Lightweight Temporal Transformer Decomposition for Federated Autonomous Driving

Tuong Do[1,2,3], Binh X. Nguyen[1], Quang D. Tran[1,3], Erman Tjiputra[1], Te-Chuan Chiu[2], Anh Nguyen[3]

***Abstract*— Traditional vision-based autonomous driving systems often face difficulties in navigating complex environments when relying solely on single-image inputs. To overcome this limitation, incorporating temporal data such as past image frames or steering sequences, has proven effective in enhancing robustness and adaptability in challenging scenarios. While previous high-performance methods exist, they often rely on resource-intensive fusion networks, making them impractical for training and unsuitable for federated learning. To address these challenges, we propose lightweight temporal transformer decomposition, a method that processes sequential image frames and temporal steering data by breaking down large attention maps into smaller matrices. This approach reduces model complexity, enabling efficient weight updates for convergence and real-time predictions while leveraging temporal information to enhance autonomous driving performance. Intensive experiments on three datasets demonstrate that our method outperforms recent approaches by a clear margin while achieving real-time performance. Additionally, real robot experiments further confirm the effectiveness of our method. Our source code can be found at: https://github.com/aioz-ai/LTFed.**

I. Introduction

Autonomous driving has the potential to revolutionize transportation by significantly improving safety, efficiency, and convenience [1], [2] for human drivers. Central to the effectiveness of autonomous vehicles is their ability to process and interpret visual data to make accurate driving decisions. However, traditional vision-based autonomous driving systems face privacy concerns, as they require collecting data from multiple users to train the model [3]. Furthermore, while recent studies have introduced various methods for autonomous driving, many of them predict trajectory information from a single image input [2]. This limitation reduces the system's ability to respond quickly and safely while maintaining the privacy of the users' data [4].

To overcome the limitation when using a single frame as input for the network, several works have included a sequence of frames to predict directly the steering control signal [5], [6]. This approach enables the system to anticipate potential hazards and take preventive measures, such as adjusting speed or changing lanes, to avoid close encounters. Despite the potential benefits, integrating temporal information into autonomous driving systems presents several challenges. In particular, the recent model complexity may necessitate substantial data for training, impede integration on devices with limited computational power, and pose significant challenges in ensuring real-time responses [7].

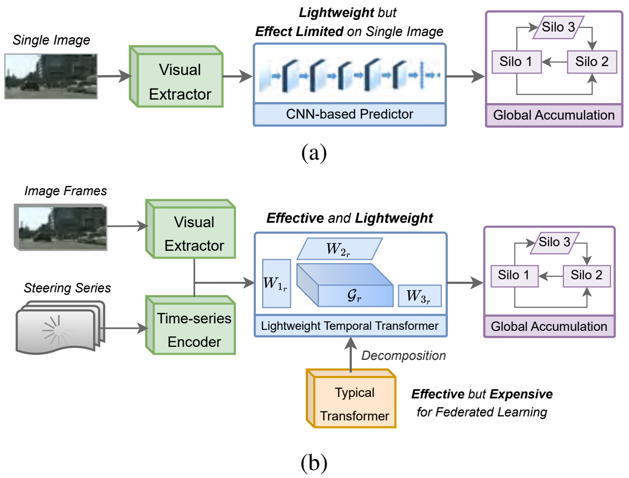

Fig. 1. Comparison between traditional approach for federated autonomous driving (a) and our lightweight temporal transformer network (b).

To address data privacy, several autonomous driving approaches utilize federated learning to train decentralized models across multiple vehicles [8]–[10]. However, most autonomous driving models still rely on the single frame as input and develop a relatively simple network to enable feasible a training in federated learning setup [11]–[13]. This single-frame approach overlooks the temporal data that each vehicle collects over time, which can provide essential context for understanding motion patterns, tracking objects, and anticipating potential hazards. As a result, these models do not fully leverage the sequence of information needed to better predict and respond to dynamic driving scenarios, ultimately limiting their performance and adaptability.

In this paper, we aim to develop a federated autonomous driving framework that takes into account the temporal information. To address the complexity challenges of the fusion model when training in a federated scenario using temporal information, we propose a Lightweight Temporal Transformer. This new approach reduces the complexity of the network in each silo by efficiently approximating the information from the inputs. Our method utilizes a decomposition method under unitary attention to break down learnable attention maps into low-rank ones, ensuring that the resulting models remain lightweight and trainable. By reducing model complexity, our approach enables the network to use temporal data while ensuring convergence. Intensive experiments show that our approach outperforms recent methods in federated autonomous driving.

[1] AIOZ, Singapore tuong.khanh-long.do@aioz.io
[2] Department of Computer Science, NTHU, Taiwan
[3] Department of Computer Science, University of Liverpool, UK

II. Related works

Autonomous Driving. Autonomous driving is a rapidly advancing field that has garnered significant research interest in recent years. Various studies have explored the application of deep learning for critical tasks, including object detection and tracking [14], [15], trajectory prediction [16], and autonomous braking and steering control [17]. For example, Xin *et al.* [18] proposed a recursive backstepping steering controller that connects yaw-rate-based path-following commands to steering adjustments, while Xiong *et al.* [19] analyzed nonlinear dynamics using proportional control methods. Yi *et al.* [20] introduced an algorithm that determines the instantaneous center of rotation within a self-reconfigurable robot's area, enabling waypoint navigation while avoiding collisions. Additionally, Yin *et al.* [21] combined model predictive control with covariance steering theory to create a robust controller for nonlinear autonomous driving systems. Moreover, recent works have leveraged temporal information to address complex environments and dynamic scene changes, demonstrating improved robustness and adaptability in challenging scenarios [22]. However, despite these advances [23], managing model complexity to enable deployment on low-level devices while maintaining effective performance remains a significant challenge.

Federated Learning. Federated learning (FL) supports decentralized training of machine learning models across multiple devices while keeping data localized, thereby preserving privacy and reducing data transfer [24]. In autonomous driving, FL enables vehicles to collaboratively learn from diverse datasets without sharing raw data [12]. Previous research has explored the use of FL in autonomous driving [9], [10], [25]. Recently, Zhao *et al.* [26] developed a federated learning framework for vehicle-to-vehicle communication that enhances model robustness and generalization. Some recent works also consider temporal information to improve performance in complex environments [27]–[30]. Other works have explored clustering-based solutions for post-processing [24], [31], learning feasibility through the modifying of the accumulator [32], topology design [33], or global architecture [9], [10]. However, fully exploiting temporal information within the constraints of federated learning remains a significant challenge due to computational complexity and limited device resources [34].

