learning process into two steps: (i) conduct the RL algorithm based on privileged inputs - rasterizing the ground-truth location of surrounding agents and traffic signs into 2D bird's-eye-view (BEV) tensors. The trained RL model is called the *teacher model* as it uses privileged inputs and thus performs well. (ii) conduct behavior cloning with raw sensor inputs on the data col-

lected by the teacher model. This model only with access to raw sensor data is called the *student model* since they are supervised by the teacher model. By decoupling the perception noise from the driving strategy learning process, Roach could achieve much better performance on more challenging benchmarks in 10 million steps. Besides the benefits of efficient RL training, in LBC [4] and PlanT [32], they demonstrate that training a teacher model from rule-based expert and then using the teacher model to provide extra supervision for the student model could bring significant performance gains as well. Due to the aforementioned advantages of the decoupled planning and perception learning, the teacher-student paradigm has been widely adopted by the state-of-the-art (SOTA) works [4, 3, 39, 32, 14].

However, there are still issues under the existing paradigm. The student model still needs to train a planning head from scratch by behavior cloning, which could result in the causal confusion issue [37]. Specifically, the causal confusion issue here refers to the phenomenon that the student model learns the visual clue of the results instead of the cause of the desired actions. For example, the well-known inertia problem [37] is that the agent sometimes keeps still forever at the intersection. It is because, during behavior cloning, the student model might learn the improper causal correlation that the ego vehicle should copy behaviors of its surrounding vehicles at the intersection. In fact, the behaviors are determined by the traffic light. However, since the traffic light is smaller in images compared to vehicles, the student tends to find the shortcut [37]. As a result, during evaluation, if there are no vehicles nearby or all vehicles are behind the ego vehicle, it might get stuck. The causal confusion issues could be solved by techniques such as reweighting the distribution of training data [33, 30, 37] or a causal prior structure/network [38, 9].

Inspired by the fact that we already have a strong teacher model trained by RL without any causal confusion issue, in this work, we aim to explore the way to *utilize the teacher model to conduct planning directly instead of training a planning head for the student model from scratch.* In this way, the learning process of perception and planning is completely decoupled and thus the disadvantage of behavior cloning could be avoided, as demonstrated in Fig. 1. As a result, we could directly benefit from the driving knowledge inside the teacher model learned by RL. One intuitive implementation of this idea is to train a student model to generate the required privileged inputs for the frozen teacher model, *e.g.*, a BEV segmentation student model for the Roach teacher model. However, we find that even equipped with the SOTA perception model: BEVFusion [24] + Mask2former [6], its final driving performance is still unsatisfying. The issue comes from the large distribution gap between the predicted BEV segmentation and ground-truth. It could be formulated as a domain transfer

problem since the teacher model has only seen the ground-truth BEV segmentation during the training process.

Inspired by the usage of adapters in the natural language processing (NLP) [13] and computer vision [12] field to adopt huge foundation models for downstream tasks, we propose **DriveAdapter**, which connects the output of the student model (perception) and the input of the teacher model (planning). Specifically, we add a learnable adapter module after each part of the teacher model and apply feature alignment objective functions on each adapter. In this way, the adapter could learn to transfer the imperfect feature from the student model's domain to the teacher model's domain in a layer-by-layer supervised way.

Additionally, we observe that the pure learning-based teacher model itself is imperfect and it is a common practice to add extra hand-crafted rules during the final decision process [40, 43, 14]. Thus, even if the student with adapters could losslessly generate the required inputs for the teacher model, it is still upper-bounded by the imperfect performance of the teacher. To this end, we propose to back-propagate an action loss to all adapters and mask all feature alignment loss if the teacher model is overridden by the rule. In this way, we force adapters to directly learn the feature required to generate good actions instead of just mimicking the teacher. By combining the two proposed techniques, DriveAdapter achieves state-of-the-art performance on two closed-loop evaluation benchmarks of the CARLA simulator. Moreover, we conduct thorough ablation studies and give results of other related attempts such as directly generating intermediate features of the teacher.

In summary, this work has the following contributions:

- To the best of our knowledge, we are the first to thoroughly explore the paradigm of directly utilizing the teacher head to conduct planning for the end-to-end autonomous driving task. Under such decoupled paradigm, the disadvantages of behavior cloning such as causal confusion could be avoided.

- The intermediate output form of BEV segmentation between the perception and planning modules has strong interpretability, shedding insights into the typically black-box pipeline of end-to-end autonomous driving. The decoupled perception model could enjoy the recent rapid progress of BEV perception and semantic segmentation.

- To deal with the imperfect perception issue as well as the imperfect teacher model issue, we propose DriveAdapter along with a masked feature distillation strategy. By combining the two techniques, it could achieve state-of-the-art performance on two public benchmarks.

- We give thorough ablation studies and other related attempts to provide more insights and understanding regarding the new decoupled paradigm.

We believe that the rich driving knowledge within an RL expert model learned by millions of steps of exploration should be utilized more extensively instead of only for behavior cloning. We hope the proposed decoupled paradigm, our failing attempts, and our working techniques could all provide useful insights for this line of study.

