Small-shot Multi-modal Distillation for Vision-based Autonomous Steering

Yu Shen[1], Luyu Yang[1], Xijun Wang[1], and Ming C. Lin[1]
https://gamma.umd.edu/researchdirections/autonomousdriving/amdsnet/

***Abstract*— In this paper, we propose a novel learning framework for autonomous systems that uses a small amount of "auxiliary information" that complements the learning of the main modality, called "small-shot auxiliary modality distillation network (AMD-S-Net)". The AMD-S-Net contains a two-stream framework design that can fully extract information from different types of data (i.e., paired/unpaired multi-modality data) to distill knowledge more effectively. We also propose a novel training paradigm based on the "reset operation" that enables the teacher to explore the local loss landscape near the student domain iteratively, providing local landscape information and potential directions to discover better solutions by the student, thus achieving higher learning performance. Our experiments show that AMD-S-Net and our training paradigm outperform other SOTA methods by up to *12.7%* and *18.1%* improvement in autonomous steering, respectively.**

I. Introduction

The core component of self-navigation systems is *autonomous steering* that requires both correct scene understanding and rapid adaptation to the changing circumstances. Because of the variant scenarios in autonomous driving, people explore the possibility of seeking auxiliary information instead of single sensor information to improve the learning of the autonomous steering task. Previous works [1], [2], [3] have tried to exploit the depth information in addition to the RGB channels, such as Lidar. The unified learning framework that involves multiple modalities of data as input is referred as multi-modality learning. However, it is computationally very expensive. Also, the framework that requires the auxiliary sensor/data for input at test time largely restricts its application to cars with less advanced equipment. Another problem is, the amount of auxiliary information may be small in some cases, e.g., expensive expert-labeled data, or sensing data from a low-frequency sensor, which makes the network harder to learn. Therefore, our aim in this work is to design a learning framework that *utilizes a small amount of auxiliary sensor/data to assist the task during training, but does not require it during test/inference time*.

In this paper, we introduce a novel learning framework for autonomous steering that uses a small amount of "auxiliary modality" data to complement the learning of the main modality, i.e., distilling knowledge from a multi-modality teacher to a single-modality student with *partially available auxiliary modality*. Specifically, we propose a small-shot auxiliary modality distillation network, **AMD-S-Net** (Sec. III-B), for the partially available setting, which is trained with our multi-modality training paradigm and meets a special "supermodel condition". It uses a special "reset operation" that allows a teacher to be aware of the exact student states (Sec. III-C). In addition, another novelty of the AMD-S-Net framework design is the classification of the input data into two types. We design a specific framework for each type of data, according to their special properties: (1) *"consistency supervision"* for the *pairwise data* and (2) *"distribution divergence supervision"* for the *unpaired data* – to fully extract information in each data type (Sec. III-B).

Furthermore, general knowledge distillation methods do not ensure that the teacher is aware of the student's states. This implies that the teacher itself may learn well, but not necessarily teach well. Consider the difference between letting a teacher teach by recording videos vs. by interacting with students. We hereby propose a multi-round online-distillation training paradigm (Sec. III-D) that utilizes the "reset operation" which can ensure that the teacher is aware of the exact student states (e.g., learning process, features, loss, etc). In each round, our training paradigm will first reset the teacher's to the student's states, then let the teacher learn independently in a higher dimensional space to explore the loss landscape near the student space, and guide the student with the local landscape information and potential direction of a better solution, leading to better student performance. This is an advantageous property of the teacher, when the student has converged to a relatively small empirical loss and is unable to further optimize in a stochastic local search.

Experiments show that our AMD-S-Net architecture outperforms other architectures by up to **12.7%**, and our training paradigm outperforms other knowledge distillation (KD) methods by up to **18.1%**. We conduct comparisons on 5 architectures, 10 KD methods, 5 backbones, 4 datasets, and 5 different auxiliary modalities to show their effectiveness. We also perform experiments on other tasks, including waypoints prediction (using RGB + point clouds) and handwritten classification (images + non-image features) to illustrate the generalizability of our method (see Sec. IV).

We summarize our key contributions as follows:

- We propose a novel framework that distill knowledge from multi-modality model to single-modality model in a partially available auxiliary modality setting, i.e., small-shot auxiliary modality distillation network (AMD-S-Net). AMD-S-Net contains a specific framework design to fully distill the information, i.e., *consistency supervision* for the pairwise data and *distribution*

[1]Yu Shen, Luyu Yang, Xijun Wang and Ming C. Lin are with the Department of Computer Science, University of Maryland at College Park {yushen, loyo, xijun, lin}@umd.edu

divergence supervision for unpaired data (Sec. III-B).
- We propose a novel knowledge distillation training paradigm (Sec. III-D) that enables teachers to explore and learn student's local loss landscape information in a higher dimension, thus making it feasible to help student get out of local minimal and boost performance, based on a special "reset operation" that allows the teacher to be aware of the exact student states.