Lightweight Models. The tensor decomposition techniques aim at breaking down complex interactions into simpler components [35], thus, reducing the computational burden and improving the interpretability of models [36]. This technique shows promise in various applications, including image recognition [37]–[39] and natural language processing [40], [41], but its potential in federated learning for autonomous driving remains largely untapped. Compared with distillation [42]–[44], pruning [45]–[47], or quantization [48], [49] that require complex training setups, decomposing the network tensor can be trained directly with less parameters without the need to modify training paradigm, which shows potentials in federated training for autonomous vehicles when handling high dimensional data inputs.

III. Methodology

A. Preliminary

We consider a federated network with N autonomous vehicles, collaboratively training a global driving policy θ by aggregating local weights θ_i from each silo i, where $i \in [1, N]$. Each silo minimizes a regression loss $\mathcal{L}$ computed using a deep network that predicts the steering angle from temporally ordered RGB images and steering series.

Local Regression Objective. We use mean squared error (MSE) as the objective function for predicting the steering angle in each local silo. Here, we only use the extracted joint features z of the local model to predict steering angles.

$$\mathcal{L} = \text{MSE}(\theta_i, \xi_i^b) \tag{1}$$

where b is the mini-batch size; ξ_i^b is the ground-truth steering angle of batch b from silo i.

Local Optimization. To ensure the model convergence under a federated training scenario, for each k communication round, we use decentralized periodic averaging stochastic gradient descent (DPASGD) [50].

$$\theta_i\left(k+1\right) = \begin{cases} \sum_{j\in\mathcal{N}_i^+\cup\{i\}} \mathbf{A}_{i,j}\theta_j\left(k\right), \\ \quad \text{if k} \equiv 0 \left(\text{mod } u+1\right) \,\&\, \left|\mathcal{N}_i^+\right| > 1, \\ \theta_i\left(k\right) - \alpha_k \frac{1}{b}\sum_{h=1}^{b} \nabla\mathcal{L}_i\left(\theta_i\left(k\right), \xi_i^b\left(k\right)\right), \\ \quad \text{otherwise.} \end{cases} \tag{2}$$

where $\mathcal{N}_i^+$ is the in-neighbors set of silo i; u is the number of local updates; $\mathbf{A}$ is a consensus matrix for parameter accumulating.

Global Accumulation. Since our method focuses on the practical application of a network under a federated learning scenario, we use the simple accumulation solution FedAvg [51] for computing the global model θ. The federated averaging process is conducted as follows:

$$\theta = \frac{1}{\sum_{i=0}^{N}\lambda_i}\sum_{i=0}^{N}\lambda_i\theta_i, \tag{3}$$

where $\lambda_i = \{0, 1\}$. Note that $\lambda_i = 1$ indicates that silo i joins the inference process and $\lambda_i = 0$ if not.

Feature Extraction. We use a standard vision transformer to extract the feature from the sequence of temporal inputs. This representation, $z \in \mathbb{R}^{d_z}$, is computed as:

$$z = \Big\langle \mathcal{T},\ \text{vec}(M_1) \circ \text{vec}(M_2) \circ \text{vec}(M_3) \Big\rangle, \tag{4}$$

where $\circ$ is the outer product; $\langle ., . \rangle$ is the inner product; $\mathcal{T} \in \mathbb{R}^{d_{M_1}\times d_{M_2}\times d_{M_3}\times d_z}$ is a learnable tensor; $M_l \in \mathbb{R}^{n_l \times d_l}$ is the modality input with n_l elements, each represented by d_l-dimension features; $d_{M_l} = n_l \times d_l$; and $vec(M_l)$ vectorizes M_l into a row vector. M_1, M_2, M_3 represent past frames, steering series, and the current RGB image, respectively. While $\mathcal{T}$ captures input interactions, learning such a large tensor is impractical with high-dimensional inputs d_{M_l}, straining vehicle computing resources and hindering model convergence due to large linear parameter correlations. Thus,

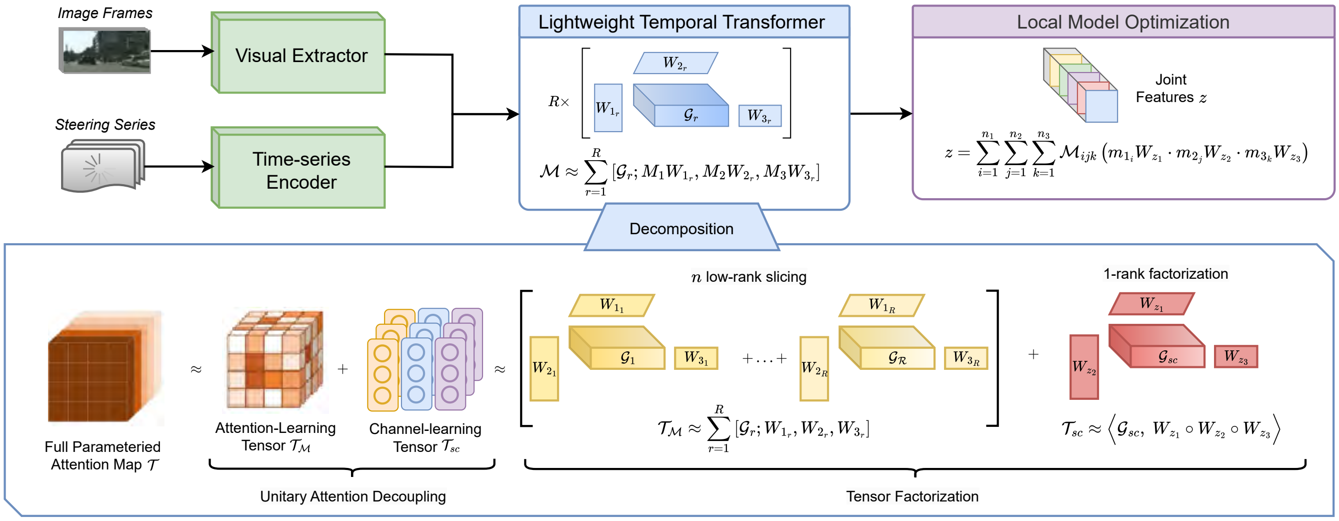
Image Frames
Visual Extractor
Steering Series
Time-series Encoder
Lightweight Temporal Transformer
$R\times$
W_{1_r}
W_{2_r}
$\mathcal{G}_r$
W_{3_r}
$\mathcal{M} \approx \sum_{r=1}^{R} [\mathcal{G}_r; M_1W_{1_r}, M_2W_{2_r}, M_3W_{3_r}]$
Local Model Optimization
Joint Features z
$z = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \mathcal{M}_{ijk}\left(m_{1_i}W_{z_1} \cdot m_{2_j}W_{z_2} \cdot m_{3_k}W_{z_3}\right)$
Decomposition
Full Parameteried Attention Map $\mathcal{T}$
$\approx$
Attention-Learning Tensor $\mathcal{T}_{\mathcal{M}}$
$+$
Channel-learning Tensor $\mathcal{T}_{sc}$
$\approx$
n low-rank slicing
W_{1_1}
W_{2_1}
$\mathcal{G}_1$
W_{3_1}
$+\ldots+$
$W_{1_\mathcal{R}}$
$W_{2_\mathcal{R}}$
$\mathcal{G}_\mathcal{R}$
$W_{3_\mathcal{R}}$
$\mathcal{T}_{\mathcal{M}} \approx \sum_{r=1}^{R} [\mathcal{G}_r; W_{1_r}, W_{2_r}, W_{3_r}]$
$+$
1-rank factorization
W_{z_1}
W_{z_2}
$\mathcal{G}_{sc}$
W_{z_3}
$\mathcal{T}_{sc} \approx \left\langle \mathcal{G}_{sc},\ W_{z_1} \circ W_{z_2} \circ W_{z_3} \right\rangle$
Unitary Attention Decoupling
Tensor Factorization

