2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

DMFuser: Distilled Multi-Task Learning for End-to-end Transformer-Based Sensor Fusion in Autonomous Driving

Pedram Agand*, Mohammad Mahdavian*, Manolis Savva and Mo Chen

***Abstract*— In end-to-end autonomous driving, current sensor fusion and navigational control techniques used by imitation learning algorithms are insufficient in challenging scenarios involving multiple dynamic agents and result in poor driving capabilities. To tackle this issue, we introduce DMFuser, a transformer-based algorithm that employs knowledge distillation between multi-task student and single-task teachers and combines attention and convolutions to fuse multiple RGB-D camera representations to produce vehicular navigational commands (throttle, steering and brake). Our model incorporates two modules. The first module, perception, encodes data from RGB-D cameras for tasks like semantic segmentation, semantic depth cloud (SDC) mapping, and traffic light state recognition. To enhance feature extraction and fusion from both RGB and depth sources, we harness local and global capabilities of convolution and transformer modules. We employ an attention-CNN fusion structure to effectively learn and fuse RGB and SDC map features. Subsequently, the control module decodes these features along with supplementary data, containing environment's static and dynamic information, to predict waypoints and vehicular control actions. We evaluate the model and conduct a comparative analysis, in various scenarios, weather conditions, and traffic situations, spanning from normal to adversarial in the CARLA simulator. We achieve better or comparable results in term of driving score (DS) and other metrics with respect to our baselines. Also, our ablation studies demonstrate the effectiveness of our contributions to improve the driving skills. Our code is available at the following github page: `https://github.com/pagand/e2etransfuser`**

I. INTRODUCTION

Many works in the autonomous driving (AD) literature have been focusing on different aspects of perception and control tasks for safe navigation [1], [2]. Recent advances in end-to-end driving neural network (NN) models have demonstrated remarkable results using single modality inputs, such as image and LiDAR [3]. However, these approaches face limitations in complex urban scenarios involving adversarial situations due to their lack of 3D scene understanding [4].

Sensor fusion has shown promise in addressing these challenges by integrating multiple modalities, to create a more comprehensive scene representation [5], [6]. Despite the improvements, these methods often require large computational resources and face challenges in balancing learning signals between perception and control tasks [7]. Moreover, integrating multiple modalities with different data shapes and representations requires advanced preprocessing techniques.

Among the recent advancements in end-to-end AD, a notable approach is presented by Natan et al. [8], wherein a

The authors are with School of Computing Science, Simon Fraser University (SFU), Burnaby, Canada {pagand, mmahdavi, msavva, mochen}@sfu.ca
* These authors contributed equally to this work

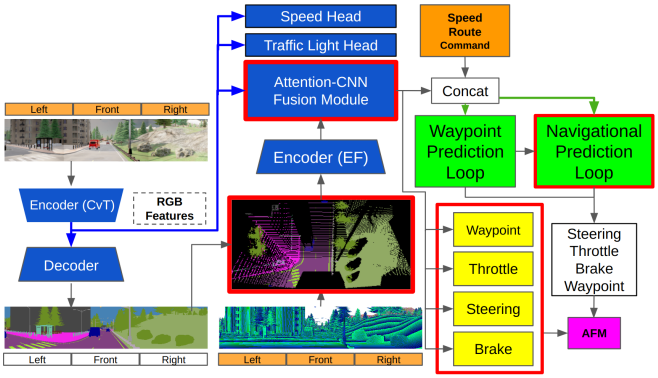


Fig. 1: Model architecture: Perception, controller and inputs are represented by blue, green and orange colored items, respectively. The single-task teachers, represented by yellow boxes, are combined with the multi-task network, which is responsible for simultaneously learning waypoints and navigational commands, to create adaptive feature matching (AFM), illustrated by a purple box. The red boxes around blue (Attn), green (VC), and yellow (Dist) items and the SDC map (SDC) declare our most important contributions.

Convolutional Neural Network (CNN) is utilized to extract features from RGB-D camera. The extracted information is fused to obtain future vehicle waypoints from semantic depth cloud (SDC). Another method by Chitta et al. [9] employs a multi-modal fusion (MMF) transformer to fuse global information and pairwise interactions into the feature extraction layers of distinct input modalities. Meanwhile, the Trajectory-guided Control Prediction (TCP), combines trajectory planning and direct control for end-to-end AD [10]. Inspired by these three methodologies, we aim to derive more efficient, yet accurate solution for end-to-end AD.

Our model is summarized in Fig. 1 with our most important contributions demonstrated in red boxes. Inspired from [11], we propose a method for end-to-end AD that leverages the complementary advantages of RGB and depth information provided by three RGB-D cameras, addressing the challenges in single modality and sensor fusion approaches. Our model consists of two main modules: the *perception module*, which encodes high-dimensional observation data and performs semantic segmentation (SS), SDC mapping, sensor fusion, and ego vehicle speed and traffic light prediction; and the *control module*, which decodes the features encoded by the perception module combined with GPS, command and speedometer information to predict waypoints and control actions.

For perception (blue boxes in Fig. 1), we utilize CvT and EfficientNet [12] to extract RGB and SDC map features. We

have improved the SDC map developed by Natan et al. [8] to include more information of the surroundings which helps to better make navigational decisions. We also developed an attention-CNN fusion module that harnesses the strengths of global context reasoning enabled by visual transformers [13] combined with local feature learning of CNNs.