II. Related Work

In our paper, we mainly focus on the setting that there are auxiliary modalities during training but only main modality during test. One kind of solution is treating the multimodal network as the teacher and single modality network as the student, and use the general knowledge distillation methods (Sec. II-A) to transfer the knowledge. Another kind of solution is designing architectures specifically for the modality distillation (Sec. II-B). For the task, we choose end-to-end learning steering under multimodal settings in general (Sec. II-C).

A. Knowledge Distillation

Knowledge distillation is the process to transfer knowledge between networks [4]. Many works have already been done in the general knowledge distillation area. Hinton et al. [4] do early research about distilling the knowledge from an ensemble of models to a single model. Then more and more works have explored the desired knowledge need to be distilled, including intermediate layers' feature [5], [6], attention map [7], paraphrased feature [8], probability distribution in the feature space [9], activation of neurons [10] and etc. Romero et al. [11] propose a method that can distill knowledge from a wide shallow network to a deep thin network (FitNet). VID [12] formulates knowledge transfer as maximizing the mutual information between the teacher and the student networks. Similarity-Preserving Knowledge Distillation [13] aims to preserve the similarity matrix of input data within a mini-batch. CRD [14] encourages the teacher and student to map the same input to close representations and different inputs to distant representations. Some other works [15], [16] focuses on correlation congruence between data samples instead of instance congruence.

Our method can be combined with these methods and achieve better performance. Specifically, our method can reset the teacher to the student's states and lead the student step by step, making it possible to escape local minima and achieve better performance. This is different from the self-distillation methods (e.g., [17], [18]), where teacher and student share the same architecture, while in our setting teacher and student do not need to have the same network architecture.

B. Modality Distillation

Modality distillation mainly focuses on distilling knowledge between different modalities. Gupta et al. [19] learn the representation of one modality with a pretrained network on another modality. Hoffman et al. [20] do an early work about modality hallucination, which contains a hallucination network with RGB image as input but tries to mimic a depth network, then combines with RGB network to achieve multimodal learning. Following works [21], [22] train the hallucination network with a different process to achieve better performance. Some other works [23], [24] use GAN or U-Net to generate another paired modality data with one modality. MSD [25] transfers knowledge from a teacher on multimodal tasks by learning the teacher's behavior within each modality. A latest work [26] trains the different modality data in different pipelines and distills the best modality pipeline knowledge to other modality pipelines. Other than action recognition, modality distillation has also been applied in medical image processing [27]. Existing work of unpaired modality distillation like [28], [29] only consider unpaired data and assume both modalities have enough samples, while ours consider both paired and unpaired data, and also only have small number of auxiliary modality data. Compared to these methods, our method is the first framework that uses consistency supervision for the pairwise data and distribution divergence supervision for the unpaired data (Sec. III-B) and provide flexibility for real-world applications.

C. Multimodal End-to-end Steering

End-to-end steering is an essential task in end-to-end autonomous driving [30]. Multimodal end-to-end steering becomes popular, because of its naturally abundant information and the improvement of multimodal architectures.

Xiao et al. [1] analyze different architectures to fuse multiple modalities in the simulator. Yang et al. [31] make the multimodal data to be the supervision of their multimodal multitask network with only image input. Huang et al. [2] propose a multimodal method with scene understanding. Recently Maanpää et al. [3] design a specific network to fuse camera and lidar data that are suitable for adverse road and weather conditions. Except for spatial methods, Abou-Hussein et al. [32] propose an LSTM-based network to utilize multimodal Spatio-Temporal information.

Compared to these works, ours considers the multimodal end-to-end steering in a more specific setting, i.e., there is a varying amount of auxiliary modality data during training that can reduce costs compared to general multimodal methods while outperforming single-modality techniques.

III. Approach

In this section, we first introduce the task formalization in Sec. III-A. Next, we explain our method in detail in Sec. III-B (AMD-S-Net) under different settings. We introduce a specific *reset* operation and *supermodel* condition in Sec. III-C, which is used by our training paradigm in Sec. III-D.