Fig. 2. An overview of our lightweight temporal transformer decomposition method for federated autonomous driving.

we aim to reduce the size of $\mathcal{T}$ by minimizing unnecessary linear combinations through the introduced Lightweight Temporal Transformer Decomposition. Specifically, we use *unitary attention decoupling* to approximate large tensor $\mathcal{T}$ into smaller ones, followed by a *tensor factorization* to factorize tensors into factor matrices.

B. Unitary Attention Decoupling

Inspired by [52], we rely on the idea of *unitary attention* mechanism to reduce the size of $\mathcal{T}$. Specifically, let $z_p \in \mathbb{R}^{d_z}$ be the joint representation of p^{th} triplet of channels where each channel in the triplet is from a different input. The representation of each channel in a triplet is $m_{1_i}, m_{2_j}, m_{3_k}$, where $i \in [1, n_1], j \in [1, n_2], k \in [1, n_3]$, respectively. There are $n_1 \times n_2 \times n_3$ possible triplets over the three inputs. The joint representation z_p resulted from a fully parameterized trilinear interaction over three channel representations $m_{1_i}, m_{2_j}, m_{3_k}$ of p^{th} triplet is computed as

$$z_p = \left\langle \mathcal{T}_{sc},\ m_{1_i} \circ m_{2_j} \circ m_{3_k} \right\rangle, \tag{5}$$

where $\mathcal{T}_{sc} \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_z}$ is the learning tensor between channels in the triplet.

Following [52], the joint representation z is approximated by using joint representations of all triplets described in (5) instead of using fully parameterized interaction over three inputs. Hence, we compute

$$z = \sum_p \mathcal{M}_p z_p, \tag{6}$$

Note that in (6), we compute a weighted sum over all possible triplets. The p^{th} triplet is associated with a scalar weight $\mathcal{M}_p$. The set of $\mathcal{M}_p$ is called as the attention map $\mathcal{M}$, where $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

The attention map $\mathcal{M}$ resulted from a reduced parameterized trilinear interaction over three inputs M_1, M_2 and M_3 is computed as follows

$$\mathcal{M} = \left\langle \mathcal{T}_{\mathcal{M}},\ M_1 \circ M_2 \circ M_3 \right\rangle, \tag{7}$$

where $\mathcal{T}_{\mathcal{M}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is the learning tensor of attention map $\mathcal{M}$. Note that the learning tensor $\mathcal{T}_{\mathcal{M}}$ in (7) has a reduced size compared to the learning tensor $\mathcal{T}$.

By integrating (5) into (6), the joint representation z in (6) can be rewritten as

$$z = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \mathcal{M}_{ijk} \langle \mathcal{T}_{sc},\ m_{1_i} \circ m_{2_j} \circ m_{3_k} \rangle, \tag{8}$$

where $\mathcal{M}_{ijk}$ in (8) is actually a scalar attention weight $\mathcal{M}_p$ of the attention map $\mathcal{M}$ in (7).

It is also worth noting from (8) that to compute z, instead of learning the large tensor $\mathcal{T} \in \mathbb{R}^{d_{M_1} \times d_{M_2} \times d_{M_3} \times d_z}$, we now only need to learn two smaller tensors $\mathcal{T}_{sc} \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_z}$ in (5) and $\mathcal{T}_{\mathcal{M}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ in (7).

C. Tensor Factorization

Although the large tensor $\mathcal{T}$ is replaced by two smaller tensors $\mathcal{T}_{\mathcal{M}}$ and $\mathcal{T}_{sc}$, there are too many linear fusions between mentioned tensors which still affect the learning of the global model. Therefore, we apply the factorization as in [53] to $\mathcal{T}_{\mathcal{M}}$ and $\mathcal{T}_{sc}$ into learnable factor matrices.

The factorization for the learning tensor $\mathcal{T}_{\mathcal{M}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ can be calculated as

$$\mathcal{T}_{\mathcal{M}} \approx \sum_{r=1}^{\mathcal{R}} \left\langle \mathcal{G}_r,\ W_{1_r} \circ W_{2_r} \circ W_{3_r} \right\rangle, \tag{9}$$

where $\mathcal{G}_r \in \mathbb{R}^{d_{1_r} \times d_{2_r} \times d_{3_r}}$ are compact learnable Tucker tensors [54], small-sized tensors that support minimizing error when approximating a larger tensor using its factorized matrices; $\mathcal{R}$ is a slicing parameter, establishing a trade-off between the decomposition rate (which is directly related to the usage memory and the computational cost) and the performance. The maximum value for $\mathcal{R}$ is usually set to the greatest common divisor of d_1, d_2 and d_3. In our experiments, we found that $\mathcal{R} = 32$ gives a good trade-off between the decomposition rate and the performance.

