# Addressing Non-IID Problem in Federated Autonomous Driving with Contrastive Divergence Loss

Tuong Do[1], Binh X. Nguyen[1], Hien Nguyen[1] Erman Tjiputra[1], Quang D. Tran[1], Anh Nguyen[2]

***Abstract*— Federated learning has been widely applied in autonomous driving since it enables training a learning model among vehicles without sharing users' data. However, data from autonomous vehicles usually suffer from the non-independent-and-identically-distributed (non-IID) problem, which may cause negative effects on the convergence of the learning process. In this paper, we propose a new contrastive divergence loss to address the non-IID problem in autonomous driving by reducing the impact of divergence factors from transmitted models during the local learning process of each silo. We also analyze the effects of contrastive divergence in various autonomous driving scenarios, under multiple network infrastructures, and with different centralized/distributed learning schemes. Our intensive experiments on three datasets demonstrate that our proposed contrastive divergence loss further improves the performance over current state-of-the-art approaches.**

## I. Introduction

Autonomous driving is an emerging field that enables vehicles to operate without a human driver by using a combination of vision, learning, and control algorithms to observe and respond to changes in the environment [1]. Recently, many works have been proposed to address different problems in autonomous driving [2]–[14]. While significant progress has been made in the field, traditional works utilize supervised learning methods and require data collection to train the model [15]–[19]. Although collecting data is necessary to improve the accuracy of the system, it strongly violates user privacy since the users' data are shared with third parties. To overcome this limitation, recent works have adapted federated learning as a new learning mechanism. Federated learning allows multiple parties to collaboratively train a model without sharing their data [20]–[23]. In practice, federated learning enables autonomous vehicles to learn a shared prediction model together, involves more diverse data, and yields real-time predictions while preserving users privacy [24].

Typically, there are two main federated learning scenarios [25]: Server-based Federated Learning (SFL) which has a central node to orchestrate the training process and receive the contributions of all clients, and Decentralized Federated Learning (DFL) which utilizes a fully peer-to-peer (P2P) setup between data silos using a predefined topology. Although SFL can enhance data privacy since only model weights are transmitted, having orchestration nodes potentially represents a bottleneck of the system since most of the data transmission has to go through the central nodes. On the other hand, DFL does not require a server and uses a fully distributed network. Therefore, it is straightforward in DFL to remove or add new silos using a dynamic topology [26] whenever a silo is within the communication range of other silos. In autonomous driving, several works have explored both DFL and SFL to address different problems such as collision avoidance [21], [27], trajectory prediction [28]–[30], and steering prediction [20], [22], [31].

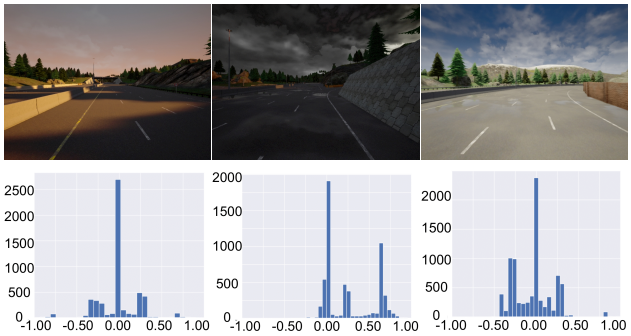



Fig. 1. Sample viewpoints over three different vehicles accompanied with their steering angle distributions in Carla dataset [20]. As we can see, the data from the three vehicles have differences in visual input as well as output steering angle distribution.

In practice, while SFL or DFL approaches have their own advantages and limitations, both of them suffer from the non-IID problem in federated learning. According to [32], [33], the non-IID (identically and independently distributed) problem occurs when data partitioning across silos has a significant distribution shift. Although the non-IID problem occurred in many contexts, it is an immense problem in autonomous driving and causes difficulties when the accumulation process for all vehicle silos is conducted [34]. In practice, each vehicle has its unique driving patterns, weather conditions, and road types, which can cause differences in the data distribution. For example, data collected from a car driving on a highway may be different from data collected from a car driving in a busy city center. When the accumulation process for all vehicle silos is conducted, the non-IID problem can cause difficulties in building a robust and accurate machine-learning model. E.g., if the model is trained mainly on data from highways, it may perform poorly in urban environments, where the distribution of the data is different. Figure 1 illustrates different scenarios when we have the non-IID problem in autonomous driving.