The novelty of AMD-S-Net is that it's trained with our novel training paradigm and should satisfy the *supermodel* condition to ensure their suitability for our training paradigm with *reset* operation. The framework of AMD-S-Net is also novel because of a specific two-stream framework design. This is the first work that introduces a *reset* operation and

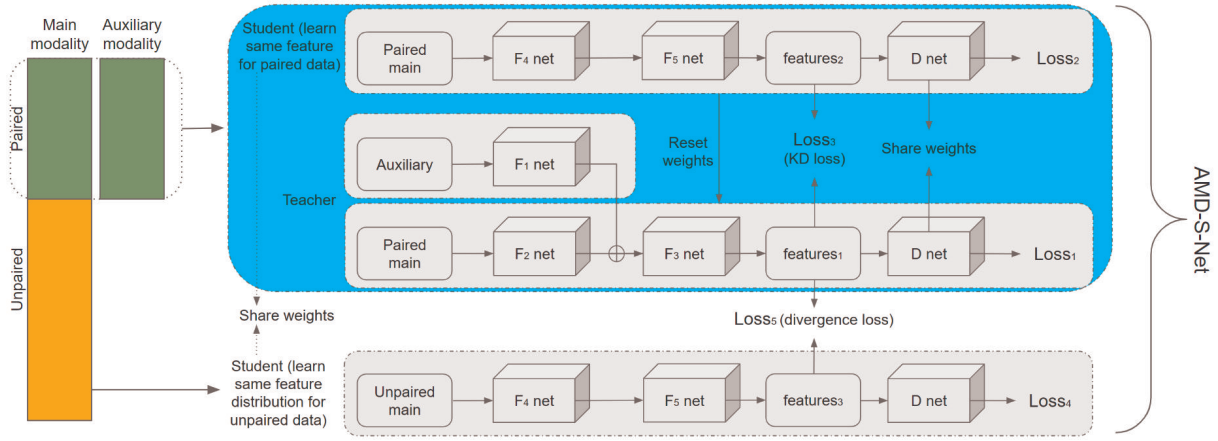

Fig. 1. **AMD-S-Net architecture.** The training process consists of t rounds and each round contains k epochs. At the beginning of each round, the student network will be used to initialize the teacher. In each round of the AMD-S-Net training process, there are 2 steps: (1) Calculate the teacher's loss, $Loss_1$; backpropagate and update teacher's networks F_1, F_2, F_3, D for k epochs; (2) Feed the paired main modality data to the student, calculate the student's loss, $Loss_2$, and feature loss, $Loss_3$, update student's networks F_4, F_5, D, and feed unpaired main modality data to the student, calculate student's loss, $Loss_4$, and divergence loss, $Loss_5$, update F_4, F_5, D, train for k epochs.

supermodel condition that can be utilized by the training paradigm to boost performance.

A. Auxiliary Modalities and Task Formalization

Given an arbitrary task that can be learned by observing a series of task-related data captured by different sensors, or processed using different techniques, we refer to these different but related data types as modalities $\mathbb{I} = \{I_k\}_{k=1}^{K}$, where K is the maximum number of modalities one can obtain with the existing devices, signal prepossessing methods, or expert annotation. We assume that among the K modalities, there is one *main modality* I^M that contributes the most information to the task. The modalities other than I^M are referred as *auxiliary modalities* $\mathbb{I}^A$. Note that each sample from the I^M is not necessarily more informative than each sample from an auxiliary modality. The main modality I^M is considered primary usually because it is the most available hence used-at-*inference* data type. One example of the main modality is the data captured with RGB cameras for autonomous driving tasks, which is plentiful and not expensive, but not necessarily more informative than depth cameras [1], [33], [34].

We first consider a model with learnable parameters θ^m that prioritizes the data from the main modality. The training data from I^M is denoted as $\mathcal{I}^M_{train} = \{i^M_n\}_{n=1}^{N_{train}}$, where N_{train} is the number of training samples from I^M. The parameter that achieves the smallest inference error ϵ^M on $\mathcal{I}^M_{test}$ is denoted as θ^{M*}. With additional data from auxiliary modalities joining the training process, a new model is learned using $\mathcal{I}^{train} = \mathcal{I}^M_{train} \cup \mathcal{I}^A$. The question is, "can we find a better model that achieves a lower inference error on $\mathcal{I}^M_{test}$". In other words, *our goal is to distill complementary information from the auxiliary modalities at training to achieve higher accuracy at test time*.

B. Small-shot Auxiliary Modality Distillation Network (AMD-S-Net)

We first consider the training samples that can find paired matches from both the auxiliary modality and main modality. Our goal is to distill the knowledge from any arbitrary paired I^A and I^M that improves the model that later inference on I^M.