2. Related Works

2.1. End-to-End Autonomous Driving

The concept of end-to-end autonomous driving could date back to 1980s [29]. In the era of deep learning, early works conduct behavior cloning from a rule-based expert. CIL [10] adopts a simple CNN to directly map the front-view image from a camera to control signals. Further, in the extending work CILRS [11], they add an auxiliary task to predict the ego vehicles' current speed to alleviate the inertia issue. Later, in MaRLn [35], they explore the way to apply reinforcement learning to obtain a driving policy to surpass the rule-based expert. However, their method suffers from the high-dimensional raw sensor inputs for urban driving. In LBC [4], they propose to first train a teacher model with privileged inputs and then utilize this teacher model to provide supervision signals for all high-level commands, which significantly boosts the performance of the student model and thus the teacher-student paradigm has dominated the field. Roach [45] trains the teacher model with reinforcement learning, which demonstrates strong robustness and diversity compared to imitation learning-based teacher models and has been adopted by multiple most recent SOTA works [40, 14]. In PlanT [32], they propose to adopt Transformer for the teacher model on the states of the environment instead of CNN on the rasterized images, which demonstrates good scalability and interpretability. As for the student models, NEAT [7] transfers representations to BEV space. Transfuser [31, 8] adopts Transformer for camera and LiDAR fusion. LAV [3] adopts PointPainting [36] and proposes to predict all agents' future trajectories to augment the dataset. TCP [40] combines the trajectory prediction [20, 19] with the control signal prediction. Interfuser [34] injects safety-enhanced rules during the decision-making of the student models. MMFN [43] adopts VectorNet for map encoding and MILE [14] proposes to learn a world model for the student model. Among concurrent works, ThinkTwice [21] proposes a DETR-like scalable decoder paradigm for the student model. CaT [42] designs a knowledge distillation framework for the Teacher-Student paradigm. ReasonNet proposes specific modules for student models to better exploit temporal and global information. In [16], they propose to formulate the output of the student as classification problems to avoid averaging.

We could observe that state-of-the-art works all use behavior cloning for student models. In this work, we explore the way to further decouple perception and planning learn-

ing by directly adopting the frozen teacher model for planning, while keeping the system end-to-end differentiable.

2.2. Adapter for Deep Learning Models

In recent days, huge foundation models [1] pretrained on an enormous amount of data have demonstrated strong transfer ability on downstream tasks. Since finetuning the whole model is computationally expensive and tends to overfit on downstream tasks with limited data, Adapter [13] is first proposed in the NLP field where they fix the parameters of the original model and add extra learnable parameters between blocks of the original model so that it could keep the generalization ability of the original model while having task-specific changes. Later, the Adapter is also shown to be effective in the computer vision field [12, 44].

In this work, we adopt the idea of Adapter to keep the knowledge in the teacher model while filling the gap between the predicted privileged inputs and the ground-truth.

3. Method

3.1. Student Model for Perception Learning

Suppose we have a teacher model with privileged information as inputs. One popular form of encoding the privilege information is the 2D bird's-eye-view (BEV) tensors [4, 45] composed of the rasterized position of surrounding agents, lanes, and traffic signs, where the value of each channel could be either 0 or 1 to represent the existence of the corresponding type of object at certain locations (In PlanT [32], they propose to encode the scene into discrete tokes so that Transformer could be adopted). Here, we choose Roach to match with SOTA works [40, 14].

The student model takes raw sensor data as inputs (in this paper, images from four cameras and the point cloud from one LiDAR) and generates the desired input for the teacher model. For Roach, it could be formulated as a semantic segmentation task under BEV space [27]. Here, we adopt the BEVFusion [24] to convert raw sensor data into BEV features. Specifically, we adopt LSS [28] to scatter the image features from cameras into their corresponding BEV grid based on their location and depth [22]. For LiDAR, we adopt the commonly used SECOND [41] backbone to convert the point cloud into a BEV feature map. By concatenating the BEV feature maps of cameras and LiDAR, we obtain the 2D BEV representation of the scene. To conduct the semantic segmentation task, we adopt the state-of-the-art Mask2former [6] head.

Nevertheless, even equipped with the most advanced perception module as of today, we find that directly feeding the predicted BEV segmentation (mIoU 0.35 on test unseen scenes) to the teacher model does no work, *i.e.*, much worse driving performance compared to SOTA works as shown in Table 1.