In this paper, we propose a new Contrastive Divergence Loss (CDL) function to address the non-IID problem in

[1]AIOZ, Singapore {tuong.khanh-long.do, binh.xuan.nguyen, hien.nguyen, erman.tjiputra, quang.tran}@aioz.io
[2]Deparment of Computer Science, University of Liverpool, UK anh.nguyen@liverpool.ac.uk

autonomous driving. Unlike prior works that address the non-IID problem by optimizing the accumulation step [34], [35], we directly reduce the impact of divergence factors from transmitted models during the learning process of each local silo. We apply our proposed CDL in the context of the Siamese network and show that it can significantly improve over state-of-the-art methods.

## II. Related works

**Autonomous Driving.** Autonomous driving is an emerging field that has attracted significant research interests in recent years. Several works have focused on using deep learning for autonomous vehicle control, such as anomaly detection [8], [9], [36], [37], object detection and tracking [10]–[12], trajectory prediction [13], [14], autonomous braking and steering [3]–[5], [7], [38]–[40]. Recently, Xin *et al.* [41] proposed a recursive backstepping steering controller that effectively links yaw-rate-based path following commands to the steering angle. Xiong *et al.* [31] analyzed the nonlinear dynamics behavior using a proportional control law. Yi *et al.* [42] presented an algorithm to select the instantaneous center of rotation within the self-reconfigurable robot's area and perform static rotation to adjust its heading angle during waypoint navigation while avoiding collisions. Recently, Yin *et al.* [28] combined model predictive control with covariance steering theory to obtain a robust controller for general nonlinear autonomous driving systems.

**Federated Learning for Autonomous Driving.** Federated learning allows many participants to cooperatively train a machine learning model without disclosing their local data [24]. Federated learning offers a privacy-aware solution to many automotive systems, such as cooperative autonomous driving and intelligent transport systems, which require efficient communication, computation, and storage [43]. Recently, many works have been proposed to address different problems in autonomous driving using federated learning [44]–[48]. Liang *et al.* [21] presented an online federated reinforcement transfer learning process where all the vehicles make actions from the knowledge learned by others. The authors in [22] proposed an autonomous driving system to preserve vehicle privacy by using a server to store shared training models between vehicles. Zhang *et al.* [44] proposed a real-time end-to-end approach that included a unique asynchronous model aggregation mechanism. The authors in [20] introduced a deep federated network for steering angle prediction. Recently, Doomra *et al.* [49] used federated learning to predict the turning signal.

Although applying federated learning in autonomous driving is trendy, there are several open challenges. Besides the crucial disadvantages correlated to the availability and quality of the local computing devices, the variability in data distribution across the participants is also a fundamental difficulty. Specifically, the data of various vehicles are usually non-independent and identically distributed (i.e., non-IID). While non-IID is a critical problem, most of the recent works focus on addressing it through the accumulation process [34], [35] and do not focus on local silos optimization.

**Contrastive Divergence.** Contrastive models have been studied in federated learning recently for dealing with the heterogeneity of local data distribution across parties [22]. This type of model has been applied to vision datasets [50]–[54], language datasets [55], and signal datasets [56]. Indeed, contrastive learning is a valuable candidate for dealing with the non-IID problem in federated autonomous driving, which is mostly caused by local data heterogeneity. It is worth noting that most contrastive-related works focus on building optimized frameworks in server-based [57], [58], modifying topology in P2P federated learning [53], or adjusting the accumulation process [59], [60]. As far as our knowledge, no works have considered the contrastive behavior in the divergence situation of loss during the optimization process in local siloes in federated autonomous driving.

In this work, unlike other approaches that focus on the accumulation step, we propose a new Contrastive Divergence Loss (CDL) to address the non-IID problem in the local silos. Our method can be applied in both SFL and DFL scenarios and improve the performance of the network.