Here, we have dimension $d_{1_r} = d_1/\mathcal{R}$, $d_{2_r} = d_2/\mathcal{R}$ and $d_{3_r} = d_3/\mathcal{R}$; $W_{1_r} \in \mathbb{R}^{d_1 \times d_{1_r}}$, $W_{2_r} \in \mathbb{R}^{d_2 \times d_{2_r}}$ and $W_{3_r} \in \mathbb{R}^{d_3 \times d_{3_r}}$ are learnable factor matrices. Fig. 2 shows the illustration of factorization for a tensor $\mathcal{T}_{\mathcal{M}}$.

The shortened form of $\mathcal{T}_{\mathcal{M}}$ in (9) can be rewritten as

$$\mathcal{T}_{\mathcal{M}} \approx \sum_{r=1}^{\mathcal{R}} [\![\mathcal{G}_r; W_{1_r}, W_{2_r}, W_{3_r}]\!], \tag{10}$$

Integrating the learning tensor $\mathcal{T}_{\mathcal{M}}$ from (10) into (7), the attention map $\mathcal{M}$ can be rewritten as

$$\mathcal{M} = \sum_{r=1}^{\mathcal{R}} [\![\mathcal{G}_r; M_1 W_{1_r}, M_2 W_{2_r}, M_3 W_{3_r}]\!], \tag{11}$$

Similar to $\mathcal{T}_{\mathcal{M}}$, we apply to $\mathcal{T}_{sc}$ in (8) to reduce the complexity. Note that the size of $\mathcal{T}_{sc}$ directly affects the dimension of the joint representation $z \in \mathbb{R}^{d_z}$. Hence, to minimize the loss of information, we set the slicing parameter $\mathcal{R} = 1$ and the projection dimension of factor matrices at d_z, i.e., the same dimension of the joint representation z.

Therefore, $\mathcal{T}_{sc} \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_z}$ in (8) is calculated as

$$\mathcal{T}_{sc} \approx \left\langle \mathcal{G}_{sc},\ W_{z_1} \circ W_{z_2} \circ W_{z_3} \right\rangle, \tag{12}$$

where $W_{z_1} \in \mathbb{R}^{d_1 \times d_z}$, $W_{z_2} \in \mathbb{R}^{d_2 \times d_z}$, $W_{z_3} \in \mathbb{R}^{d_3 \times d_z}$ are learnable factor matrices and $\mathcal{G}_{sc} \in \mathbb{R}^{d_z \times d_z \times d_z \times d_z}$ is a smaller tensor (compared to $\mathcal{T}_{sc}$).

Up to now, we already have $\mathcal{M}$ by (11) and $\mathcal{T}_{sc}$ by (12). Hence, we can compute z using (8) as

$$\begin{aligned} z = &\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \\ &\mathcal{M}_{ijk} \langle \mathcal{G}_{sc},\ (m_{1_i} W_{z_1}) \circ (m_{2_j} W_{z_2}) \circ (m_{3_k} W_{z_3}) \rangle, \end{aligned} \tag{13}$$

Here, it is interesting to note that $\mathcal{G}_{sc} \in \mathbb{R}^{d_z \times d_z \times d_z \times d_z}$ in (13) has rank 1. Thus, the result obtained from (13) can be approximated by the Hadamard products without the presence of rank-1 tensor $\mathcal{G}_{sc}$ [35]. In particular, z in (13) can be computed without using $\mathcal{G}_{sc}$ as

$$z = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{M}_{ijk} \left(m_{1_i} W_{z_1} \cdot m_{2_j} W_{z_2} \cdot m_{3_k} W_{z_3} \right), \tag{14}$$

The joint embedding dimension d_z is a user-defined parameter that makes a trade-off between the capability of the representation and the computational cost. In our experiments, $d_z = 1,024$ gives a good trade-off.

D. Convergence Analysis

Proposition 1. *Lightweight Temporal Transformer Decomposition in Equation 4 can be considered a form of Bilinear Attention [55], naturally inheriting its convergence ability.*

Proof. Let the input contain two representations of two modalities, i.e., $M_1^b \in \mathbb{R}^{n_1^b \times d_1^b}$ and $M_2^b \in \mathbb{R}^{n_2^b \times d_2^b}$, where n_1^b and n_2^b are number of channels; d_1^b and d_2^b are the representation dimension of each corresponding channel. Following Equation 4, the joint representation $z \in \mathbb{R}^{d_z}$ can now be described as

$$z = \left\langle \mathcal{T}_b,\ \mathrm{vec}(M_1^b) \circ \mathrm{vec}(M_2^b) \right\rangle, \tag{15}$$

where $\mathcal{T}_b \in \mathbb{R}^{d_{n1}^b \times d_{n2}^b \times d_z}$ is learnable tensor; $d_{n1}^b = n_1^b \times d_1^b$; $d_{n2}^b = n_2^b \times d_2^b$. By applying parameter factorization (Sec. III-C), z in (15) can be approximated based on (14) as

$$z = \sum_{i=1}^{n_1^b} \sum_{j=1}^{n_2^b} \mathcal{M}_{ij} \left({M_{1_i}^b}^T W_{z_{b1}} \cdot {M_{2_j}^b}^T W_{z_{b2}} \right), \tag{16}$$

where $W_{z_{b1}} \in \mathbb{R}^{d_1^b \times d_z}$ and $W_{z_{b2}} \in \mathbb{R}^{d_2^b \times d_z}$ are learnable factor matrices; $\mathcal{M}_{ij}$ is an attention weight of attention map $\mathcal{M} \in \mathbb{R}^{n_1^b \times n_2^b}$ which can be computed from (11) as

$$\mathcal{M} = \sum_{r=1}^{\mathcal{R}} [\![\mathcal{G}_r; {M_1^b}^T W_{1_r}, {M_2^b}^T W_{2_r}]\!], \tag{17}$$

where $W_{1_r} \in \mathbb{R}^{d_1^b \times d_{1_r}^b}$ and $W_{2_r} \in \mathbb{R}^{d_2^b \times d_{2_r}^b}$ are learnable factor matrices; $d_{1_r}^b = d_1^b/\mathcal{R}$; $d_{2_r}^b = d_2^b/\mathcal{R}$; each $\mathcal{G}_r \in \mathbb{R}^{d_{1_r}^b \times d_{2_r}^b}$ is a learnable Tucker tensor. By extracting k-element and reorganize the multiplication computations over tensors, (16) can be rewritten as

$$z_k = \sum_{i=1}^{n_1^b} \sum_{j=1}^{n_2^b} \mathcal{M}_{ij} \left({M_{1_i}^b}^T \left(W_{z_{1\,k}} W_{z_{2\,k}}^T \right) M_{2_j}^b \right), \tag{18}$$

where z_k is k^{th} element of the joint representation z; $W_{z_{1\,k}}$ and $W_{z_{2\,k}}$ are k^{th} column in factor matrices W_{z_1} and W_{z_2}.