Method	Input	Driving Score ↑
Transfuser [31, 8]	Camera + LiDAR	31.0
LAV [3]	Camera + LiDAR	46.5
Student Model + Frozen Roach	Camera + LiDAR	8.9
Roach [45]	Privileged Info.	74.2
Roach + Rule [40]	Privileged Info.	**87.0**

Table 1: **Performance comparison among different teacher and student models on Town05 Long.** Refer to Sec. 4 for details of methods and metrics.
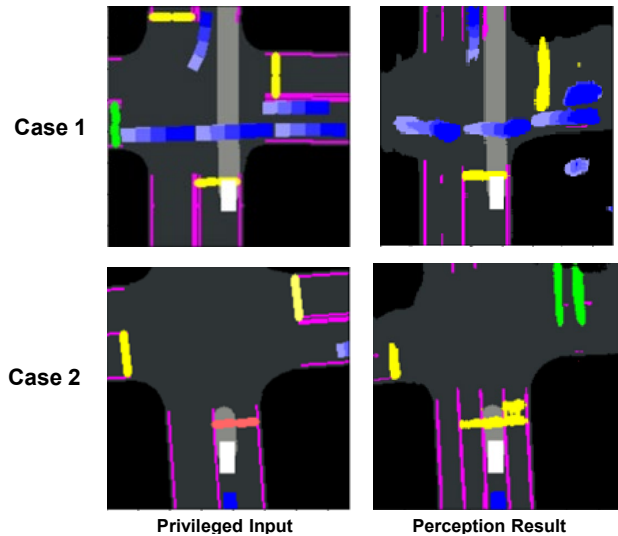

Figure 2: **Visualization of BEV segmentation results.** The distribution gap between the ground truth and perception results is obvious. Note that the routes and the ego vehicle are directly painted instead of being predicted.

There are two factors that impede the performance of the student model + frozen Roach:

- *Imperfect perception results.* Though we have carefully tuned the BEVFusion + Mask2former on the CARLA scene, the predicted BEV segmentation still has a relatively large gap compared to the ground-truth. As illustrated in Fig. 2, the predicted surrounding agents' location and lanes are incomplete and blurry. Worse still, the states of traffic lights could also be wrong. As a result, the teacher model could not handle such inputs since it has only seen the ground-truth during its training process.

- *Imperfect teacher model.* It is a common practice in the community [43, 40, 14] to apply hand-crafted rules on top of the teacher models' outputs for situations such as emergency braking as the learning-based teacher model would make mistakes as well. As shown in Table. 1, after adding rules to the Roach, we observe its performance
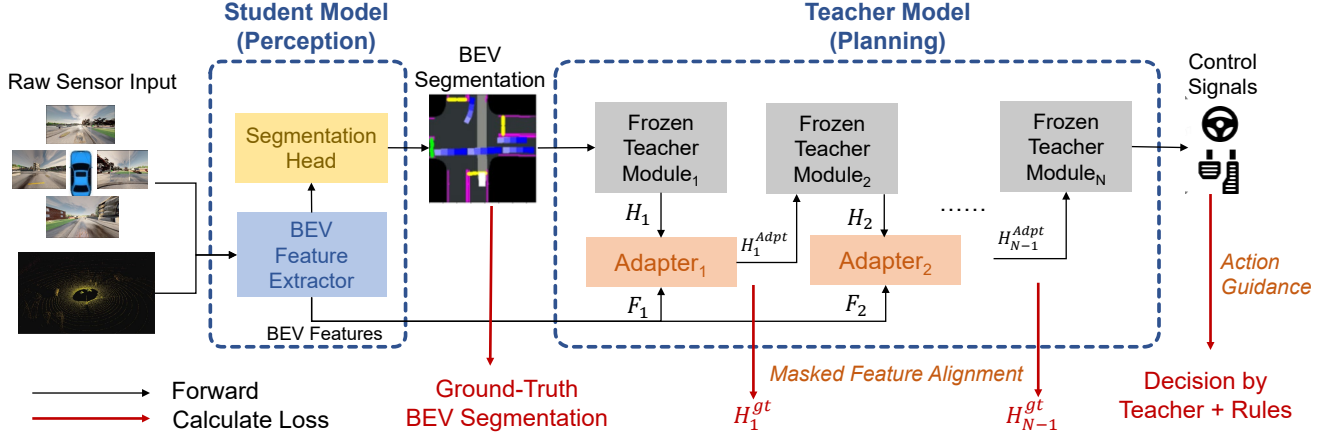
Figure 3: **Overall architecture of DriveAdapter.** (a) The student model takes raw sensor data as inputs and extracts BEV features for the usage of BEV segmentation and adapter module. (b) The predicted BEV segmentation is fed into the frozen teacher model and the plug-in adapter module. (c) The adapter module receives supervision from the feature alignment objective with the ground-truth teacher feature. For cases the teacher model is taken over by rules, a mask is applied on the alignment loss and the supervision of all adapter modules is from the backpropagation of action loss.

is boosted by a large margin. Therefore, even if the student model could generate perfect BEV segmentation, its performance is still upper bounded.

3.2. Adapter Module

Since the teacher model has only been trained by the BEV segmentation ground-truth, it is sensitive to the noise in the predicted BEV segmentation because of the large distribution gap. Inspired by the usage of the Adapter [13, 12, 44] on foundation models [1] to adopt them into downstream tasks with lower cost and less overfitting, we propose to add Adapters between the student and teacher model. The overall architecture of DriveAdapter is shown in Fig. 3.