## III. Preliminary

### A. Notation

We summarize the mathematical notations of our paper in Table I.

TABLE I
Mathematical Notations.

| Not. | Description | Not. | Description |
|---|---|---|---|
| $\xi$ | Data in a mini-batch | $i, j$ | Workers (silos) |
| $\alpha$ | Learning rate | $m$ | Mini-batch size |
| $\mathbf{A}$ | Consensus matrix | $\mathcal{L}$ | Loss |
| $k$ | One specific iteration | $\vartheta \in \{0, 1\}$ | Aggregation status |
| $x$ | Input image of $\xi$ in $k$-th iteration | $x'$ | Input image of $\xi$ in $(k-1)$-th iteration |
| $\mathcal{H}$ | Kullback-Leibler Divergence | $\theta$ | Learnable parameters |
| $\beta$ | Pulling control parameter for Kullback-Leibler Divergence | $\mathcal{N}_i^+$ | In-neighbors of silo $i$ |
| $\mathbf{K}$ | Transition kernel | $\mathbf{k}_1$ | 1-st transitional kernel constraint |

### B. Federated Learning for Autonomous Driving

In autonomous driving, we consider each autonomous vehicle as a data silo. Our goal is to collaboratively train a global driving policy $\theta$ from $N$ silos by aggregating all local learnable weights $\theta_i$ of each silo. Each silo computes the current gradient of its local loss function and then updates the model parameter using an optimizer. After completing the local update, each silo engages in several inter-silo interactions using network topology. Mathematically, in the local update stage, at each silo $i$, in each iteration $k$, the model weights of the local silo can be computed as:

$$\theta_i\left(k+1\right) = \theta_i\left(k\right) - \alpha_k \frac{1}{m} \sum_{h=1}^{m} \nabla \mathcal{L}_{\mathrm{lr}}\left(\theta_i\left(k\right), \xi_i^h\left(k\right)\right) \quad (1)$$

where $\mathcal{L}_{\mathrm{lr}}$ is the local regression loss for autonomous steering. To update the global model, each silo interacts with the

associated ones through a predefined topology. The global model is then updated as:

$$\theta\left(k+1\right)=\sum_{i=0}^{N-1}\vartheta_i\theta_i\left(k\right) \tag{2}$$

In practice, the local model in each silo is trained using the regression loss, parameterized by a deep network that takes RGB images as inputs and predicts the associated steering angles [20].

## IV. Methodology

In this section, we discuss the distribution synchronization challenge which is the key motivation that leads to our proposed contrastive divergence loss. We then discuss the Siamese network and the detail of our contrastive divergence loss to handle the non-IID problem for autonomous driving.

### A. Overview

**Motivation.** Due to the non-IID problem, the federated learning algorithms for autonomous driving only achieve good results when two conditions are met: *i)* the local silo $i$ can effectively learn from its local data, and *ii)* the synchronization between nearby silos is sufficient to minimize the effect of the non-IID problem. Currently, most recent works address both of these problems by optimizing the accumulation progress and optimizer [34], [47], [48], proposing new topologies [45], or utilizing high-performed deep networks that are robust to the non-identical characteristic of distributed data [35], [61]. However, according to [62], naively adopting high-performed deep architecture on centralized local learning and its corresponding optimizations into federated scenarios can increase the weight variance of local silo weights during the weights accumulation between silos. As a consequence, it affects the convergence ability of the model and may also cause divergence.

**Siamese Network.** In this work, we propose to address the non-IID problem directly from each local silo by solving the learning good feature problem and synchronization challenge separately. Consequently, we want to have *two networks in each silo*. One network is responsible for learning meaningful features from the local image data, and the other network is responsible for minimizing the distribution gap between the current model weight and other nearby weights. To this end, the Siamese Network [63] which includes two networks is well-fitted to our needs. In particular, the Siamese network has two branches. We consider the first branch (*backbone* network) of the Siamese to have the local regression loss $\mathcal{L}_{\text{lr}}$ to learn the local image features for autonomous steering and the positive contrastive divergence loss $\mathcal{L}_{\text{cd}^+}$ to learn knowledge transmitted from neighbor silos. The second branch (*sub-network*) is ultilized to control the divergence factors from knowledge of the backbone through the contrastive regularizer term $\mathcal{L}_{\text{cd}^-}$ (See Fig. 2).