Formally given a task, we denote a learner composed of feature network F and a predictor of fully-connected layers D. We design a student that takes $\mathcal{I}^M_{train}$ as input, and update via iterations of mini-batches,

$$\theta_{stu} \leftarrow \theta_{stu} - \eta \nabla \mathcal{L}^M \tag{1}$$

where θ_{stu} is the parameter of the student network, $\mathcal{L}^M$ is the loss function, and η is the learning rate. Meanwhile, we design a teacher that takes $\{\mathcal{I}^M_{train}, \mathcal{I}^A\}$ as input, and update via an independent feature network F_{tea} (F_1, F_2, F_3 in Fig. 1) and a predictor D that share weights with that of the student network. The teacher network is updated via

$$\theta_{tea} \leftarrow \theta_{tea} - \eta \nabla \mathcal{L}^A \left(D(F_{tea}(\{i^M_n, i^A_n\})), y^M_n \right). \tag{2}$$

The teacher and student learn different representations related to the same task by being exposed to different modalities. The teacher has access to the auxiliary modality I^A, the knowledge of the teacher is distilled to assist the student through a consistency loss $\mathcal{L}_{con}$ that measures the pairwise distance between $F_{stu}(i^M_n)$ and $F_{tea}(i^M_n, i^A_n)$ as part of the student's objective $\mathcal{L}^M$, specifically,

$$\begin{aligned} \mathcal{L}^M = \alpha \mathcal{L}_{sup} \left(D(F_{stu}(i^M_n)), y^M_n \right) + \\ \beta \mathcal{L}_{con} \left(F_{stu}(i^M_n), F_{tea}(\{i^M_n, i^A_n\}) \right) \end{aligned} \tag{3}$$

where $\mathcal{L}_{sup}$ supervises the learning on the main modality.

When auxiliary data is hard to obtain, utilizing a small amount of paired auxiliary data based on the main data is an alternative. We refer to distillation under such a condition as

small-shot auxiliary modalities distillation. Data modalities such as intermediate annotations, expert commentary for hard examples, etc. usually come in *small amounts but are exceptionally informative*, e.g. the doctor's coarse annotation of medical images for tumor segmentation, or human-in-the-loop interactive systems [35]. Except for the consistency supervision by the pairwise feature distance, we also use a *divergence metric* to estimate the difference of the distributions for the unpaired data, such as Kullback–Leibler divergence [36]. During the training (Sec. III-D), after updating the student network via loss, as defined in Eq. 1, for all paired data, we update the student network again with unpaired main modality data via the following loss:

$$\mathcal{L}_u^M = \gamma \mathcal{L}_{sup-u}\left(D(F_{stu}({}_{up}i_m^M)), y_n^M\right) + \lambda \mathcal{L}_{div}\left(\{F_{stu}({}_{up}i_m^M)\}, \{F_{tea}(\{{}_pi_n^M, i_n^A\})\}\right) \quad (4)$$

where $\{{}_pi_n^M\}$ and $\{{}_{up}i_m^M\}$ are paired and unpaired main modality data that meet $\{{}_pi_n^M\} \cup \{{}_{up}i_m^M\} = \mathcal{I}_{train}^M$, and $\mathcal{L}_{div}(,)$ measures the divergence between the distributions of the feature representation sets $\{F_{stu}({}_{up}i_m^M)\}, \{F_{tea}(\{{}_pi_n^M, i_n^A\})\}$. Other training process is shared with the paired process. See this AMD-S-Net framework illustration in Fig. 1 and Algorithm. 1.

One key consideration of this design is, what kind of information is important under this problem setting (input data in Fig. 1). One is the relation between paired main modality feature and the combination feature of paired main and auxiliary modality, which can be extracted by the paired data using consistency supervision knowledge distillation. Another one is the relation between the unpaired main modality feature and the combination feature of main and auxiliary modality. Since the auxiliary modality data is missing for the unpaired main modality data, the combination feature is actually *unknown*. Thus we use the distribution space of combination features of paired main and auxiliary modality to be an approximation of the unknown distribution space of combination features of the unpaired data. Also, since we don't have one-to-one mapping for unpaired data, we use divergence supervision on the distribution-level, instead of consistency supervision on the sample-level. To the best of our knowledge, our AMD-S-Net is the first method that uses consistency supervision for pairwise data and distribution divergence supervision for unpaired data, making this method unique and different from others.

C. Reset Operation

The reset operation plays an important role in our method, but a condition is needed to apply this operation. Inspired by "superset", we introduce "supermodel".

Definition 3.1: Given a model $M_{\theta_A}^{(A)}(I_A)$ (weights θ_A and input I_A), and a model $M_{\theta_B}^{(B)}(I_B)$ (weights θ_B and input I_B), if for any θ_A, there is a θ_B, such that $M_{\theta_A}^{(A)}(I_A) = M_{\theta_B}^{(B)}(I_B)$ for any arbitrary valid input data I_A and its superset I_B. We call model M_B as a *"supermodel"* of M_A.

The "reset operation" is the process of constructing the weights of supermodel θ_B with the given model weights θ_A, defined as:

Definition 3.2: Given a model $M_{\theta_A}^{(A)}(I_A)$ (weights θ_A and input I_A), and its supermodel $M_{\theta_B}^{(B)}(I_B)$ (weights θ_B and input I_B), we define "reset B with A" to be the process of constructing a new θ_B that meet $M_{\theta_A}^{(A)}(I_A) = M_{\theta_B}^{(B)}(I_B)$ for given θ_A and any valid input I_A and its superset I_B.