For our control structure (Green boxes in Fig. 1), we utilize two branches to process the outputs of the perception module, thereby promoting diversified and resilient decision-making. Two interconnected loops with Gated Recurrent Unit (GRU) were developed to address dynamic and static behaviors in the environment, generating waypoints and vehicular control actions. A PID agent is used for the waypoint branch, while an MLP agent is employed for the vehicular control branch, computing the final navigational command for the ego vehicle. To balance learning signals and ensure a uniform learning pace across all tasks, we employ a Modified Gradient Normalization (MGN) method [14], along with knowledge distillation to enforce feature matching across single-task networks via AFM.

Finally, our model is evaluated on the CARLA simulator with various scenarios demonstrating improved performance against baseline methods. To summarize, our contributions are as follows: (1) developing an efficient attention-CNN based sensor fusion module to better fuse the sensor data on the global and local levels. (2) Introducing vehicular control structure to predict future navigational commands by learning environment static/dynamic behaviours. (3) Utilizing distillation knowledge from single task teachers to boost overall performance of multi-task student. (4) Improving SDC map to encompass more contextual and broader information.

II. Related works

Multi-Modality: Advancements in multi-modal autonomous driving have highlighted the potential of using RGB images alongside depth and semantic information to enhance driving performance [15]. In our work, we focus on combining RGB and Depth inputs, which are readily available in autonomous vehicles and provides complementary scene representations in autonomous systems.

Sensor Fusion: Most of the recent sensor fusion research has focused on perception tasks such as object detection and motion forecasting [16], [17]. We first implemented a multi-scale geometry-based fusion mechanism inspired by authors in [18], [19], but found it insufficient for complex urban driving situations. Our proposed attention-CNN-based multi-modal fusion module incorporates global and local contextual reasoning which improves performance.

Bird's Eye View (BEV): Researchers project the depth map with the SS to create a semantic depth cloud (SDC) with a BEV perspective [8]. Huang et al. [20] fused RGB and depth to capture a deeper global context, while Prakash et al. [21] combined RGB and preprocessed LiDAR point clouds to leverage front-view and BEV using sparse GPS locations. We consider a sequence of waypoints instead of high-level navigational commands, as it better reflects real-world driving conditions [22].

Imitation Learning (IL): Studies in AD usually fall into two categories: reinforcement learning (RL) and IL. Authors in [23] have shown the potential of RL, while IL approaches such as LBC [1] and NEAT [24] have shown impressive performance. Our work adapts the auto-regression scheme in IL [9].

Multi-task (MT) learning: MT aims to learn a shared representation by utilizing information from related tasks' training data [25], [26]. Examples of MT approaches for scene understanding include task weighting schemes [27], gradient-based methods like GradNorm, and attention-based methods [28]. Our work utilizes MGN [14] to handle imbalances in task difficulty, dynamically prioritize tasks, and compute weights based on loss values.

Knowledge distillation (KD): Originally, Ma and Mei [29] have explored KD by transferring knowledge from ensemble or pre-trained models. Instead of using pre-trained models or using auxiliary task predictions, Jacob et al. [30] jointly trains single and multi-task (MT) networks.We used bottleneck knowledge distillation from different single-task students to jointly update the MT network.

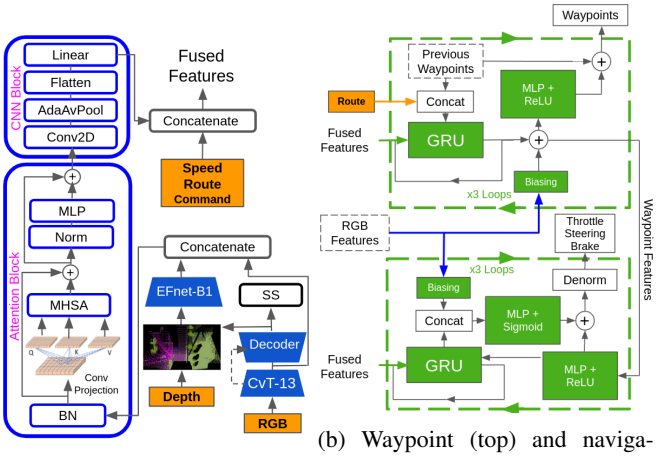
End-to-End AD: These approaches offer training efficiency and integration simplicity. IL-based methods have been extensively studied for AD tasks with the former approach delving into additional perception tasks to enhance feature extraction [1]. Combining diverse AD tasks, including object detection, lane detection, and SS within a MT framework, has demonstrated remarkable performance [31], [32]. In this study, we train two agents to control the vehicle according to the scene situation. The first agent employs an on-the-fly PID approach rmmbased on the predicted waypoints. The second one involves a multi-layer perceptron (MLP) for vehicular control within an end-to-end framework [8]. Unlike InterFuser [33], we avoid using extra collision avoidance steps during evaluations which makes our model more realistic. Also, opposite to InterFuser [33] and ReasonNet [34], we do not require multiple GPUs during training. We also have reduced the number of parameters used for data fusion to almost 35% of those used in TransFuser [9] for feature extraction, fusion, and generating vehicle motion.

III. Methodology

Our method is structured around two key components: the perception and control modules, each fulfilling a specific set of tasks to enable the vehicle navigation. The overal structure of our method is summarized in Fig. 1 and perception and control modules are further explored in Fig. 2.

A. Perception Module

As illustrated in Fig. 2a, the perception module is designed to provide an interpretable representation of the driving environment using RGBD cameras. To achieve this, we extract features from RGB images utilizing the CvT [13] to perform SS and subsequently generate a BEV SDC map. The SDC map offers a top-down view of the vehicle, highlighting surrounding objects, their classifications, and corresponding distances. This accurate map represents the most comprehensive information required for effective navigation.