In practice, the sub-network shares the same weights with the backbone during the first communication round. However, from the next communication round, after the backbone is accumulated using Equation 1, the local model of each silo is trained by the contrastive divergence loss. The sub-network outputs support features that have the same size as the output features of the backbone. During the training, we expect that the weights between the backbone and the sub-network should not have significant differences while applying the contrastive divergence loss. The weights of all silos are synchronized whenever gradients from the learning process of the backbone and the sub-network of all silos are not significantly different.

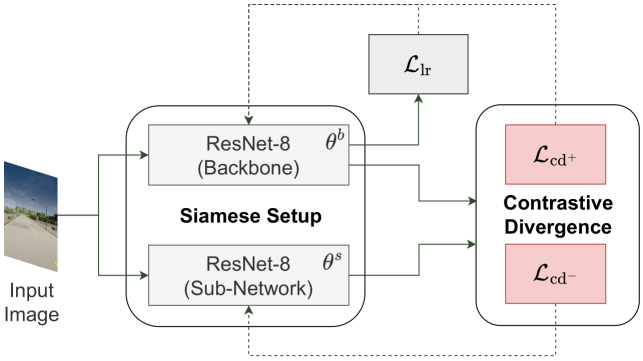



Fig. 2. The Siamese setup when our CDL is applied for training federated autonomous driving model. ResNet-8 is used in the backbone and sub-network in the Siamese setup. During inference, the sub-network will be removed. Dotted lines represent the backward process. Our CDL has two components: the positive contrastive divergence loss $\mathcal{L}_{\text{cd}^+}$ and the negative regularize term $\mathcal{L}_{\text{cd}^-}$. The local regression loss $\mathcal{L}_{\text{lr}}$ for automatic steering prediction is calculated only from the backbone network.

### B. Contrastive Divergence Loss

In practice, we observe that the early stages of federated learning mostly have poor accumulated models. Different from other works that deal with the non-IID problem by optimizing the accumulation step whenever silos transmit their models, we directly decrease the effect of divergence factors during the local learning process of each silo. To achieve that, we reduce the distance between distribution from accumulated weights $\theta_i^b$ at silo $i$ in the backbone network, which contains information from other silos known as divergence factors, and its $i$-th silo weights $\theta_i^s$ in the sub-network, which only contains knowledge learned from local data. When the distribution between silos has been synchronized at an acceptable rate, we lower the effectiveness of the sub-network and focus more on the steering angle prediction task. Our proposed Contrastive Divergence Loss is motivated by the contrastive loss from the original Siamese Network [63] and is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cd}} &= \beta\mathcal{L}_{\text{cd}^+}+(1-\beta)\mathcal{L}_{\text{cd}^-}\\ &= \beta\mathcal{H}(\theta_i^b,\theta_i^s)+(1-\beta)\mathcal{H}(\theta_i^s,\theta_i^b)\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{cd}^+}$ is the positive contrastive divergence term and $\mathcal{L}_{\text{cd}^-}$ is the negative regularizer term; $\mathcal{H}$ is the Kullback-Leibler Divergence loss function [64]:

$$\mathcal{H}(\hat{y},y)=\sum\mathbf{f}(\hat{y})\log\left(\frac{\mathbf{f}(\hat{y})}{\mathbf{f}(y)}\right) \tag{4}$$

where $\hat{y}$ is the predicted representation, $y$ is dynamic soft label.