Interestingly, from (18), we can rewrite it as a computational form of a Bilinear Attention as below:

$$\begin{aligned} z_k &= \sum_{i=1}^{n_1^b} \sum_{j=1}^{n_2^b} \mathcal{M}_{ij} \left({M_{1_i}^b}^T \left(W_{z_{1\,k}} W_{z_{2\,k}}^T \right) M_{2_j}^b \right) \\ &= \sum_{i=1}^{n_1^b} \sum_{j=1}^{n_2^b} \mathcal{M}_{ij} \left({M_{1_i}^b}^T W_{z_{1\,k}} \right) \left(W_{z_{2\,k}}^T M_{2_j}^b \right) \\ &= ({M_1^b}^T W_{z_1})_k^T \mathcal{M} ({M_2^b}^T W_{z_2})_k. \end{aligned} \tag{19}$$

□

Remark 1. *Proposition 1 suggests that the results of our decomposition method can be considered as a Bilinear Attention, which inherits its convergence ability and ensures the network will converge during the training.*

Method	**Main Focus**	**Inputs**	**Learning Scenario**	**RMSE**			**MAE**			**#Params (M)**	**Avg. Cycle Time (ms)**
				Udacity+	***Gazebo***	***Carla***	***Udacity+***	***Gazebo***	***Carla***		
MobileNet [56]	Archtecture	Realtime Vision	CLL	0.193	0.083	0.286	0.176	0.057	0.200	2.22	_
DroNet [57]				0.183	0.082	0.333	0.15	0.053	0.218	0.31	_
St-p3 [6]		Temporal		0.092	0.071	0.132	0.090	0.049	0.132	1247.87	_
ADD [4]				0.097	0.049	0.166	0.092	0.042	0.121	3234.22	_
HPO [5]				0.088	0.044	0.157	0.070	0.044	0.105	5990.19	_
FedAvg [58]	Aggregation/ Optimization	Realtime Vision	SFL	0.212	0.094	0.269	0.185	0.064	0.222	0.31	152.4
FedProx [59]				0.152	0.077	0.226	0.118	0.063	0.151	0.31	111.5
STAR [60]				0.179	0.062	0.208	0.149	0.053	0.155	0.31	299.9
FedTSE [32]		Temporal		0.144	0.063	0.079	0.075	0.051	0.154	89.1	1172
TGCN [61]	Clustering			0.137	0.069	0.193	0.069	0.047	0.179	78.33	224
Fed-STGRU [24]				0.129	0.059	0.151	0.080	0.048	0.156	91.01	370
BFRT [10]	Archtecture			0.113	0.054	0.111	0.081	0.043	0.133	427.26	1256
MFL [9]				0.108	0.052	0.133	0.093	0.043	0.138	173.87	781
CDL [25]	Optimization	Realtime Vision	DFL	0.141	0.062	0.183	0.083	0.052	0.147	0.63	72.7
MATCHA [62]	Topology Design			0.182	0.069	0.208	0.148	0.058	0.215	0.31	171.3
MBST [63], [64]				0.183	0.072	0.214	0.149	0.058	0.206	0.31	82.1
FADNet [12]				0.162	0.069	0.203	0.134	0.055	0.197	0.32	62.6
PriRec [33]		Temporal		0.137	0.066	0.196	0.093	0.052	0.127	325.57	272
PEPPER [31]	Clustering			0.124	0.055	0.115	0.078	0.054	0.122	89.13	438
Ours	Compact Network	Temporal	CLL	**0.088**	0.045	0.091	0.078	0.039	0.114	5.01	_
			SFL	0.107	0.049	**0.072**	**0.069**	**0.035**	0.119	5.01	180
			DFL	0.091	**0.043**	0.076	0.076	0.038	**0.104**	5.01	121

TABLE I

Performance comparison between different methods. The Gaia topology is used.

IV. Experiment

A. Implementation Details

Dataset. Udacity+ [65], Gazebo Indoor [12], and Carla Outdoor dataset [12] are used as benchmarking datasets, which are similar to setups mentioned in [3], [12]. To provide temporal information, we further preprocess the training data by chunking videos into multiple consequences. Each consequence includes a current input image, 5 previous frames, and their corresponding 5 past steering angles.

Training. Each local model is trained with a dynamic batch size and an adaptive learning rate, utilizing the RMSprop [66] optimizer. The training process is executed in decentralized silos, where local updates are periodically transmitted and aggregated following Equation 3. The Early stopping criterion is applied to ensure convergence and the simulation setup follows the framework outlined in [12]. As in [64], our experiments explore three federated network topologies: Gaia [67], the NWS [68], and Exodus framework [67]. While we adopt the NWS topology in primary evaluations to reflect real-world cloud-based federated learning scenarios, Gaia and Exodus are analyzed in an ablation study to assess the impact of varying network structures on performance and convergence behavior.

Baselines. We evaluate our approach across various learning settings, including real-time vision-based and temporal-based methods. For Centralized Local Learning (CLL), we benchmark against MobileNet-V2 [56], Dronet [57], St-p3 [6], ADD [4], and HPO [5]. In the Server-based Federated Learning (SFL) setting, comparisons are made with FedAvg [58], FedProx [59], STAR [60], FedTSE [32], TGCN [61], Fed-STGRU [24], BFRT [10], and MFL [9]. For Decentralized Federated Learning (DFL), we assess performance against MATCHA [62], MBST [64], FADNet [12], PriRec [33], and PEPPER [31]. We assess model performance using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Additionally, we measure computational efficiency by recording the wall-clock time (ms) for training each method on an NVIDIA A100 GPU.