(a) Perception module

(b) Waypoint (top) and navigational (bottom) branches

Fig. 2: a) In perception, we use attention-CNN fusion module to concatenate RGB and SDC features and pass the tensor through two-stepped fusion module to learn the global and local feature. b) The top and bottom figures, show the waypoint and navigational prediction control loops, respectively. The process inside the green dashed line boxes are iterated three times during the training, in which the model predicts waypoints and vehicular controls.

Our perception module processes three RGB images: one from the front-facing camera and two from side cameras angled 60 degrees to the left and right. The front RGB and depth images have a resolution of 160×320, while the non-overlapping side cameras capture images with a resolution of 160×224, resulting in a total of 160×768 pixels for both RGB and depth images.

1) Convolutional Vision Transformer (CvT): We employed CvT [13] that has been pretrained on the ImageNet [35] as our feature extractor. CvT is responsible for obtaining information from three RGB images shown in Fig. 2a as the "Encoder (CvT)". By combining the CNN's local and transformer's global feature extraction ability, CvT ensures that our feature extractor is capable of capturing both local and global features. CvT-13 [13], a light version of the CvT, has been selected to have a reasonable model capacity and avoid over-fitting at the same time. Therefore, after passing the RGB images, the extracted features contains 384 2D components, each with a size of 10×48.

2) Semantic Segmentation: After the feature maps are extracted, they are exploited in different sections of the model. First, we train a SS decoder capable of accurately identifying 23 different classes depicted in the Fig. 2a as "Decoder". Later, a segmentation decoder consisting of three convolutional layers uses a final pointwise convolution with sigmoid activation to create a segmentation map. By leveraging residual connections, we can render accurate and comprehensive segmentation maps.

3) Semantic Depth Cloud (SDC): The SDC map represents the ego-vehicle's surrounding environment and contains information with respect to 23 semantic class layers. Each layer represents one class of environment object. We define and cover a 64-meter distance range in front of the vehicle and 32 meters to each side of the camera. The SDC maps have dimensions of 160×320 for the front and 160×224 for the sides. We generate a transformation matrix for the x-axis using camera parameters and normalize the coordinates to align with the SDC tensor spatial dimension. One-hot encoding is applied to yield a 23-channel SDC tensor. The resulting maps are copied to an empty tensor with dimensions of 160×768, with side maps rotated at a 42-degree angle to create an aligned 180 degrees labeled point-cloud. For SDC map feature extraction, we use the compact EfficientNet-B1 [12] demonstrated in Fig. 2a as EFnet-B1 network to generate a tensor of 192 features, each with a size of 10×48.

4) Attention-CNN Fusion: In order to enhance the performance of our model, we utilize a fusion module that is anchored on attention-CNN combination to effectively process features drawn from both RGB images and the SDC map. Fig. 2a shows the attention mechanism that leverages the global feature-learning ability while making the most of CNN's skill for learning local features. We concatenate the features extracted from RGB images and the SDC map on features dimension and apply a layer of batch normalization. Following that, we guide these normalized features through our attention block. We pass the features through 2D convolutional projection layers to create Query, Key and Value tensors. Next, multi-head self-attention (MHSA), normalization and MLP layer in combination with residual connections are applied. One concern for this approach is that the RGB features come from perspective point-of-view as the SDC map features are extracted from a top-down angle. But similar studies have shown that transformers are capable of fusing the features from different modalities and point-of-views such as RGB and 3D Lidar data [9]. To better learn the local features, the outputs are passed through a CNN based module. It consists of a 2D convolution layer followed by an adaptive average pooling layer, flattening and a linear layer to reduce the features size. These fused visual features are concatenated with navigational measurements to create the fused features.

B. Controller Module

The control module depicted with green color items in the Fig. 1 receives extracted RGB image, SDC map features, and navigational measurements including route location, cruising command provided by the global planner and the ego vehicle speed. The cruising command specifies the vehicle's general direction, such as left, right, forward, stop, etc., and is embedded with a one-hot vector. The controller module predicts the appropriate navigational commands, including steering, throttle, and brake. To predict them as a sequence, we employ a GRU that is a suitable choice as it addresses the vanishing gradient problem while maintaining a better performance-cost ratio compared to other RNNs. To train a model that predicts current control actions based on current input, behavior cloning is commonly used but relies on the assumption of independent and identically distributed (IID) data, which is not valid for closed-loop tests [10]. To overcome this issue without resorting to RL, we predict multi-step control actions into the future as a sequence. To

this end, we first employ a waypoint branch that utilizes fused visual features concatenated with navigational measurement (creating fused features) along with environment-agent static knowledge through RGB features. Further, we deploy a navigational branch to capture the environment-agent dynamic interaction given the learned static knowledge. The navigational branch provides information regarding objects' motion and traffic light changes, while the waypoint branch incorporates static information like curbs and lanes to improves spatial consistency across branches.

1) Waypoint Branch: The GRU in the waypoint branch, depicted in the top half of Fig. 2b, takes fused features as the initial hidden state, and the current waypoint in the BEV space with the route location coordinate transformed to the BEV space as the inputs. The next hidden state from the GRU is added to RGB features (RGB features that have passed through a biasing module). The bypassing module consists of adaptive global pooling and a linear layer applied to the RGB features followed by a sigmoid function. We apply a MLP network containing two linear layers and a rectified linear unit (ReLU) to the biased GRU hidden state.

2) Navigational Branch: The GRU in the navigational branch, shown in bottom half of Fig. 2b, takes the same fused features as the waypoint branch for the initial hidden state to improve consistency. It uses the predicted waypoint features from the waypoint branch as GRU input to transfer knowledge between branches. The result then concatenate with the same extracted RGB features, representing the abstract static coarse simulator and then fed to MLP and sigmoid to create adjusted control output. To determine the suitable vehicular control, we compute them in two different ways. First, for MLP agent, we denormalized the summation of the predicted vehicular command from navigational branch. Second, for PID agent, we use two separate PID controllers; one for finding the steering command (lateral) and the other for finding the throttle and brake (longitudinal) based on the predicted waypoints. Our control policy calculates the control commands using both methods based on the scenario.