Consider $\mathcal{L}_{\text{cd}^+}$ in Equation 3 as a Bayesian statistical inference task, our goal is to estimate the model parameters $\theta^{b*}$ by minimizing the Kullback-Leibler divergence $\mathcal{H}(\theta_i^b, \theta_i^s)$ between the measured regression probability distribution of the observed local silo $P_0(x|\theta_i^s)$ and the accumulated model $P(x|\theta_i^b)$. Hence, we can assume that the model distribution has a form of $P(x|\theta_i^b) = e^{-E(x,\theta_i^b)}/Z(\theta_i^b)$, where $Z(\theta_i^b)$ is the normalization term. However, evaluating the normalization term $Z(\theta_i^b)$ is not trivial, which leads to risks of getting stuck in a local minimum. Inspired by Hinton [65], we use samples obtained through a Markov Chain Monte Carlo (MCMC) procedure with a specific initialization strategy to deal with the mentioned problem. Additionally inferred from Equation 1, the $\mathcal{L}_{\text{cd}^+}$ can be expressed under the SGD algorithm in a local silo by setting:

$$\mathcal{L}_{\text{cd}^+} = -\sum_x P_0(x|\theta_i^s)\frac{\partial E(x;\theta_i^b)}{\partial \theta_i^b} + \sum_x Q_{\theta_i^b}(x|\theta_i^s)\frac{\partial E(x;\theta_i^b)}{\partial \theta_i^b} \tag{5}$$

where $Q_{\theta_i^b}(x|\theta_i^s)$ is the measured probability distribution on the samples obtained by initializing the chain at $P_0(x|\theta_i^s)$ and running the Markov chain forward for a defined step. The convergence analysis of $\mathcal{L}_{\text{cd}^+}$ can be found in our Appendix.

Consider $\mathcal{L}_{\text{cd}^-}$ regularizer in Equation 3 as a Bayesian statistical inference task, we can calculate $\mathcal{L}_{\text{cd}^-}$ as in Equation 5, however, the role of $\theta^s$ and $\theta^b$ is inverse:

$$\mathcal{L}_{\text{cd}^-} = -\sum_x P_0(x|\theta_i^b)\frac{\partial E(x;\theta_i^s)}{\partial \theta_i^s} + \sum_x Q_{\theta_i^s}(x|\theta_i^b)\frac{\partial E(x;\theta_i^s)}{\partial \theta_i^s} \tag{6}$$

We note that although Equation 5 and Equation 6 share the same structure, the key difference is that while the weight $\theta_i^b$ of the backbone is updated by the accumulation process from Equation 2, the weight $\theta_i^s$ of the sub-network, instead, is not. This lead to different convergence behavior of contrastive divergence in $\mathcal{L}_{\text{cd}^+}$ and $\mathcal{L}_{\text{cd}^-}$. The negative regularizer term $\mathcal{L}_{\text{cd}^-}$ will converge to state $\theta_i^{s*}$ provided $\frac{\partial E}{\partial \theta_i^s}$ is bounded:

$$g(x, \theta_i^s) = \frac{\partial E(x;\theta_i^s)}{\partial \theta_i^s} - \sum_x P_0(x|(\theta_i^b, \theta_i^s))\frac{\partial E(x;\theta_i^s)}{\partial \theta_i^s} \tag{7}$$

and

$$(\theta_i^s - \theta_i^{s*}) \cdot \left\{ \sum_x P_0(x) g(x, \theta_i^s) - \sum_{x', x} P_0(x') \mathbf{K}_{\theta_i^s}^m (x', x) g(x, \theta_i^{s*}) \right\} \geq \mathbf{k}_1 |\theta_i^s - \theta_i^{s*}|^2 \tag{8}$$

for any $\mathbf{k}_1$ constraint. Note that, $\mathbf{K}_{\theta^s}^m$ is the transition kernel. The proof for the above result is analyzed in [66].

Note that the negative regularizer term $\mathcal{L}_{\text{cd}^-}$ is only used in training models on local silos. Thus, it does not contribute to the accumulation process of federated training.

### C. Total Training Loss

**Local Regression Loss.** We use mean square error (MAE) to compute loss for predicting the steering angle in each local silo. Note that, we only use features from the backbone for predicting steering angles.

$$\mathcal{L}_{\text{lr}} = \text{MAE}(\theta_i^b, \hat{\xi}_i) \tag{9}$$

where $\hat{\xi}_i$ is the ground-truth steering angle of the data sample $\xi_i$ collected from silo $i$.