B. Main Results

Table I shows a comparison between our approach and state-of-the-art methods, both with and without temporal information. The results demonstrate a clear performance advantage, as our method achieves notably lower RMSE and MAE across all three datasets: Udacity+, Carla, and Gazebo. Besides, we also provide visualization of method comparisons in Fig. 3 and in our supplementary video, which emphasizes our approach's effectiveness in optimizing model complexity while maintaining model convergence and real-time performance, making it suitable for deployment in federated autonomous driving scenarios.

C. Ablation Study

Temporal Analysis. Table II shows the performance and the computing trade-off between our compact network and a full-parametrized network HPO [5] in handling different modalities for decentralized autonomous driving. The results show that using more temporal information provides higher change to increase model performance. However, they also

Method	Type	Inputs			RMSE			#Params (M)	Avg. Inference Time (ms)
		Current Image	*Previous Frame*	*Steering Series*	*Udacity+*	*Gazebo*	*Carla*		
HPO [5]	Full-Parametrized Network	✓	✓		0.78	0.23	0.26	207.74	426
		✓		✓	0.223	0.137	0.149	121.03	128
		✓	✓	✓	1.127	_	0.972	5,990.19	_
Ours	Compact Network	✓	✓		0.162	0.091	0.109	1.42	19
		✓		✓	0.144	0.092	0.092	0.97	17
		✓	✓	✓	**0.091**	**0.043**	**0.076**	5.01	22

TABLE II

PERFORMANCE OF METHODS UNDER MULTI-MODALITY INPUTS.

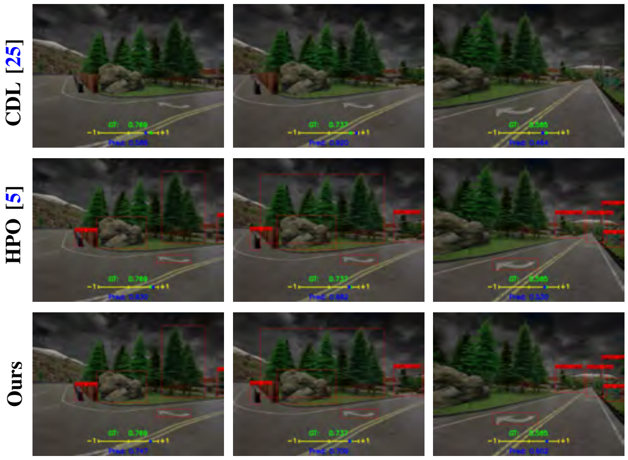

Fig. 3. Qualitative results between different methods.

burst complexity and cause divergence. These results also imply our method's effectiveness in handling complexity while fully leveraging temporal information to maximize performance. Besides, the compact network also ensures convergence and real-time computation.

Robustness Analysis. Training federated algorithms becomes increasingly challenging as the number of vehicle data silos grows. To evaluate the robustness of our approach, we test it alongside baseline methods across different topology sizes. Table III compares the performance of FADNet [12], CDL [25], and our method across three network topology infrastructures: Gaia [67] (11 silos), NWS [68] (22 silos), and Exodus [67] (79 silos). The results indicate that our approach consistently outperforms the baselines in all setups, demonstrating its scalability and effectiveness in large-scale vehicle networks. Moreover, the consistent performance of our approach across different environments, as shown in Fig. 4, further confirms its robustness and adaptability.

Decomposition Effectiveness. We further compute the decomposition rate of our Lightweight Temporal Transformer Decomposition. For a full interaction between the multi-modal inputs with the original vision transformer network, we would need to learn 5.9 billion parameters, which is infeasible in practice in the federated learning setup. By using the proposed decomposition method with the provided settings, i.e., the number of slicing $\mathcal{R} = 32$ and the dimension of the joint representation $d_z = 1024$, the number of parameters that need to be learned is only around 5 million. In other words, we achieve a decomposition rate of approximately 1179 times.

Topology	Architecture	Dataset		
		Udacity+	*Gazebo*	*Carla*
Gaia *(11 silos)*	FADNet	0.162(↓0.071)	0.069(↓0.026)	0.203 (↓0.127)
	CDL	0.141(↓0.050)	0.062(↓0.019)	0.183(↓0.107)
	Ours	**0.091**	**0.043**	**0.076**
NWS *(22 silos)*	FADNet	0.165(↓0.084)	0.07(↓0.017)	0.2(↓0.082)
	CDL	0.138(↓0.057)	0.058(↓0.005)	0.182(↓0.064)
	Ours	**0.081**	**0.053**	**0.118**
Exodus *(79 silos)*	FADNet	0.179(↓0.087)	0.081(↓0.026)	0.238(↓0.117)
	CDL	0.138(↓0.046)	0.061(↓0.006)	0.176(↓0.055)
	Ours	**0.092**	**0.055**	**0.121**

TABLE III

RESULTS OF OUR METHOD UNDER DIFFERENT TOPOLOGIES.

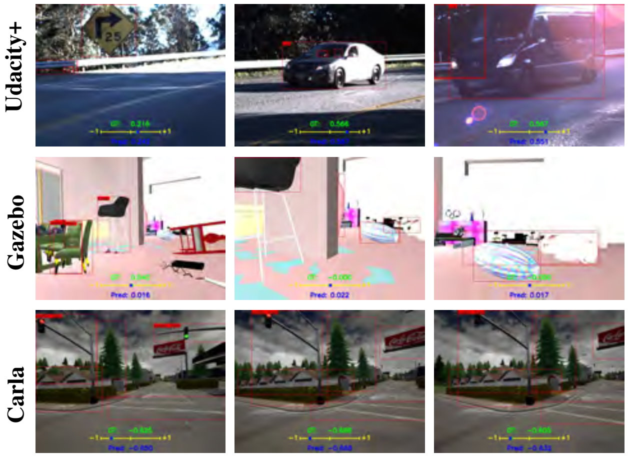

Fig. 4. Visualization results of our method on different environments.

D. Robotic Demonstration

We deploy the aggregated trained model on an autonomous mobile platform for real-world validation. The training process utilized the Udacity+ dataset using Gaia topology. The mobile robot is equipped with a 12-core ARM Cortex-A78AE 64-bit CPU and an NVIDIA Orin NX GPU, providing sufficient computational resources for edge-based inference. With an optimized inference time of *18 ms*, our approach enables low-latency, real-time steering angle predictions, crucial for responsive autonomous navigation (Fig. 5). Real-world visualizations are in our demo video.