Formally, denote the predicted BEV segmentation as H_0 where the subscript 0 means that they could be the initial input of the teacher model. Suppose the teacher model has N modules in a sequential order[1], where Teacher_i denotes its ith module and H_{i-1} and H_i denote the original input and output of Teacher_i respectively: $H_i = \text{Teacher}_i(H_{i-1})$. Besides, since we want the adapter to have access to the raw sensor inputs so that the model could enjoy the benefits of end-to-end learning, we denote the raw BEV feature from the student model as F and we use a series of convolutional layers to downsample F so that F_i could have the same resolution with H_i.

Adapter: The forward process of the frozen teacher model

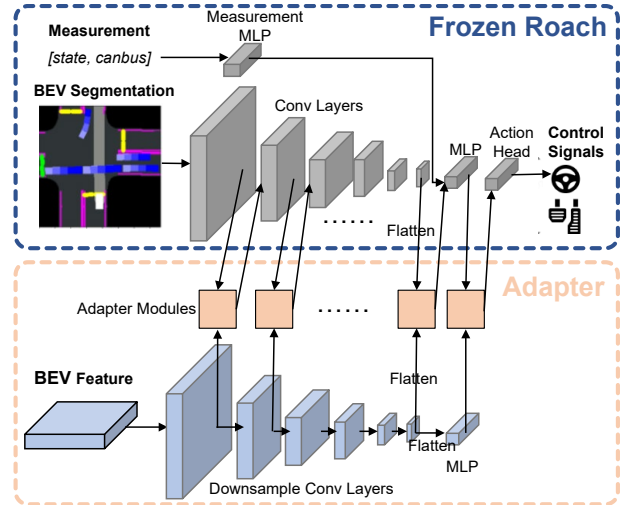[1]For more complex teacher models, specific adapter module could be designed accordingly.

Figure 4: **Details of Adapter Modules wtih Roach.**

with adapter modules is:

$$H_{i-1}^{\text{Adpt}} = \text{Adapter}_{i-1}([H_{i-1}; F_{i-1}]), \qquad (1)$$

$$H_i = \text{Teacher}_i(H_{i-1}^{\text{Adpt}}), \qquad (2)$$

where Adapter_{i-1} in Eq. 1 denotes the adapter module for i^{th} module of the teacher model which takes the feature from the previous layer H_{i-1} and the raw feature F_{i-1} as inputs and outputs the adapted feature for the next layer H_{i-1}^{Adpt} which has exactly the same tensor shape with H_{i-1}. The Adapter_{i-1} is implemented as CNN layers for 2D feature maps and as multi-layer perceptrons (MLPs) for 1D

feature maps. Specifically, with Roach as the teacher model, after each layer of Roach's network (both convolutional layers and linear layers), there is a corresponding adapter module and we apply feature alignment loss on each output. There are two exceptions: (i) the measurement encoder because it takes the state of the ego vehicle as inputs which is provided directly by the sensor and thus there is no error; (ii) the output linear layer since it generates actions instead of features. Fig. 4 gives the details of the adapter modules.

Feature Alignment: To fill the gap between the student's prediction and the ground-truth inputs for the teacher model, we apply a *feature alignment objective function* for each adapter module:

$$\mathcal{L}_i = \text{Reg}(\boldsymbol{H}_{i-1}^{\text{Adpt}}, \boldsymbol{H}_{i-1}^{\text{gt}}), \tag{3}$$

where $\boldsymbol{H}_{i-1}^{\text{gt}}$ denotes the feature from $(i-1)^{th}$ module of the teacher model (without the adapter module) when the input is the ground-truth BEV segmentation. Reg denotes the regression loss function and we simply adopt smooth L1 loss here. The intuition behind this design is that we want each adapter module to recover the ground-truth feature required by the teacher model with an additional information source - the raw BEV feature. In this way, the distribution gap between the prediction and the ground-truth feature could be reduced gradually in a layer-by-layer supervised way.

Mask & Action Guidance: As for the imperfect teacher model issue, we inject the priors of the hand-crafted rules into the training process in two ways: (i) *Mask for Feature Alignment*: For the cases where the teacher model is wrong and is taken over by the rules, masks are applied on all feature alignment losses since the original features in the teacher model lead to wrong decisions and thus we did not want the adapter module to recover them. (ii) *Action Guided Feature Learning*: In order to let the adapter module transform the features of the original teacher model into the features leading to the final decision by rules, we calculate the loss between the model prediction and the actual decision and make it backpropagate all the way through the frozen teacher model and adapter modules. In this way, for those cases, the adapter module is able to turn the final output of the model into the right decision. Experiment results show that the mask and the action guidance could explicitly improve driving performance.

4. Experiments

4.1. Dataset and Benchmark

We use the widely adopted CARLA simulator (version 0.9.10.1) for data collection and closed-loop driving performance evaluation.