**Local Silo Loss.** The local silo loss computed in each communication round at each silo before applying the accumulation process is described as:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{lr}} + \mathcal{L}_{\text{cd}} \tag{10}$$

In practice, we observe that both the contrastive divergence loss $\mathcal{L}_{\text{cd}}$ to handle the non-IID problem and the local regression loss $\mathcal{L}_{\text{lr}}$ for predicting the steering angle is equally important and indispensable. Hence, we do not set a parameter to control their contributions in Equation 10.

Combining all losses together, at each iteration $k$, the update in the backbone network is defined as:

$$\theta_i^b(k+1) = \begin{cases} \sum_{j \in \mathcal{N}_i^+ \cup \{i\}} \mathbf{A}_{i,j} \theta_j^b(k), & \text{if } k \equiv 0 \pmod{u+1}, \\ \theta_i^b(k) - \alpha_k \frac{1}{m} \sum_{h=1}^m \nabla \mathcal{L}_{\text{b}} \left(\theta_i^b(k), \xi_i^h(k)\right), & \text{otherwise.} \end{cases} \tag{11}$$

where $\mathcal{L}_{\text{b}} = \mathcal{L}_{\text{lr}} + \mathcal{L}_{\text{cd}^+}$, $u$ is the number of local updates.

In parallel, the update in the sub-network at each iteration $k$ is described as:

$$\theta_i^s(k+1) = \theta_i^s(k) - \alpha_k \frac{1}{m} \sum_{h=1}^m \nabla \mathcal{L}_{\text{cd}^-} \left(\theta_i^s(k), \xi_i^h(k)\right) \tag{12}$$

## V. Experimental Results

### A. Implementation

**Dataset.** We use three datasets (Table II) in our experiment: Udacity+ [67], Gazebo Indoor [20], and Carla Outdoor dataset [20]. Gazebo and Calar are non-IID datasets while Udacity+ is the non-IID version of Udacity dataset.

TABLE II
THE STATISTIC OF DATASETS IN OUR EXPERIMENTS.

| Dataset | Total samples | Average samples in each silo | | |
|---|---|---|---|---|
| | | *Gaia [68] (11 silos)* | *NWS [69] (22 silos)* | *Exodus [68] (79 silos)* |
| Udacity+ | 38,586 | 3,508 | 1,754 | 488 |
| Gazebo | 66,806 | 6,073 | 3,037 | 846 |
| Carla | 73,235 | 6,658 | 3,329 | 927 |

**Network Topology.** Following [25], we conduct experiments on three federated topologies: the Internet Topology Zoo [68] (Gaia), the North American data centers [69] (NWS), and the Zoo Exodus network (Exodus) [68]. We use Gaia topology in our main experiment and provide the comparison of two other topologies in our ablation study.

**Training.** The model in a silo is trained with a batch size of 32 and a learning rate of $0.001$ using Adam optimizer. In each communication round, the local training process

TABLE III

PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS. THE GAIA TOPOLOGY IS USED.