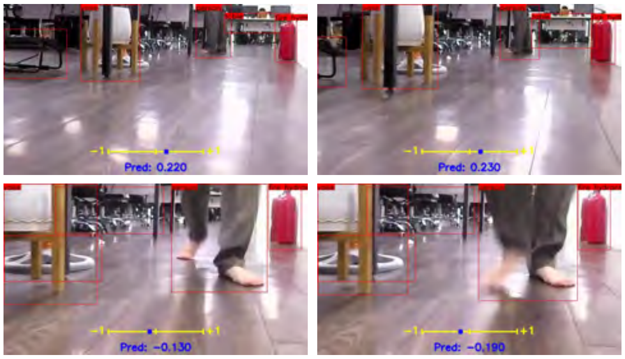

Fig. 5. Visualization results in real robot experiments. Our proposed model is lightweight and can be integrated into robot edge devices.

E. Limitation and Discussion

While our proposed lightweight temporal transformer decomposition demonstrates significant improvements in reducing parameter complexity and enhancing computational efficiency, certain limitations remain. Specifically, because our method approximates a large transformer into a smaller one with fewer parameters, as the temporal input length increases (e.g., past frames increasing from 5 to 30), the model may be insufficient to learn useful information from the input, potentially leading to degraded performance. Additionally, the choice of the slicing parameter $\mathcal{R}$ and the embedding dimension d_z impacts the trade-off between accuracy and efficiency. Moreover, the reliance on the rank-1 tensor approximation can lead to a loss of expressiveness, preventing architecture-based solutions from effectively addressing non-independent and identically distributed issues. While our experiments show promising results, addressing these limitations in future work could involve exploring adaptive mechanisms to dynamically adjust the tensor decomposition parameters or integrating regularization techniques to mitigate approximation errors and enhance model stability.

V. Conclusion

We propose temporal transformer decomposition, a new method designed to efficiently learn image frames and temporal steering series in a federated autonomous driving context. By leveraging unitary attention decoupling and tensor factorization, we decompose learnable attention maps into small-sized learnable matrices, maintaining an efficient model suitable for real-time predictions while preserving critical temporal information to enhance autonomous driving performance. Extensive evaluations conducted across three datasets demonstrate the effectiveness of our approach, validating its potential for practical deployment. In the future, we intend to validate our approach with a broader range of data sources and deploy trained models in more real-world scenarios using autonomous vehicles on public roads.

References

[1] U. M. Gidado, H. Chiroma, N. Aljojo, *et al.*, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, 2020.

[2] A. Nguyen and Q. D. Tran, "Autonomous navigation with mobile robots using deep learning and the robot operating system," in *IROS*, 2021.

[3] D. C. Nguyen, M. Ding, Q.-V. Pham, *et al.*, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IoT-J*, 2021.

[4] X. Zhao, M. othersQi, Z. Liu, S. Fan, C. Li, and M. Dong, "End-to-end autonomous driving decision model joined by attention mechanism and spatiotemporal features," *IET Intelligent Transport Systems*, 2021.

[5] M. A.bou Hussein, S. H. othersMüller, and J. Boedecker, "Multimodal spatio-temporal information in end-to-end networks for automotive steering prediction," in *ICRA*. IEEE, 2019.

[6] S. Hu, L. Chen, P. othersWu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *ECCV*. Springer, 2022.

[7] J. Wang, Y. othersLi, Z. Zhou, C. Wang, Y. Hou, L. Zhang, X. Xue, M. Kamp, X. L. Zhang, and S. Chen, "When, where and how does it fail? a spatial-temporal visual analytics approach for interpretable object detection in autonomous driving," *TVCG*, 2022.

[8] L. Barbieri, S. Savazzi, M. Brambilla, *et al.*, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Vehicular Communications*, 2021.

[9] T. Zeng, J. othersGuo, K. J. Kim, K. Parsons, P. Orlik, S. Di Cairano, and W. Saad, "Multi-task federated learning for traffic prediction and its application to route planning," in *IV*. IEEE, 2021.

[10] C. Meese, H. Chen, S. A. othersAsif, W. Li, C.-C. Shen, and M. Nejad, "Bfrt: Blockchained federated learning for real-time traffic flow prediction," in *CCGrid*. IEEE, 2022.

[11] Y. Li, X. Tao, X. Zhang, *et al.*, "Privacy-preserved federated learning for autonomous driving," *T-ITS*, 2021.

[12] A. Nguyen, T. Do, M. Tran, *et al.*, "Deep federated learning for autonomous driving," in *IV*, 2022.

[13] X. Liang, Y. Liu, T. Chen, *et al.*, "Federated transfer reinforcement learning for autonomous driving," in *FTL*, 2022.

[14] H. Hu, Z. Liu, *et al.*, "Investigating the impact of multi-lidar placement on object detection for autonomous driving," in *CVPR*, 2022.

[15] Y.-N. Chen, H. Dai, and Y. Ding, "Pseudo-stereo for monocular 3d object detection in autonomous driving," in *CVPR*, 2022.

[16] J. Wang, T. Ye, Z. Gu, *et al.*, "Ltp: Lane-based trajectory prediction for autonomous driving," in *CVPR*, 2022.

[17] N. Ijaz and Y. Wang, "Automatic steering angle and direction prediction for autonomous driving using deep learning," in *ISCSIC*, 2021.

[18] M. Xin *et al.*, "Slip-based nonlinear recursive backstepping path following controller for autonomous ground vehicles," in *ICRA*, 2020.

[19] J. Xiong, B. Li, R. Yu, *et al.*, "Reduced dynamics and control for an autonomous bicycle," in *ICRA*, 2021.

[20] L. Yi, V. Le, *et al.*, "Anti-collision static rotation local planner for four independent steering drive self-reconfigurable robot," in *ICRA*, 2022.

[21] J. Yin *et al.*, "Trajectory distribution control for model predictive path integral control using covariance steering," in *ICRA*, 2022.

[22] H. Shao, L. Wang, R. othersChen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *CVPR*, 2023.

[23] J. Liu, J. othersYin, Z. Jiang, Q. Liang, and H. Li, "Attention-based distributional reinforcement learning for safe and efficient autonomous driving," *RA-LIEEE Robotics and Automation Letters*, 2024.

[24] G. Kaur, S. K. othersGrewal, and A. Jain, "Federated learning based spatio-temporal framework for real-time traffic prediction," *WPCireless Personal Communications*, 2024.