As for the data collection, we collect data with the teacher model *Roach* [45] + *rules* to match with SOTA

works [40, 14]. Following the common protocol, we collect data at 2Hz on Town01, Town03, Town04, and Town06 for training and the total number of frames is 189K which is similar to [31, 8, 3, 40]. At each frame, we collect raw sensor data from 4 cameras and 1 LiDAR and we store the labels including depth, segmentation, control signals, ground-truth features, and the states of the ego vehicle. Additionally, recent works have collected more data and even on more towns for training: [34] (3 million frames, 8 towns) and [14] (2.9 million frames but at 25 Hz), which might trigger the unfair comparison issue as discussed in the community[2,3]. To this end, we collect a dataset with 2 million frames on 8 towns and we denote all models trained with extra data with *.

As for the evaluation process, we conduct closed-loop running under two public benchmarks: Town05Long (most widely used) and Longest6 (36 challenging routes selected by [8]). Both benchmarks are composed of tens of routes while each route contains a series of target points to indicate the destination of driving. In addition, there are manually defined challenging events randomly happening during driving to evaluate the driving agent's ability to deal with long-tail problems. For example, jaywalking pedestrians might suddenly appear. At the intersection, an opposite vehicle might illegally run a red traffic light. Thus, the model should have comprehensive knowledge about driving instead of simple lane following. Note that during the evaluation process, the model only has access to the raw sensor data and the use of privileged information is prohibited. For implementation details of the simulation and the model, please refer to supplemental materials.

4.2. Metrics

Official metrics of CARLA are used. **Infraction Score (IS)** measures the number of infractions made along the route, with pedestrians, vehicles, road layouts, red lights, *etc*. **Route Completion (RC)** is the percentage of the route completed by the autonomous agent. **Driving Score (DS)** is the **main metric** which is the product of Route Completion and Infraction Score.

4.3. Comparison with State-of-the-Art Works

We compare with SOTA works on two widely used public benchmarks: Town05 Long and Longest6 as shown in Table 2 and Table 3 respectively. We could observe that DriveAdapter performs the best under the limited data setting and after adopting the dual outputs trick in [40] to improve safety, it even performs on par with competitors trained on 10× data. After feeding more data to the model, DriveAdapter sets new records on both benchmarks. After investigation, we find that the major gain of 10× data comes

---

[2]https://github.com/opendilab/InterFuser/issues/3
[3]https://github.com/wayveai/mile/issues/4

Method	Teacher	Student	Reference	DS↑	RC↑	IS↑
CILRS [11]	Rule-Based	Behavior Cloning	CVPR 19	7.8	10.3	0.75
LBC [4]	Imitation Learning	Behavior Cloning + DAgger	CoRL 20	12.3	31.9	0.66
Transfuser [31, 8]	Rule-based	Behavior Cloning	TPAMI 22	31.0	47.5	**0.77**
Roach [45]	Reinforcement Learning	Behavior Cloning + DAgger	ICCV 21	41.6	96.4	0.43
LAV [3]	Imitation Learning	Behavior Cloning	CVPR 22	46.5	69.8	0.73
TCP [40]	Reinforcement Learning	Behavior Cloning	NeurIPS 22	57.2	80.4	0.73
ThinkTwice [21]	Reinforcement Learning	Behavior Cloning	CVPR 23	65.0	95.5	0.69
DriveAdapter	Reinforcement Learning	Frozen Teacher + Adapter	Ours	61.7	92.3	0.69
DriveAdapter + TCP	Reinforcement Learning	Frozen Teacher + Adapter	Ours	**65.9**	94.4	0.72
MILE*† [14]	Reinforcement Learning	Model-Based Imitation Learning	NeurIPS 22	61.1	**97.4**	0.63
Interfuser* [34]	Rule-Based	Behavior Cloning + Rule	CoRL 22	68.3	95.0	-
ThinkTwice* [21]	Reinforcement Learning	Behavior Cloning	CVPR 23	70.9	95.5	0.75
DriveAdapter + TCP*	Reinforcement Learning	Frozen Teacher + Adapter	Ours	**71.9**	97.3	0.74

Table 2: **Performance on Town05 Long benchmark.** ↑ means the higher the better. * denotes using extra data. † denotes no scenarios are used, which is a much easier benchmark. *+TCP* means we adopt its dual output technique by adding an additional trajectory prediction head [40].

Method	DS↑	RC↑	IS↑
WOR [2]	23.6	52.3	0.59
LAV [3]	34.2	73.5	0.53
Transfuser [31, 8]	56.7	**92.3**	0.62
PlanT with Perception [32]	57.7	88.2	0.65
ThinkTwice [21]	61.3	73.0	0.81
ThinkTwice* [21]	66.7	77.2	0.84
DriveAdapter	59.4	82.0	0.68
DriveAdapter + TCP	62.0	82.3	0.70
DriveAdapter + TCP*	**71.4**	88.2	**0.85**

Table 3: **Performance on Longest6 benchmark.**

Method	DS↑	RC↑	IS↑
DriveAdapter	61.7	92.3	0.69
w/o Feature Alignment Loss	45.4	69.1	0.66
w/o Mask for Feature Alignment	56.9	85.4	0.65
w/o Action Loss	47.1	90.5	0.52

Table 4: **Ablation on loss terms of the adapter.**

from better detection of the red light - a *perception* issue. This demonstrates the benefit of having an explainable intermediate representation of BEV segmentation. There are also some common issues happening under both limited and 10× data settings. We give more thorough investigations about failure cases in Sec. 5.