| Model | Main Focus | Learning Method | RMSE | | | MAE | | | # Training Parameters | Avg. Cycle Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | *Udacity+* | *Gazebo* | *Carla* | *Udacity+* | *Gazebo* | *Carla* | | |
| Random [70] | - Statistical | - | 0.358 | 0.117 | 0.464 | 0.265 | 0.087 | 0.361 | - | - |
| Constant [70] | | | 0.311 | 0.092 | 0.348 | 0.209 | 0.067 | 0.232 | - | - |
| Inception [71] | Architecture Design | CLL [70] | 0.209 | 0.085 | 0.297 | 0.197 | 0.062 | 0.207 | 21,787,617 | - |
| MobileNet [72] | | | 0.193 | 0.083 | 0.286 | 0.176 | 0.057 | 0.200 | 2,225,153 | - |
| VGG-16 [73] | | | 0.190 | 0.083 | 0.316 | 0.161 | 0.050 | 0.184 | 7,501,587 | - |
| DroNet [70] | | | 0.183 | 0.082 | 0.333 | 0.150 | 0.053 | 0.218 | 314,657 | - |
| FedAvg [74] | Aggregation Optimization | SFL [32] | 0.212 | 0.094 | 0.269 | 0.185 | 0.064 | 0.222 | 314,657 | 152.4 |
| FedProx [75] | | | 0.152 | 0.077 | 0.226 | 0.118 | 0.063 | 0.151 | 314,657 | 111.5 |
| STAR [32] | | | 0.179 | 0.062 | 0.208 | 0.149 | 0.053 | 0.155 | 314,657 | 299.9 |
| MATCHA [76] | Topology Design | DFL [20] | 0.182 | 0.069 | 0.208 | 0.148 | 0.058 | 0.215 | 314,657 | 171.3 |
| MBST [25], [77] | | | 0.183 | 0.072 | 0.214 | 0.149 | 0.058 | 0.206 | 314,657 | 82.1 |
| FADNet [20] | | | 0.162 | 0.069 | 0.203 | 0.134 | 0.055 | 0.197 | 317,729 | **62.6** |
| **CDL (ours)** | Loss Optimization | CLL [20] | 0.169 | 0.074 | 0.266 | 0.149 | 0.053 | 0.172 | 629,314 | - |
| | | SFL [20] | 0.150 | **0.060** | 0.208 | 0.104 | **0.052** | 0.150 | 629,314 | 102.2 |
| | | DFL [20] | **0.141** | 0.062 | **0.183** | **0.083** | **0.052** | **0.147** | 629,314 | 72.7 |

is done in each silo before their models are transmitted and aggregated using Equation 2. The training process is conducted with $3,600$ communication rounds. We apply the simulation as in [20] to do the training with an NVIDIA 1080 11GB GPU.

**Baselines.** We compare our results with various recent methods in different learning scenarios. For the non-learning scenario, we compare our method with the Random baseline and the Constant baseline [70]. For the Centralized Local Learning (CLL) scenario, Inception-V3 [71], MobileNet-V2 [72], VGG-16 [73], and Dronet [70] are used as baselines. For the Server-based Federated Learning (SFL) scenario, we compare our method with FedAvg [74], FedProx [75], and STAR [32]. For the Decentralized Federated Learning (DFL), MATCHA [76], MBST [25], and FADNet [20] are used as baselines. Note that the Centralized Local Learning (CLL [70]) strategy has the data collected and model trained in one local machine; Server-based Federated Learning (SFL [32]) strategy has models trained in distributed clients and collected by a server; Decentralized Federated Learning (DFL [20]) has models trained in a fully P2P manner.

**Evaluation Metric.** We use the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to evaluate the effectiveness of models. Besides, cycle time (ms) and wall-clock time (ms) are used to calculate the time needed for training and testing each method.

### *B. Results*

Table III summarises the performance of our method and recent state-of-the-art approaches. This table clearly shows the proposed CDL under Siamese setup with two ResNet-8 outperforms other methods by a large margin. In particular, our proposal significantly reduces both RMSE and MAE in all three datasets, including Udacity+, Carla, and Gazebo. These results validate that our method reduces the negative effect of the non-IID problem during the training. As a loss function, CDL does not increase the number of parameters of the network. However, under the Siamese setup, the model size during the training is increased as the Siamese requires an additional sub-network.

TABLE IV

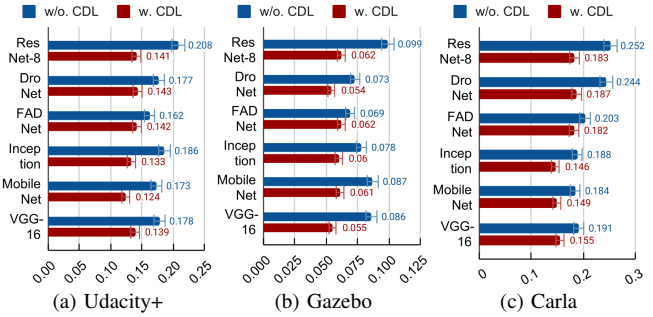
PERFORMANCE UNDER DIFFERENT TOPOLOGIES.