C. Training via Knowledge Distillation

We have four teacher networks demonstrated as yellow boxes in Fig. 1, each with a single-task objective: to individually imitate waypoints and vehicular controls. At the beginning, the three vehicular control networks are identical and have 1/3 the output size of the MT network, whereas the waypoint network is identical to the waypoint head in MT network. All these single-task networks are isolated and only distill knowledge to the MT via AFM. Since different layers of the shared backbone contribute differently to each task [36], we distill features from bottleneck nodes of the single-task networks to the MT network in each iteration during training. This is achieved via AFM, a method for sharing intermediate features of the backbone models. Let γ be an adaptive weight based on the convergence level of the single-task networks. The AFM loss function is defined as:

$$\mathcal{L}_{\text{AFM}} = \gamma \sum_{i} \left\| BN_{\text{MT}}^{(i)} - \sum_{j \in \{W,S,T,B\}} BN_{j}^{(i)} \right\|^2 \quad (1)$$

where, $BN_{\text{MT}}^{(i)}$ is the i-th bottleneck feature of the MT network, and $BN_{j}^{(i)}$ for $j \in \{W, S, T, B\}$ are the corresponding features in the single-task networks for the waypoint, steering, throttle, and break, respectively. For the adaptive weight we consider the following form $\gamma = a_0\sqrt{a_1 \times \text{epoch} + a_2}$, where a_i are tuning parameters. One can optimize them according to how fast the single task networks are roughly valid (maturity rate). To compute the losses in the MT network, we need to compute each loss separately. A mixture of dice and binary cross-entropy loss is employed to calculate the SS loss ($\mathcal{L}_{\text{SEG}}$), which is computed using the following:

$$\mathcal{L}_{\text{SEG}} = 1 - \frac{2|\hat{y} \cup y|}{|\hat{y}| + |y|} + \frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \quad (2)$$

where $N, y_i, \hat{y}_i$ represent the count of the elements in the SS decoder final layer, the values of the i-th element in the ground truth, and the corresponding prediction, respectively. Conversely, simpler L1 loss functions are employed for the remaining tasks, including the traffic light state loss ($\mathcal{L}_{\text{TL}}$), steering loss ($\mathcal{L}_{\text{ST}}$), throttle loss ($\mathcal{L}_{\text{TH}}$), brake loss ($\mathcal{L}_{\text{BR}}$), velocity loss ($\mathcal{L}_{\text{VE}}$), and waypoints loss ($\mathcal{L}_{\text{WP}}$). Providing supplementary loss criteria to the SS task is imperative, since the remaining components of the network structure rely on it. As we have multiple predicted waypoints, waypoints loss necessitates averaging. However, for the vehicular control, we only use the first prediction. Ultimately, the final loss can be calculated as follows:

$$\mathcal{L} = \sum_{i} \lambda_i \mathcal{L}_i \quad (3)$$

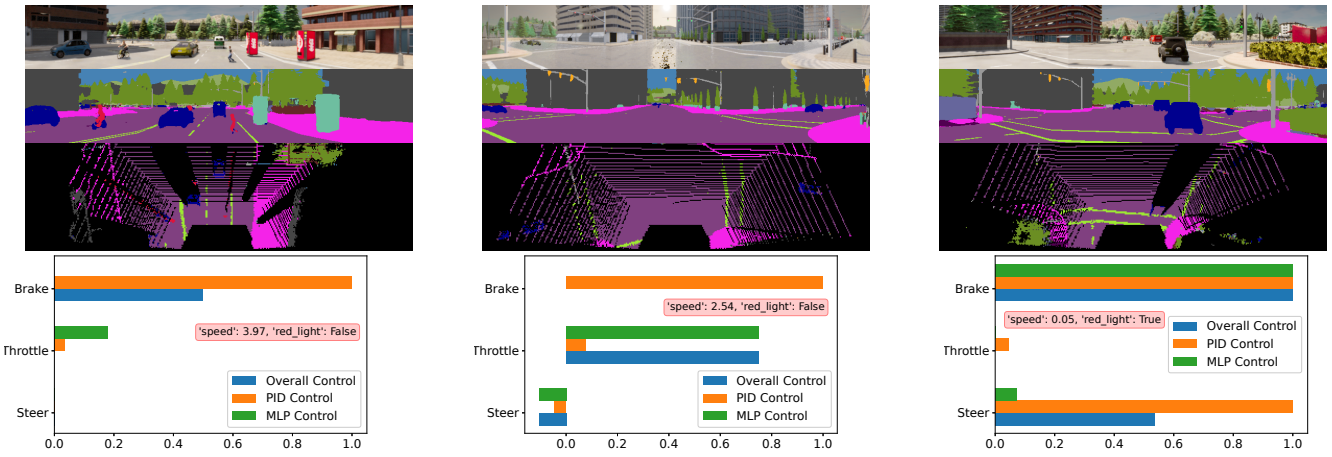
where (i) includes all the learned tasks in the MT network and AFM. The $\lambda_{(.)}$ is an adaptive MT learning rate obtained from MGN to adaptively adjust the loss weights.