A simple example is, suppose B is a supermodel of A (e.g., $B = A + A'$), reset B with A is constructing $\theta_B = [\theta_A, 0]$, where θ_A is the weights of A and 0 is the weights of A'. In Fig. 1, the teacher network is a supermodel of the student network, because for any weights of student network, we can construct a teacher network that meet $D(F_{tea}(\{i_n^M, i_n^A\})) = D(F_{stu}(\{i_n^M\}))$ by resetting the F_2 weights with F_4 weights, F_3 weights with F_5 weights, and set F_1 weights to 0. Indeed the reset operation in our method requires that the teacher model is a supermodel of the student model. We also introduce a lemma on the optimal training loss of the supermodel and its base model in Appendix. C.

To summarize: (1) The *supermodel* condition ensures the student parameter space is a subspace of the teacher parameter space, thus enable the *reset* operation. (2) The *reset* operation can reset the teacher to be in exactly the same states as the student, which is then utilized by our training paradigm when the teacher gets far from the student, thus allowing the teacher to explore around the student space and teach local landscape information and potential direction of a better solution to the student, achieving superior performance.

D. Training Paradigm

In this section, we propose a simple yet effective training paradigm based on the "reset operation" (Sec. III-C), which can reset the teacher to exact student states.

Algorithm 1 AMD-S-Net Training Paradigm

Input: Training data from main modality ${}_p\mathcal{I}_{train}^M$ (with paired auxiliary data) and ${}_{up}\mathcal{I}_{train}^M$ (no paired auxiliary data), training data from auxiliary modality $\mathcal{I}^A$ (paired with ${}_p\mathcal{I}_{train}^M$)
Output: student network weights θ_{stu}
Initialisation:
Training Round number t, epoch number in each round k, loss correlation $\alpha, \beta, \gamma, \lambda$, network weights θ_{stu} and θ_{tea}.
for $r = 1$ to t **do**
  Reset teacher weights with student weights
  for $e = 1$ to k **do**
    Feed ${}_p\mathcal{I}_{train}^M$ and $\mathcal{I}^A$ into teacher, update teacher weights θ_{tea} with Eq. 2
  end for
  for $e = 1$ to k **do**
    Feed ${}_p\mathcal{I}_{train}^M$ and $\mathcal{I}^A$ into teacher, and feed ${}_p\mathcal{I}_{train}^M$ into student, update student weights θ_{stu} with Eq. 1 and loss 3
    Feed ${}_{up}\mathcal{I}_{train}^M$ into student, update student weights θ_{stu} with Eq. 1 (replace loss 3 with loss 4).
  end for
end for=0

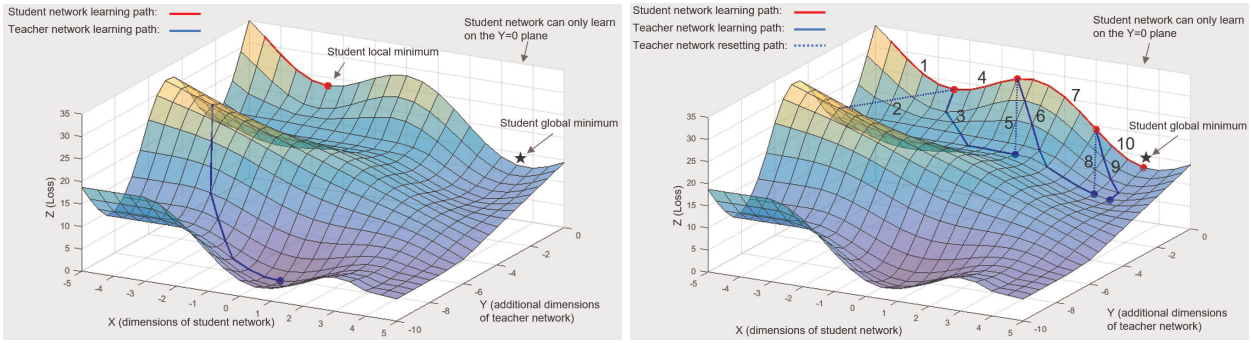

Fig. 2. **Training path comparison on loss landscape.** Given the teacher network is a *supermodel* of the student network, the student parameter space (along X axis with Y=0) is a subspace of the teacher parameter space (XY plane). LEFT: Without our training paradigm, the teacher is not aware of the student states, the training path and the final state of the teacher can be far away from the student space, i.e. the landscape may be totally different, thus providing limited guidance and lead to the student getting stuck in a local minimum. RIGHT: In our method, the teacher is reset to the student states at the beginning of each round, and does optimization with additional dimensions but within a certain range of the student space, teaching the student with local landscape information and potential direction to a better solution. The number 1∼10 is the step order of these processes, see details in Sec. III-C.