4.4. Ablation Study

In this section, we conduct ablation studies to verify the effectiveness of each design of DriveAdapter. All experiments are conducted on Town05 Long benchmark with 189K training data.

4.4.1 Loss Design

In DriveAdapter, we have two kinds of loss terms for the adapter modules: the feature alignment loss to deal with the distribution gap between the prediction and ground-truth BEV segmentation, and the action loss to handle the cases

when the learning-based teacher model makes mistakes and the decision is overridden by rules.

The ablation study regarding the two loss terms is in Table 4. We can observe that:

- If we do not conduct feature alignment, the supervision for the adapter module is only the action loss which is very similar to behavior cloning. Thus, we could observe a drop in route completion due to the inertia issue.

- If we do not apply the masking strategy for those cases where the learning-based teacher model made mistakes, the supervision signal of action loss and feature alignment loss is conflict. As a result, the overall performance drops.

- If we discard the action loss, we could observe a drastic drop in the IS (infraction score) which comes from more collisions due to the relatively aggressive teacher model.

In summary, we can find out that both loss terms, as well as the masking strategy, are significant for the learning process. The feature alignment loss allows the adapter to exploit the driving knowledge within the teacher model while the action loss and mask strategy inject information about hand-crafted rules which leads to a more conservative and thus safer driving strategy.

Method	DS↑	RC↑	IS↑
DriveAdapter	61.7	92.3	0.69
Adapter at Early Stage	47.2	93.9	0.47
Adapter at Late Stage	54.3	79.9	0.69
w/o BEV Raw Feature	34.8	82.3	0.43
Unfrozen Teacher Model	49.0	73.2	0.72

Table 5: **Ablation on the design of the adapter.**

4.4.2 Adapter Design

In DriveAdapter, all modules of the teacher model are frozen and after each module, there is an adapter that takes both the feature from the previous layers and the raw BEV features as inputs. In Table 5, we give ablation studies regarding those designs. We can conclude that:

- If we only set adapters at the early stage of the teacher model - only on 2D feature maps, the agent became very reckless with a significant drop of IS. It might be due to the cumulative errors at the late stage.

- If we only set adapters at the late stage - only on the flattened 1D feature maps, the agent stuck more often (lower RC). We conjecture that the lack of details about the scene, *i.e.*, the information at the early stage, makes it hard to fully utilize the causal inference ability of the teacher model and the adapter might serve more as a behavior cloning module.

- If we do not feed the raw BEV feature to the adapters, the agent performs poorly. It is natural since the information source of the downstream module is only the blurry and incomplete predicted BEV segmentation now, which is not enough to recover features.

- As expected, unfreezing the teacher model would lead to worse performance. The reason is that the behavior cloning process on the dataset with limited size (compared to tens of millions of steps exploration during reinforcement learning) would cause catastrophic forgetting [25], which has also been observed in foundation models [1] when finetuning on one downstream task. In fact, the model's final performance is very similar to LAV [3], a SOTA behavior cloning-based model. This experiment demonstrates the importance of freezing the teacher model.

4.5. Beyond BEV Segmentation

In this section, we explore the possibility of directly regressing the middle feature maps of the teacher model instead of predicting the BEV segmentation. In other words, the student model does not generate inputs at layer 0 for the teacher model, for example, we let the student generate the

Target	DS↑	RC↑	IS↑
BEV Segmentation	8.9	93.2	0.09
CNN-2	16.0	88.9	0.12
CNN-4	19.0	88.2	0.23
CNN-6	36.9	94.9	0.38
Latent	39.0	100.0	0.39
Action	39.2	61.2	0.62

Table 6: **Experiments about different learning targets for the student model.** The driving performance is evaluated by feeding the prediction of the student model directly into the corresponding layer of the teacher model. *CNN-i* denotes the CNN feature map at the ith layer of Roach [45]. *Latent* denotes the 1D feature map after the linear layer on the flattened BEV feature. *Action* denotes a behavior cloning student model.

feature map at layer 1 of the teacher model and then we feed the predicted feature map into the rest of the frozen teacher model. In this spirit, we conduct experiments with feature maps at different layers of the teacher model and the results are in Table 6.

From the results, we find out that as the learning target of the student model becomes deeper, the driving performance increases. We hypothesize that feeding features directly into deeper layers of the teacher model would encounter fewer cumulative errors. The only exception is the pure behavior cloning agent. Since it only has action supervision and does not utilize the teacher model at all, it encounters severe inertia issue which leads to a low route completion (RC).