IV. Experiments

This section covers the process of generating and representing data, as well as defining the task and presenting various evaluation scenarios for the model. We employed Adam optimizer with a decoupled weight decay of 0.001. Initially, the learning rate was set to 0.0001 and gradually halved if the validation metric shows no decline for three consecutive epochs. Also, to reduce computational expenses, training was halted if there is no progress for 15 consecutive epochs or reached the maximum of 60 epochs. The model was implemented using the PyTorch framework [37] trained on a single A100 80GB GPU with a batch size of 40 and evaluated on an NVIDIA GeForce RTX-3090.

A. Dataset

We used CARLA [4] (0.9.10) for the simulating environment which has 8 available towns for training and testing. We generated the dataset based on TransFuser [9] augmented with all standard CARLA long and tiny paths containing total of more than 350K frames. For more detailed information about the data, please refer to the supplementary file.



(a) Pedestrian crossing suddenly (b) Turning left in intersection (c) Sudden red traffic light

Fig. 3: Qualitative results: (a) Wet Noon: Our model effectively stops the ego vehicle as a pedestrian suddenly crosses the road, thereby preventing a collision. (b) Hard Rain Sunset: Following a green traffic light, our model predominantly follows the MLP agent for waypoint tracking while avoiding veering onto the sidewalk during left turns. (c) Wet Noon: When the traffic light changes to red while other vehicles are in motion, the vehicle maintains a complete stop.

TABLE I: Performance comparison of DMFuser (ours) with baselines: E2E-F/A [8], MMF-F/A [38], TF-A [9], and Expert.

Type	Model	Normal			Adversarial		
Town5	↑	**DS**	RC	IP	**DS**	RC	IP
Clear Noon Short	E2E-F	50.08 ± 4.53	67.51 ± 3.59	0.66 ± 0.00	38.30 ± 1.34	62.84 ± 8.22	0.57 ± 0.06
	E2E-A	56.45 ± 9.28	74.58 ± 10.98	0.67 ± 0.02	41.35 ± 3.26	65.26 ± 12.23	0.56 ± 0.07
	MMF-F	11.00 ± 4.07	13.46 ± 5.80	0.91 ± 0.00	8.14 ± 0.59	10.64 ± 0.68	0.88 ± 0.01
	MMF-A	12.9 ± 5.11	14.58 ± 5.17	**0.93** ± 0.01	12.81 ± 2.28	15.61 ± 2.69	**0.88** ± 0.03
	TF-A	81.22 ± 1.16	95.74 ± 0.61	0.84 ± 0.01	62.65 ± 3.09	90.26 ± 4.37	0.69 ± 0.05
	Ours	**82.28** ± 0.76	**96.78** ± 1.73	0.86 ± 0.04	**68.85** ± 2.47	**91.07** ± 0.24	0.75 ± 0.03
All Weathers Short	E2E-F	42.19 ± 2.93	64.34 ± 0.57	0.65 ± 0.02	36.65 ± 5.15	65.20 ± 5.65	0.54 ± 0.01
	E2E-A	47.96 ± 0.33	74.05 ± 4.91	0.59 ± 0.04	37.80 ± 4.50	61.03 ± 2.25	0.55 ± 0.12
	MMF-F	11.02 ± 3.91	13.47 ± 5.39	0.91 ± 0.02	9.96 ± 1.44	13.06 ± 1.92	**0.88** ± 0.01
	MMF-A	12.52 ± 1.39	14.34 ± 1.40	**0.91** ± 0.00	11.86 ± 2.04	13.92 ± 3.52	0.88 ± 0.02
	TF-A	**83.86** ± 1.09	97.66 ± 0.73	0.85 ± 0.00	62.59 ± 2.61	86.40 ± 2.24	0.73 ± 0.01
	Ours	83.45 ± 3.22	**98.15** ± 2.23	0.85 ± 0.01	**66.71** ± 2.49	**95.76** ± 2.48	0.69 ± 0.04
	Expert	99.33 ± 0.82	99.95 ± 0.05	0.99 ± 0.00	77.79 ± 2.66	96.82 ± 2.09	0.80 ± 0.041
Clear Noon Long	E2E-F	11.11 ± 0.23	60.69 ± 5.93	0.29 ± 0.06	14.49 ± 7.93	35.48 ± 7.51	0.51 ± 0.03
	E2E-A	13.43 ± 8.14	47.20 ± 1.17	0.32 ± 0.04	7.98 ± 0.05	25.92 ± 3.03	0.44 ± 0.12
	MMF-F	3.67 ± 0.95	5.16 ± 2.39	0.87 ± 0.07	2.97 ± 0.02	4.51 ± 0.25	0.88 ± 0.02
	MMF-A	5.34 ± 1.06	5.76 ± 0.52	**0.96** ± 0.02	4.69 ± 0.12	4.73 ± 0.14	**0.97** ± 0.01
	TF-A	35.46 ± 3.97	**93.32** ± 5.36	0.39 ± 0.02	30.76 ± 5.80	**84.40** ± 0.83	0.42 ± 0.08
	Ours	**52.98** ± 1.83	86.43 ± 5.32	0.57 ± 0.03	**45.24** ± 2.95	76.26 ± 1.42	0.58 ± 0.25
All Weathers Long	E2E-F	6.55 ± 2.73	60.30 ± 10.03	0.15 ± 0.09	9.46 ± 1.32	45.51 ± 3.84	0.38 ± 0.09
	E2E-A	7.18 ± 1.03	45.29 ± 7.61	0.28 ± 0.07	9.02 ± 0.77	34.70 ± 4.85	0.34 ± 0.00
	MMF-F	3.75 ± 0.94	4.12 ± 1.29	0.94 ± 0.02	3.99 ± 0.39	5.72 ± 2.13	0.91 ± 0.06
	MMF-A	4.13 ± 0.82	4.15 ± 0.85	**0.99** ± 0.01	5.28 ± 0.82	4.723 ± 2.01	**0.98** ± 0.02
	TF-A	42.81 ± 1.69	**97.14** ± 1.16	0.43 ± 0.02	**32.83** ± 1.95	76.28 ± 11.71	0.52 ± 0.08
	Ours	**42.94** ± 1.28	96.87 ± 4.15	0.45 ± 0.03	32.69 ± 3.87	**81.90** ± 0.95	0.49 ± 0.07
	Expert	70.31 ± 13.43	96.80 ± 4.51	0.73 ± 0.17	26.71 ± 4.77	68.27 ± 13.29	0.52 ± 0.13