	Accuracy (%) on different angle threshold τ (degree)						
Method	$\tau = 1.5$	$\tau = 3.0$	$\tau = 7.5$	$\tau = 15$	$\tau = 30$	$\tau = 75$	**Mean**
Oracle (100% auxiliary modality data)	42.7	68.0	88.0	94.4	96.6	98.6	81.4
one stream (RGB only)	27.3	49.0	77.4	90.2	95.4	98.1	72.9
two streams (shared regressor)	25.9	47.2	77.7	88.4	93.6	97.8	71.8
Modified Xiao et al. [1]	40.8	64.1	84.7	92.7	95.8	98.2	79.4
Modified DMCL [26]	39.1	67.5	88.3	93.9	96.7	98.2	80.6
Ours (AMD-S-Net)	**52.6**	**72.7**	**91.3**	**95.0**	**97.0**	**98.3**	**84.5**

TABLE I

Performance comparison for AMD-S-Net under the small amount of auxiliary modality data setting (20%). OUR METHOD OUTPERFORMS OTHER METHODS BY UP TO **12.7%** MEAN ACCURACY IMPROVEMENT.

As shown in Algorithm. 1, the training paradigm contains t rounds. In each round, we first *reset* the teacher with the student, then train the teacher independently while training the student with both the general label loss and knowledge distillation loss for k epochs. k should not be too large to avoid the teacher being far away from the student. The training process stops when the student converges between different rounds or until finishing t rounds.

Fig. 2 shows the training path comparison on loss landscape between general methods and our training paradigm with *reset* operation. Given the teacher network is a *supermodel* of the student network, the student parameter space (along X axis with Y=0) is a subspace of the teacher parameter space (XY plane). Without the *reset* operation, the teacher is not aware of the student states, the training path and the final state of the teacher can be far away from the student space, i.e. the landscape may be totally different, thus providing limited guidance and lead to the student getting stuck in a local minimum (LEFT of Fig. 2). In our method, the teacher is reset to the student states at the beginning of each round, and do optimization with additional dimensions but within a certain range of the student space, teaching the student with local landscape information and potential direction to a better solution (right part of Fig. 2). Specifically, when the student is potentially stuck in a local minimum (step 1 in the right part of Fig. 2), e.g., already converges with a basic method, we can reset the teacher to the student's states (step 2) and continue to train it (step 3). Then the teacher will be exactly no worse, hopefully better than the student (final position of step 3 is better than the final position of step 1). Then in step 4, which is the distillation training, the student will take both general loss (the force of going downward) and distillation loss (the force of getting closer to the teacher). The distillation loss makes it possible to go upward. After the student pass the loss hill on Y=0, both losses will make it move towards the better solution on Y=0 (final position of step 10).

IV. EXPERIMENTS

We first introduce experiment setups in Sec. IV-A, then show the results on the real-world dataset in Sec. IV-B.

A. Implementation Details

Setting. All experiments are conducted using one Intel(R) Xeon(TM) W-2123 CPU, two Nvidia GTX 1080 GPUs, and 32G RAM. We use the SGD optimizer with learning rate 0.001 and batch size 128 for training. The number of epochs is 2,000. The loss correlations are $\alpha = 1, \gamma = 1$, while β are set with different values for different knowledge distillation methods following [14], and $\lambda = \beta/10$. We pick epoch number in each round $k = 5$ from ablation study of $k = 1, 2, 5, 20$. We set the round number $n = 400$ for Audi dataset and $n = 40$ for Honda dataset. In the experiments, each training process is finished within 24 hours.

Evaluation metric. We use the same evaluation metric as a lastest work [37], i.e., the accuracy w.r.t a threshold τ as $acc_\tau = count(|v_{predicted} - v_{actual}| < \tau)/n$, where n denotes the number of test cases; $v_{predicted}$ and v_{actual} indicate the predicted and ground-truth value, respectively. We compute mean accuracy (mAcc) as $\sum_\tau acc_{\tau \in \mathcal{T}}/|\mathcal{T}|$,

	Mean Accuracy (mAcc in %)		
Method	$20\%\mathcal{I}^A$	$20\%\mathcal{I}^A$ (ours)	Diff
kd [4]	67.7	73.9	**6.2**
hint [11]	72.7	83.1	**10.4**
similarity [13]	66.4	84.5	**18.1**
correlation [15]	68.5	68.7	**0.2**
rkd [16]	71.2	74.6	**3.4**
pkt [9]	73.4	74.4	**1**
abound [10]	70.6	70.6	0
factor [8]	72.2	84.4	**12.2**
fsp [6]	71.6	71.8	**0.2**
Average	70.5	76.2	**5.7**
Teacher (img+seg)	79.4	-	-
Student (img)	72.9	-	-

TABLE II

Performance comparison *with* vs. *without* our training paradigm (containing *reset* operation). BY APPLYING OUR TRAINING PARADIGM ON OTHER KNOWLEDGE DISTILLATION METHODS, WE CAN ACHIEVE BETTER PERFORMANCE IN MOST CASES (UP TO **+18.1%**) IN FULLY PAIRED OR A SMALL AMOUNT OF ADDITIONAL MODALITY DATA.

where $\mathcal{T} = \{1.5, 3.0, 7.5, 15, 30, 75\}$ contains empirically selected thresholds of steering angles.