[25] T. Do, B. X. othersNguyen, Q. D. Tran, H. Nguyen, E. Tjiputra, T.-C. Chiu, and A. Nguyen, "Reducing non-iid effects in federated autonomous driving with contrastive divergence loss," in *ICRA*. IEEE, 2024.

[26] Y. Zhao, M. Li, L. Lai, *et al.*, "Federated learning with non-iid data," *arXiv*, 2018.

[27] Z. Zhang, H. othersWang, Z. Fan, J. Chen, X. Song, and R. Shibasaki, "Gof-tte: Generative online federated learning framework for travel time estimation," *IoT*, 2022.

[28] X. Zhou, R. Ke, Z. Cui, Q. Liu, and W. Qian, "Stfl: Spatio-temporal federated learning for vehicle trajectory prediction," in *DTPI*. IEEE, 2022.

[29] X. Shen, J. Chen, J. Yan, RanChen, and R. Yan, "A spatial–temporal model for network-wide flight delay prediction based on federated learning," *ASCpplied Soft Computing*, 2024.

[30] Q. Liu, S. Sun, M. othersLiu, Y. Wang, and B. Gao, "Online spatio-temporal correlation-based federated learning for traffic flow forecasting," *T-ITS*, 2024.

[31] Y. Belal, A. Bellet, S. B. Mokhtar, and V. Nitu, "Pepper: Empowering user-centric recommender systems over gossip learning," *ACMProceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2022.

[32] X. Yuan, J. othersChen, N. Zhang, C. Zhu, Q. Ye, and X. S. Shen, "Fedtse: Low-cost federated learning for privacy-preserved traffic state estimation in iov," in *INFOCOM*. IEEE, 2022.

[33] C. Chen, J. others Zhou, B. Wu, W. Fang, L. Wang, Y. Qi, and X. Zheng, "Practical privacy preserving poi recommendation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[34] V. P. Chellapandi, L. othersYuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IV*, 2023.

[35] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, 2009.

[36] N. D. Sidiropoulos, L. otherDe Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *Signal ProcessingIEEE Transactions on signal processing*, 2017.

[37] M. Zhang, Y. othersGao, C. Sun, and M. Blumenstein, "Robust tensor decomposition for image representation based on generalized correntropy," *TIP*, 2020.

[38] M. Yin, Y. othersSui, S. Liao, and B. Yuan, "Towards efficient tensor decomposition-based dnn model compression with optimization framework," in *CVPR*, 2021.

[39] W. Dai, J. othersFan, Y. Miao, and K. Hwang, "Deep learning model compression with rank reduction in tensor decomposition," *NNLSIEEE Transactions on Neural Networks and Learning Systems*, 2023.

[40] A. Tjandra, S. othersSakti, and S. Nakamura, "Recurrent neural network compression based on low-rank tensor representation," *IEICE TRANSACTIONS on Information and Systems*, 2020.

[41] D. Wang, B. othersWu, G. Zhao, M. Yao, H. Chen, L. Deng, T. Yan, and G. Li, "Kronecker cp decomposition with fast multiplication for compressing rnns," *NNLSIEEE Transactions on Neural Networks and Learning Systems*, 2021.

[42] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, 2002.

[43] C. Sautier, G. Puy, S. othersGidaris, A. Boulch, A. Bursuc, and R. Marlet, "Image-to-lidar self-supervised distillation for autonomous driving data," in *CVPR*, 2022.

[44] G. Li, Z. Ji, S. othersLi, X. Luo, and X. Qu, "Driver behavioral cloning for route following in autonomous vehicles using task knowledge distillation," *IV*, 2022.

[45] J. Im Choi and Q. Tian, "Visual-saliency-guided channel pruning for deep visual detectors in autonomous driving," in *IV*. IEEE, 2023.

[46] W. Yang, H. Yu, B. othersCui, R. Sui, and T. Gu, "Deep neural network pruning method based on sensitive layers and reinforcement learning," *AIrtificial Intelligence Review*, 2023.

[47] K. Samal, M. othersWolf, and S. Mukhopadhyay, "Attention-based activation pruning to reduce data movement in real-time ai: A case-study on local motion planning in autonomous vehicles," *IEEE*, 2020.

[48] S. Gheorghe and M. Ivanovici, "Model-based weight quantization for convolutional neural network compression," in *EMES*. IEEE, 2021.

[49] G. Sciangula, F. Restuccia, A. othersBiondi, and G. Buttazzo, "Hardware acceleration of deep neural networks for autonomous driving on fpga-based soc," in *DSD*. IEEE, 2022.

[50] J. Wang and G. othersJoshi, "Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms," in *ICLRWW*, 2018.

[51] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2019.

[52] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016.

[53] R. Bro, R. A. othersHarshman, N. D. Sidiropoulos, and M. E. Lundy, "Modeling multi-way data with linearly dependent loadings," *Journal of Chemometrics: A Journal of the Chemometrics Society*, 2009.

[54] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, pp. 164–189, 1927.

[55] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *NIPS*, 2018.

[56] M. Sandler, A. Howard, M. Zhu, *et al.*, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018.

[57] A. Loquercio, A. I. Maqueda, C. R. del Blanco, *et al.*, "Dronet: Learning to fly by driving," *RA-L*, 2018.

[58] B. McMahan, E. Moore, D. Ramage, *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[59] T. Li, A. K. Sahu, M. Zaheer, *et al.*, "Federated optimization in heterogeneous networks," *arXiv*, 2018.

[60] F. Sattler, S. Wiedemann, *et al.*, "Robust and communication-efficient federated learning from non-iid data," *TNNLS*, 2019.

[61] B. Yu, H. Yin, and Z. othersZhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAIProceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2018.

[62] J. Wang, A. K. Sahu, Z. Yang, *et al.*, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *ICC*, 2019.

[63] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, 1957.

[64] O. Marfoq, C. Xu, G. Neglia, *et al.*, "Throughput-optimal topology design for cross-silo federated learning," in *NIPS*, 2020.

[65] Udacity, "An open source self-driving car," 2016.

[66] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," *University of Toronto, Technical Report*, vol. 6, 2012.

[67] S. Knight, H. X. Nguyen, N. Falkner, *et al.*, "The internet topology zoo," *J-SAC*, 2011.

[68] F. P. Miller, A. F. Vandome, and J. McBrewster, "Amazon web services," *Retrieved November*, 2011.