B. Evaluation Metrics

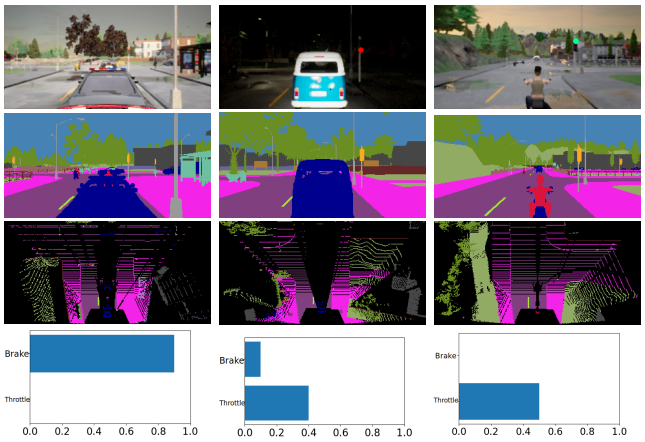
As per CARLA leaderboard evaluation setting, we have employed the driving score (DS) for our principal metric. The higher the DS value, the more exemplary the driving ability. The DS for a given route (DS_i) is calculated by the product of the percentage of the route that was completed (RC_i) and the corresponding infraction penalty (IP_i) as follows:

$$\text{DS} = \frac{1}{N_r}\sum_{i=1}^{N_r} \text{RC}_i \times \text{IP}_i \tag{4}$$

where RC_i is calculated by dividing the distance correctly driven to its total length for each route, and N_r is the total number of the routes. This calculation excludes any incorrect paths taken (e.g., driving on sidewalks). To calculate IP_i, we consider different infractions penalties ($0 < p^j < 1$) similar to [9] and calculate it as follows:

$$\text{IP}_i = \prod_{j=1}^{M} (p_i^j)^{\#\text{infractions}_j}, \tag{5}$$

where M denotes the number of different infractions. The IP_i at the beginning of each route is 1.0, and it decreases each time an infraction occurs. The final RC and IP scores are calculated by averaging over different routes.



(a) Behind a car (b) Toward a van (c) Turns bicycle

Fig. 4: Qualitative results showing front camera RGB, SS, and SDC map with throttle and brake values: (a) The agent has stopped behind a police car (b) The agent is moving toward a high ceiling van at the intersection and is reducing throttle and increases the brake value to stop. (c) The agent has stopped with a safe distance to the bicycle and slightly increases throttle once the light turn green.

C. Baselines

We have opted to compare our method with some of the state of the art (SOTA) techniques in Table I. As our first baseline, we have selected E2E [8], which mainly relies on EfficientNet to extract features for identifying the vehicle's navigational waypoints. We trained two different versions of E2E: the first one (E2E-F) is identical to the model presented in the paper that uses RGB and depth information (RGB-D) as input, and the second one (E2E-A) includes three RGB-D data from left, front, and right sensors similar to ours. This will show the potential for utilizing multiple sensors instead of one, while facilitating a fair comparison with other models. To ensure a fair and comprehensive comparison with E2E-F and our approach, we refined the SDC mapping used in the E2E-A to correctly include all the surrounding information. We also elevated the dimensions of the CNN fusion, waypoint and navigational prediction to enhance predictions based on the fused features. For the second baseline, we implemented two versions of the MMF: MMF-F/A [21]. In the first version, MMF-F, according to the baseline, the algorithm utilizes a combination of ResNet and transformer architecture to process an RGB image and LiDAR data (RGB-L). In MMF-A, we fed three RGB-L and scale the network to match the inputs. As a third baseline, we also considered an enhanced version of the MMF called Transfuser (TF-P) [9]. TF-P benefits from a relatively large size model that receives 3 RGB-L to generate waypoints. We avoid comparing our work with some recent methods such as InterFuser [33] as they use extra heuristic obstacle avoidance steps while evaluation which makes the comparison unfair. Also, our focus in this has been to develop a method that can be trained with a single GPU in a reasonable amount of time and therefore we avoid comparing with large sized models such as ReasonNet [34] and InterFuser [33].

D. Results

Table I presents the final results for different methods. The bold and underlined numbers show the best and second best results, respectively. To achieve confidence in the results, we simulated each scenario atleast twice and report the results' average and standard deviation between the runs. Please note that a higher IP or RC does not necessarily indicate better driving performance. A vehicle may complete all routes and receive a high RC, but drive poorly, resulting in low IP and DS, or vice versa. As one can see in Table I, our method achieved the highest DS scores in all clear noon scenarios. Also, in all weather paths, we rank the best DS for normal and second best for adversarial scenarios. These findings demonstrate both the accuracy and conservativeness of our approach.

While E2E-F and E2E-A performed reasonably well, their performance was not comparable with our method despite our modifications on E2E-A. This shows the effectiveness of our perception, fusion and control modules as E2E-A and our method both receive similar inputs (RGB and depth) to generate the same outputs (navigational commands). Also, both MMF-F and MMF-A exhibited poor performance. This is likely due to the highly conservative nature of these methods, which causes them to stop frequently during driving, resulting in high IP score but low DS. TF-A on the other hand, performed better but still inferior compared to our method in almost all scenarios. In general, long routes pose a greater challenge with respect to short ones, where rare accidents or infractions can reduce DS.