B. Results on Real Dataset

We perform main comparisons for our key contributions, i.e., AMD-S-Net, and our training paradigm. We also perform other comparisons on different datasets, modalities, and tasks to show the generalizability of our method, as well as performing comparisons for the robustness of our method. More experiments can be found in the **Appendix PDF** in our project page.

Comparison for AMD-S-Net. Since there's no existing method specifically for the small-shot auxiliary modality distillation, we compare our AMD-S-Net with 2 straightforward frameworks and 2 modified frameworks based on SOTA modality distillation methods. We use Audi dataset [38] and Nvidia PilotNet [30] for this experiment. We use 100% RGB images and 20% segmentation data in this experiment. Specifically, the one stream (RGB only) method uses 100% RGB images only with the student network; two streams (shared regressor) method contains RGB and segmentation pipelines with a feature extractor for each pipeline and a shared regressor. For modified Xiao et al. [1] and modified DMCL [26], we keep the 20% paired RGB and segmentation to go through the original pipeline, and let the rest 80% RGB data go through a single RGB pipeline. Table. I shows that our method outperforms other methods by up to **12.7%** mean accuracy improvement.

Combination for our training paradigm. Since our training paradigm can be applied on existing knowledge distillation methods, we conduct experiments by combining ours with kd [4], hint [11], similarity [13], correlation [15], rkd [16], pkt [9], abound [10], factor [8], fsp [6]. One set of experiments use 100% RGB + 100% segmentation, and another set of experiments use 100% RGB + 20% segmentation. From Table. II, our method achieves up to 18.1% improvement in both settings, showing the effectiveness of our training paradigm (containing *reset* operation).

Comparison on different datasets and modalities. We also conduct experiments with different modalities and datasets to show the effectiveness of our method. Specifically, we perform comparison on Audi [38], Honda [33], and SullyChen [39] dataset with RGB image, segmentation, depth map, and edge map modalities. The segmentation is generated by Tao et al. [40], the depth map is generated by [41], and the edge map is generated by DexiNet [42]. Our method outperforms others with up to 11% improvement.

Comparison on different backbones. Except for the Nvidia PilotNet [30], we change the backbone to four other backbones, ResNet [43], ShuffleV2 [44], MobileNetV2 [45], and WRN [46]. Our method outperforms others in all the cases with up to 18.1% improvement.

Comparison on other tasks. Although here we mainly focus on image format auxiliary modalities because it's the most available format, our method can also perform well on other tasks with different data formats, e.g., end-to-end "waypoints prediction task" with point cloud as an auxiliary modality (2.6% improvement), and handwriting classification task with non-image features as auxiliary modalities (2.9% improvement).

Robustness. We also test the robustness of our distilled model following a SOTA work [47] on clean and perturbed Audi dataset (generated with ImageNet-C effects [48]). Our method achieves 4.8% accuracy improvement compared to the RGB only baseline.

V. CONCLUSION

In this paper, we study the problem of how to introduce a variant amount of auxiliary modality data to increase the performance of single modality learning in an end-to-end steering task. We propose a new framework that can take in the main modality and a variant amount of auxiliary modality data to address this problem (AMD-S-Net). In addition, we propose a novel training paradigm that utilizes *reset* operation to help knowledge transfer. Our AMD-S-Net and training paradigm achieve up to 12.7% and 18.1% performance improvement, respectively.

Limitations: Our training paradigm assumes that the teacher network is a *supermodel* of the student network. For general knowledge distillation, which usually distills knowledge from a large network to a small network with different architectures, this requirement can possibly limit overall performance gain. However, for modality distillation, when the goal is to reduce the modality instead of reducing the model size, it is common to use a teacher network that has similar architecture as a student network, except for the additional pipeline for auxiliary modalities, as assumed.

Given that it is possible to use a small amount of expert annotation as the auxiliary modality data to improve the performance, what form of expert annotations can be used in the end-to-end steering task or other tasks would be a possible topic for exploration. Also, under the current setting, the auxiliary modality data is *paired with* the main modality data. It is unclear if the same can be applied to unpaired auxiliary modality data to improve the performance, especially without ground truth.