Nevertheless, as discussed in Sec. 4.4.2, features at the early stage contain more detailed information about the scene and some of them might be important for the teacher model to make decisions and the usage of the adapter could alleviate the aforementioned cumulative errors. Thus, in this work, we stick to the BEV segmentation target. Besides, compared to high-dimensional features, the semantic segmentation is human-readable, which could be helpful for debugging the perception issues (*e.g.*, the fog is so heavy that the model could not detect the traffic light far away).

5. Failure Case Analysis

Though DriveAdapter achieves state-of-the-art performance, its performance is still far from perfection (Driving Score 71). For the future development of the work, we summarize three kinds of major failure cases as shown in Fig. 5. We could conclude that:

- As for the running red light case in Fig. 5 (a), the traffic light which controls the ego vehicle's current lane is small in the front camera and the background of the traffic light is green, which makes it extremely difficult to be detected
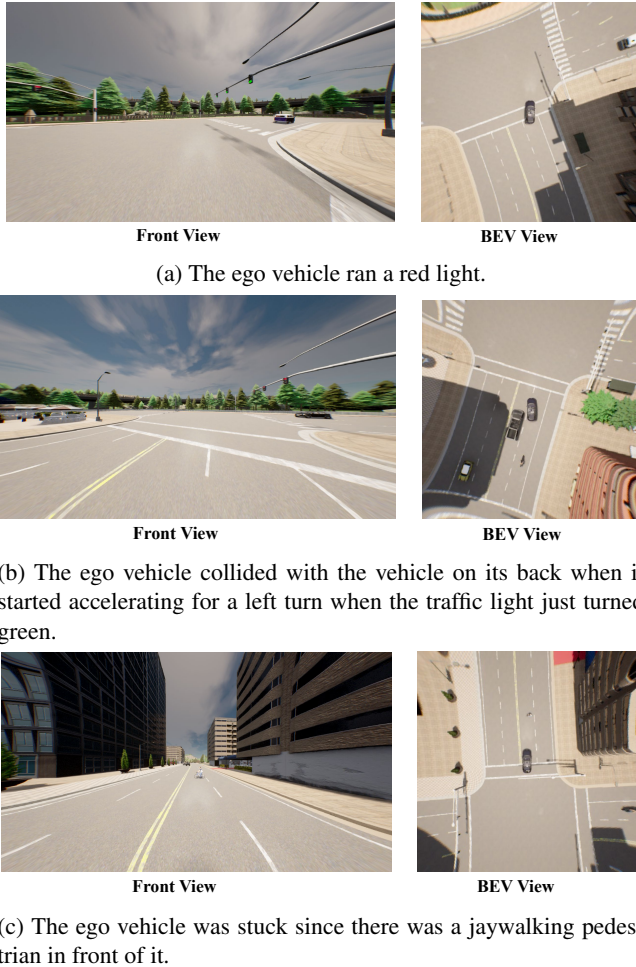
Front View — **BEV View**

(a) The ego vehicle ran a red light.

Front View — **BEV View**

(b) The ego vehicle collided with the vehicle on its back when it started accelerating for a left turn when the traffic light just turned green.

Front View — **BEV View**

(c) The ego vehicle was stuck since there was a jaywalking pedestrian in front of it.

Figure 5: **Visualization of failure cases.**

by the model. The traffic light issue has been mentioned frequently in the community [3, 14]. One possible solution is to set a specific camera and a specific neural net to detect the traffic light like in [3]. Besides, when we scale up the dataset size (from limited data setting to 10x data setting), the issue is alleviated.

- In the collision case at the intersection in Fig. 5 (b), the ego vehicle stopped too far from the stop line of the intersection. As a result, when the traffic light turned green, the ego vehicle and its neighbor started accelerating at the same, which results in the collision. Actually, it is a bad habit of the teacher model (Roach) and we find that this kind of collision happens to Roach as well. Thus, a better teacher model could solve the problem.

- As for the stuck case in Fig. 5 (c), similar to the one in (b), it is due to too much safety distance. This case is a scenario of the benchmark where the simulator would generate a jaywalking pedestrian in front of the ego vehicle. The moving logic of this pedestrian is that it would

cross the road when the distance between the ego vehicle and the pedestrian is smaller than a threshold. As a result, for a cautious ego agent, it stop before their distance under the threshold. As a result, both sides can not move and get stuck forever. One solution is to change the logic of the pedestrian to avoid stuck, which might trigger unfair comparisons with other models since it is a public benchmark. Another solution is to conduct object detection and make the ego agent creep slowly if there is no obstacle around the ego vehicle like in [8]. However, this strategy might cause more violations of rules or collisions sometimes.

In conclusion, though some failure cases are due to the inherent difficulty of perception or the buggy logic of the simulation, the decision-making process still needs improvements for perspectives such as not being too conservative. A better learning-based teacher model could further improve the performance of DriveAdapter.

6. Discussion about Real World Application of DriveAdapter

BEV segmentation provides an abstraction of the driving scene which is useful for sim2real setting. Thus, it might be feasible to train a behavior policy with a simulator (for example CARLA) or offline dataset (for example a policy learned on nuPlan or Waymo Motion Prediction data) and finetune it with a perception module and an adapter in the real world.