We have also conducted qualitative analysis of driving footage under various weather conditions presented in Fig. 3, which yielded several notable observations. First, it has been established that all models face challenges in adversarial scenarios, necessitating advanced perception and control modules. Second, the proposed model has achieved the highest DS due to its ability to respond effectively to anomalies. The model's performance is further enhanced by its capacity to strike a balance between RC and IP by leveraging the expertise of two agents to comprehend various facets of driving. The vehicle comes to a full stop when both throttle values are less than the 0.2 threshold. In cases where the MLP agent's throttle value exceeds 0.2, full control is handed over to the MLP agent (scenario b), and the same principle applies to the PID agent (scenario a). This implies that the MLP agent, derived from the navigation branch, excels in precise movements, while the PID agent from the waypoint branch demonstrates greater agility in response to sudden environmental changes. Finally, the augmentation of the center camera and depth with side sensors as a single image has enabled the ego vehicle to expand its knowledge to encompass a perpendicular viewpoint while reducing the input-output time, thereby facilitating more efficient training.

In order to further investigate our model performance, we have extended our qualitative comparison on more specific cases shown in Fig. 4, that the RGB and SDC map fusion plays a more significant role. Comparing Fig. 4a and 4b demonstrates the model's dependence on SDC information

TABLE II: Detailed DMFuser ablations: without (1) side SDC maps (no-SDC), (2) vehicular control GRU loop (no-VC), (3) attention CNN (no-Attn), and (4) distillation (no-Dist). The metrics include collision with Pedestrian (Ped), Vehicles (Veh), and Layout (Lay) in addition to violation in Redlight (Red), Stop sign (Stop) and Offroad (Off), and Agent block (AB)

Type	Model	Scores↑			Collisions↓			Violations↓			
Town5	Adversarial	**DS**	RC	IP	Ped	Veh	Lay	Red	Stop	Off	AB
Average of hard weathers	Ours-No-SDC	25.542	68.908	0.453	0.042	0.790	0.407	0.204	0.683	**0.064**	0.681
	Ours-No-VC	18.427	54.145	0.482	0.042	0.193	0.652	0.054	0.162	0.243	0.843
	Ours-No-Attn	13.282	30.443	**0.604**	0.089	0.320	0.555	0.043	**0.082**	0.256	1.654
	Ours-No-Dist	24.352	62.453	0.502	0.039	0.235	**0.273**	0.057	0.151	0.153	0.841
	Ours	**33.786**	**71.732**	0.526	**0.036**	**0.172**	0.284	**0.035**	0.189	0.122	**0.624**

TABLE III: Ablation study with task specific metrics

Removed	SSDC	Attn	VC	Dist	-
Acc_{TL}	0.9838	0.8734	0.880	0.9921	**0.9930**
MAE_{SP}	0.2667	0.2779	0.3305	0.2472	**0.2410**
BCE_{SEG}	0.1909	0.1921	**0.1827**	0.1831	0.1850
MAE_{WP}	0.0705	0.0736	0.0692	0.0682	**0.0614**
MAE_{ST}	0.0198	0.0186	0.0163	0.0164	**0.0153**
MAE_{TH}	0.0434	0.0434	0.0427	**0.0323**	0.0335
MAE_{BR}	0.0528	**0.0209**	0.0217	0.0239	0.0248
$Epoch^{*}$	50	43	49	35	57

to avoid collisions with various vehicles, such as a car and a high-ceiling van. Without depth information, if the model relied solely on image data to stop the vehicle, it would stop only when a specific portion of the front car became visible in the image. This approach could either cause an accident with the front car or result in maintaining a long distance from the van. Fig 4b also shows the performance of the fusion module to detect the red light, but gives priority to SDC map information not to stop too soon. Therefore, it will reduce the speed to stop at a safe distance to the front vehicle. Fig. 4c shows the performance of our model in detecting visual information, i.e. light's color and the front vehicle, as the human and the bicycle are barely visible in the SDC map and the model needs to rely on visual data (RGB features) to navigate.

E. Ablation Studies

We conducted four ablation studies to evaluate the effectiveness of different modules. This includes removing the side SDC (no SSDC), the attention block from the fusion module (no Attn), the vehicular control GRU loop (no VC), and the single-task distillation network (no Dist). To examine the impact of the changes suggested by our method, we conducted experiments shown in Table II, in Town5 long routes, adversarial scenarios, and averaged over hard weather condition (Hard Rain Sunset, Wet Noon and Cloudy Sunset). This helps to demonstrate the improvement caused by our proposed components to handle the challenging conditions.

The ablation study revealed that removing the attention block and vehicular control GRU loop will increase the collisions and violations caused by the agent. Although introducing single-task teachers may increase violations in some cases, removing them will deteriorate the route completion and driving score. Also, removing the side views from the SDC map decreased the RC and IP which is caused by more accidents with side vehicles due to lack of depth data from the sides. To better investigate this, we present the IL validation results of our method on different task-specific metrics in Table III. The metrics are binary cross-entropy BCE_{SEG} for SS and accuracy Acc_{TL} for traffic light state, similar to Natan et al. [8]. Mean absolute error (MAE) is used to assess the model's performance in predicting ego vehicle speed (MAE_{SP}), waypoints (MAE_{WP}), steering (MAE_{ST}), throttle (MAE_{TH}), and brake (MAE_{BR}) similar to their loss formula. All cases were trained for 60 epochs to ensure a fair comparison. We also present the epoch number of best validation ($epoch^{*}$) for the DMFuser and its variations.