REFERENCES

[1] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[2] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 781–11 790, 2020.

[3] J. Maanpää, J. Taher, P. Manninen, L. Pakola, I. Melekhov, and J. Hyyppä, "Multimodal end-to-end learning for autonomous steering in adverse road and weather conditions," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 699–706.

[4] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[5] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *arXiv preprint arXiv:1707.01219*, 2017.

[6] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.

[7] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv preprint arXiv:1612.03928*, 2016.

[8] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," *arXiv preprint arXiv:1802.04977*, 2018.

[9] N. Passalis, M. Tzelepi, and A. Tefas, "Probabilistic knowledge transfer for lightweight deep representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2030–2039, 2020.

[10] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3779–3787.

[11] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *International Conference on Learning Representations (ICLR)*, 2015.

[12] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9163–9171.

[13] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1365–1374.

[14] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *International Conference on Learning Representations*, 2020.

[15] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5007–5016.

[16] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3967–3976.

[17] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3713–3722.

[18] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1607–1616.

[19] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2827–2836.

[20] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 826–834.

[21] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 103–118.

[22] ——, "Learning with privileged information via adversarial discriminative modality distillation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2581–2593, 2019.

[23] L. Wang, C. Gao, L. Yang, Y. Zhao, W. Zuo, and D. Meng, "Pm-gans: Discriminative representation learning for action recognition using partial-modalities," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 384–401.

[24] N. Piasco, D. Sidibé, V. Gouet-Brunet, and C. Demonceaux, "Improving image description with auxiliary modality for visual localization in challenging conditions," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 185–202, 2021.

[25] W. Jin, M. Sanjabi, S. Nie, L. Tan, X. Ren, and H. Firooz, "Modality-specific distillation," *arXiv preprint arXiv:2101.01881*, 2021.

[26] N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Morerio, V. Murino, and S. Sclaroff, "Distillation multiple choice learning for multimodal action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2755–2764.

[27] Z. Gao, J. Chung, M. Abdelrazek, S. Leung, W. K. Hau, Z. Xian, H. Zhang, and S. Li, "Privileged modality distillation for vessel border detection in intracoronary imaging," *IEEE transactions on medical imaging*, vol. 39, no. 5, pp. 1524–1534, 2019.

[28] J. Jiang, A. Rimner, J. O. Deasy, and H. Veeraraghavan, "Unpaired cross-modality educed distillation (cmedl) applied to ct lung tumor segmentation," *arXiv preprint arXiv:2107.07985*, 2021.

[29] Q. Dou, Q. Liu, P. A. Heng, and B. Glocker, "Unpaired multi-modal segmentation via knowledge distillation," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2415–2425, 2020.

[30] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[31] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, "End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2289–2294.

[32] M. Abou-Hussein, S. H. Müller, and J. Boedecker, "Multimodal spatio-temporal information in end-to-end networks for automotive steering prediction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8641–8647.

[33] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7699–7707.

[34] S. Huch, A. Ongel, J. Betz, and M. Lienkamp, "Multi-task end-to-end self-driving architecture for cav platoons," *Sensors*, vol. 21, no. 4, p. 1039, 2021.

[35] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016.

[36] J. Goldberger, S. Gordon, H. Greenspan *et al.*, "An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures." in *ICCV*, vol. 3, 2003, pp. 487–493.

[37] M. Shu, Y. Shen, M. C. Lin, and T. Goldstein, "Adversarial differentiable data augmentation for autonomous systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 069–14 075.

[38] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi Autonomous Driving Dataset," 2020. [Online]. Available: https://www.a2d2.audi

[39] S. Chen, "A collection of labeled car driving datasets, https://github.com/sullychen/driving-datasets," 2018.

[40] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

[41] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *Advances in neural information processing systems*, vol. 32, pp. 35–45, 2019.

[42] X. S. Poma, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust cnn model for edge detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1923–1932.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical

guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.
[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
[46] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.
[47] Y. Shen, L. Zheng, M. Shu, W. Li, T. Goldstein, and M. C. Lin, "Gradient-free adversarial training against image corruption for learning-based steering," in *Neural Information Processing Systems (NIPS)*, 2021.
[48] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.
[49] W. Li, D. Wolinski, and M. C. Lin, "ADAPS: Autonomous driving via principled simulations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7625–7631.
[50] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7077–7087.
[51] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
[52] Z. Han, C. Zhang, H. Fu, and J. T. Zhou, "Trusted multi-view classification," *arXiv preprint arXiv:2102.02051*, 2021.
[53] UCI, "Multiple Features Data Set, https://archive.ics.uci.edu/ml/datasets/multiple+features," 1998.
[54] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[55] M. van Breukelen, R. P. Duin, D. M. Tax, and J. Den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.