7. Conclusion

In this work, we propose DriveAdapter which could directly utilize the driving knowledge within a teacher model learned via reinforcement learning, in an end-to-end autonomous driving pipeline. To overcome the imperfect perception and the imperfect teacher model issue, we propose the masked feature alignment and action guidance objective function for adapters. DriveAdapter achieves state-of-the-art performance on two closed-loop autonomous driving evaluation benchmarks. We hope this could establish a new direction of research in end-to-end autonomous driving.

Limitation & Future Work: Since the performance of the teacher model (currently DS 87) is the upper bound of DriveAdapter, pushing the learning-based teacher model's performance to perfection could benefit the DriveAdapter.

Acknowledgement

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 4, 5, 8

[2] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *ICCV*, 2021. 7

[3] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8, 9

[4] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *CoRL*, pages 66–75. PMLR, 2020. 1, 2, 3, 4, 7

[5] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv*, 2306.16927, 2023. 2

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 2, 4

[7] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *ICCV*, pages 15793–15803, 2021. 3

[8] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2022. 3, 4, 6, 7, 9

[9] Chia-Chi Chuang, Donglin Yang, Chuan Wen, and Yang Gao. Resolving copycat problems in visual imitation learning via residual action prediction. In *ECCV*, pages 392–409, 2022. 2

[10] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, pages 4693–4700, 2018. 3

[11] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *CVPR*, pages 9329–9338, 2019. 3, 7

[12] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 3, 4, 5

[13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pages 2790–2799, 2019. 3, 4, 5

[14] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *NeurIPS*, 2022. 1, 2, 3, 4, 6, 7, 9

[15] 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 *CVPR*, 2023. 2

[16] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *ICCV*, 2023. 3

[17] Xiaosong Jia, Li Chen, Penghao Wu, Jia Zeng, Junchi Yan, Hongyang Li, and Yu Qiao. Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach. In *CoRL*, volume 205, pages 910–920, 2023. 2

[18] Xiaosong Jia, Liting Sun, Masayoshi Tomizuka, and Wei Zhan. Ide-net: interactive driving event and pattern extraction from human data. *IEEE RA-L*, 6(2):3065–3072, 2021. 2

[19] Xiaosong Jia, Liting Sun, Hang Zhao, Masayoshi Tomizuka, and Wei Zhan. Multi-agent trajectory prediction by combining egocentric and allocentric views. In *CoRL*, volume 164, pages 1434–1443, 2022. 3

[20] Xiaosong Jia, Penghao Wu, Li Chen, Yu Liu, Hongyang Li, and Junchi Yan. Hdgt: heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *TPAMI*, pages 1–15, 2023. 3

[21] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 3, 7

[22] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: acquisition of reliable depth for multi-view 3d object detection. *AAAI*, 2023. 4

[23] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, pages 1–18. Springer, 2022. 2

[24] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 2, 4

[25] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist metworks: the sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. 8

[26] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann Cun. Off-road obstacle avoidance through end-to-end learning. *NeurIPS*, 18, 2005. 2

[27] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE RA-L*, 5(3):4867–4873, 2020. 4

[28] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210, 2020. 4

[29] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *NeurIPS*, 1, 1988. 2, 3

[30] Aditya Prakash, Aseem Behl, Eshed Ohn-Bar, Kashyap Chitta, and Andreas Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *CVPR*, pages 11763–11773, 2020. 2

[31] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multimodal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 3, 4, 6, 7

[32] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: explainable planning transformers via object-level representations. In *CoRL*, 2022. 1, 2, 3, 4, 7

[33] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011. 2

[34] Hao Shao, Letian Wang, RuoBing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. *CoRL*, 2022. 3, 6, 7

[35] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *CVPR*, pages 7153–7162, 2020. 1, 2, 3

[36] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: sequential fusion for 3d object detection. In *CVPR*, pages 4604–4612, 2020. 3

[37] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *NeurIPS*, 33:2564–2575, 2020. 1, 2

[38] Chuan Wen, Jianing Qian, Jierui Lin, Jiaye Teng, Dinesh Jayaraman, and Yang Gao. Fighting fire with fire: avoiding dnn shortcuts through priming. In *ICML*, pages 23723–23750, 2022. 2

[39] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. In *ICLR*, 2023. 2

[40] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: a simple yet strong baseline. *NeurIPS*, 2022. 1, 3, 4, 6, 7

[41] Yan Yan, Yuxing Mao, and Bo Li. Second: sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 4

[42] Jimuyang Zhang, Zanming Huang, and Eshed Ohn-Bar. Coaching a teachable student. In *CVPR*, 2023. 3

[43] Qingwen Zhang, Mingkai Tang, Ruoyu Geng, Feiyi Chen, Ren Xin, and Lujia Wang. Mmfn: multi-modal-fusion-net for end-to-end driving. *IROS*, 2022. 3, 4

[44] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 4, 5

[45] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *ICCV*, 2021. 2, 3, 4, 6, 7, 8