According to Table III, using distillation, the agent makes a trade-off by sacrificing throttle and brake control in favor of improving steering and waypoint predictions. This trade-off can be attributed to the fact that the waypoint control modules are situated deeper within the network architecture, and steering is very sensitive to waypoint following. In MT learning, these deeper modules may receive less priority in comparison to features closer to the network's output layers. However, with the application of our proposed AFM, we are able to accentuate the significance of these internal features and intermediate tasks, thereby enhancing overall performance. This further ensures that the shallower features do not dominate the learning process. LiDAR sensors provide data on the height dimension, whereas SDC offers semantic information pertaining to each class on every layer, thus facilitating the acquisition of valuable insights by the model.

Combining attention and CNNs to fuse RGB and SDC features in the fusion block, allows learning the relationship between them, thus preventing information loss which results in great RC improvement in Table II. Incorporating traffic light and ego vehicle speed data enhances the RGB feature extraction process which also results in better scene understanding. Acc_{TL} and MAE_{SP} losses in Table III, show improved scene understanding of our method. This table also emphasizes the importance of the GRU in the navigational branch, predicting the required adjustments to vehicular control obtained from the waypoint branch by incorporating a dynamic coarse simulator. Removing it will reduce both waypoint and navigational commands performance.

V. Conclusion

This study introduces an end-to-end MT learning model that can concurrently manage perception and control tasks for AD vehicle. The model is designed to address a point-to-point navigation task in which the vehicle is required to follow a predetermined waypoint sequence established by a global planner. The CARLA simulator, is rendered to evaluate the model and examine various driving aspects. According to the ablation study, using the distillation module improves the prediction of waypoints and steering by mitigating misleading multi-task signals from other tasks. Enforcing AFM to adhere to single-task teachers prevents the loss of

crucial knowledge in the bottleneck nodes. This underscores the significance of fusing multiple sensors and utilizing attention and CNN modules rather than simple concatenation. For future direction, the integration of RL algorithms and explainable AI could be considered to enhance decision-making and interpretability of autonomous systems.

REFERENCES

[1] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[2] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger, "Label efficient visual abstractions for autonomous driving," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2338–2345.

[3] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" in *International Conference on Machine Learning*. PMLR, 2020, pp. 3145–3153.

[4] 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.

[5] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722–739, 2021.

[6] P. Agand, M. Taherahmadi, A. Lim, and M. Chen, "Human navigational intent inference with probabilistic and optimal approaches," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8562–8568.

[7] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2020.

[8] O. Natan and J. Miura, "End-to-end autonomous driving with semantic depth cloud mapping and multi-agent," *IEEE Transactions on Intelligent Vehicles*, 2022.

[9] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[10] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," *arXiv preprint arXiv:2206.08129*, 2022.

[11] P. Agand, M. Mahdavian, M. Savva, and M. Chen, "Letfuser: Lightweight end-to-end transformer-based sensor fusion for autonomous driving with multi-task learning," *arXiv preprint arXiv:2310.13135*, 2023.

[12] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[13] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.

[14] O. Natan and J. Miura, "Towards compact autonomous driving perception with balanced learning and multi-sensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 249–16 266, 2022.

[15] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?" *Science Robotics*, vol. 4, no. 30, p. eaaw6661, 2019.

[16] S. Fadadu, S. Pandey, D. Hegde, Y. Shi, F.-C. Chou, N. Djuric, and C. Vallespi-Gonzalez, "Multi-view fusion of sensor data for improved perception and prediction in autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2349–2357.

[17] N. Djuric, H. Cui, Z. Su, S. Wu, H. Wang, F.-C. Chou, L. San Martin, S. Feng, R. Hu, Y. Xu *et al.*, "Multixnet: Multiclass multistage multimodal motion prediction," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 435–442.

[18] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 641–656.

[19] P. Agand, M. Chang, and M. Chen, "Dmode: differential monocular object distance estimation module without class specific information," in *2024 13th International Workshop on Robot Motion and Control (RoMoCo)*. IEEE, 2024, pp. 261–266.

[20] 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.

[21] 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.

[22] C. Guo, T. Owaki, K. Kidono, T. Machida, R. Terashima, and Y. Kojima, "Toward human-like lane following behavior in urban environment with a learning-based behavior-induction potential map," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1409–1416.

[23] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[24] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 793–15 803.

[25] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[26] Y. Fan, P. Agand, M. Chen, E. J. Park, A. Kennedy, and C. Bae, "Sequential modeling of complex marine navigation: Case study on a passenger vessel (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 484–23 485.

[27] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[28] B. Lin, F. Ye, Y. Zhang, and I. W. Tsang, "Reasonable effectiveness of random weighting: A litmus test for multi-task learning," *arXiv preprint arXiv:2111.10603*, 2021.

[29] J. Ma and Q. Mei, "Graph representation learning via multi-task knowledge distillation," *arXiv preprint arXiv:1911.05700*, 2019.

[30] G. M. Jacob, V. Agarwal, and B. Stenger, "Online knowledge distillation for multi-task learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2359–2368.

[31] D. Wu, M.-W. Liao, W.-T. Zhang, X.-G. Wang, X. Bai, W.-Q. Cheng, and W.-Y. Liu, "Yolop: You only look once for panoptic driving perception," *Machine Intelligence Research*, pp. 1–13, 2022.

[32] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao *et al.*, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Springer, 2022, pp. 550–567.

[33] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

[34] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 723–13 733.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[36] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8728–8740, 2020.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[38] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.