| Topology | Architecture | Dataset | | |
|---|---|---|---|---|
| | | *Udacity+* | *Gazebo* | *Carla* |
| Gaia (11 silos) | DroNet [70] | 0.177 | 0.073 | 0.244 |
| | FADNet [20] | 0.162 | 0.069 | 0.203 |
| | CDL (ours) | 0.141 | 0.062 | 0.183 |
| NWS (22 silos) | DroNet [70] | 0.183 | 0.075 | 0.239 |
| | FADNet [20] | 0.165 | 0.070 | 0.200 |
| | CDL (ours) | 0.138 | 0.058 | 0.182 |
| Exodus (79 silos) | DroNet [70] | 0.448 | 0.208 | 0.556 |
| | FADNet [20] | 0.179 | 0.081 | 0.238 |
| | CDL (ours) | 0.138 | 0.061 | 0.176 |

Table III also summarises the accuracy of our CDL with ResNet-8 backbone and other baselines when we train them under different learning scenarios: CLL, SFL, and DFL. We can see that our CDL with ResNet-8 outperforms other baselines by a large margin in the DFL learning scenario, and by a moderate value in SFL and CLL setup. This confirms that our method is not only a good solution for dealing with the non-IID problem in federated autonomous driving, but also improves the convergence ability of the backbone in different training setups. Moreover, the outperformed gaps are stable among all three benchmarking datasets.

### *C. Ablation Study*

**CDL on Different Topologies.** In practice, it is usually more challenging to train federated algorithms when the topology has more vehicle data silos. To verify the effective-




(a) Udacity+ (b) Gazebo (c) Carla

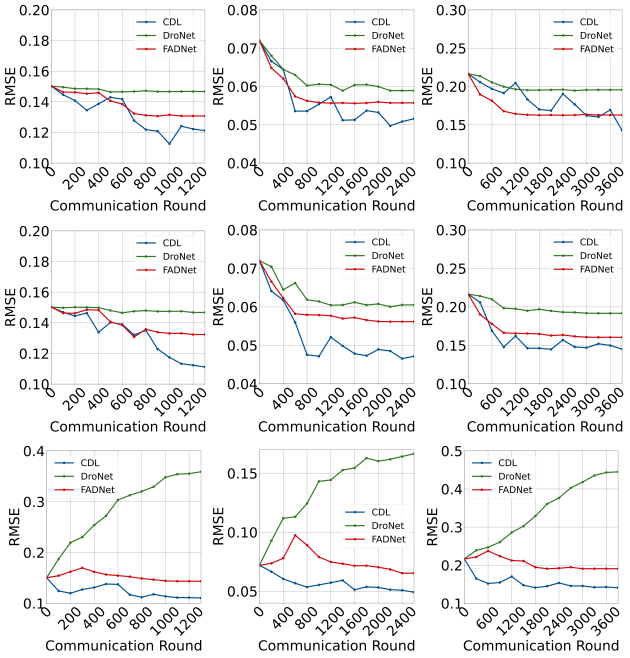
Fig. 3. Performance of our CDL when we use different networks under Siamese setup. Note that, we train on the train set and test on the test set of the Udacity+, Gazebo, and Carla datasets. We use Gaia topology under the DFL scenario for the training. The results are reported using the RMSE. Best viewed in color.

ness of our CDL, we train our method and compare it with other baselines on topologies with different numbers of silos. Table IV illustrates the performance of DroNet, FADNet, and our CDL with ResNet-8 backbone when we train them using DFL under three distributed network infrastructures with different numbers of silos: Gaia (11 silos), NWS (22 silos), and Exodus (79). This table shows that our CDL clearly achieves the highest results in all topology setups, while DroNet meets divergence, and FADNet does not perform well in the Exodus topology which has 79 silos.

**CDL on Different Backbones.** Since our proposed CDL is a loss function, it can be applied to different networks under the Siamese setup to improve performance. Figure 3 illustrates the effectiveness of CDL when we change the network inside the Siamese to DroNet, FADNet, Inception, MobileNet, and VGG-16 under Gaia Network in the DFL scenario. The results show that CDL works well with different architectures to address the non-IID problem. In all cases, CDL greatly improves performance on different datasets with different networks inside the Siamese setup.

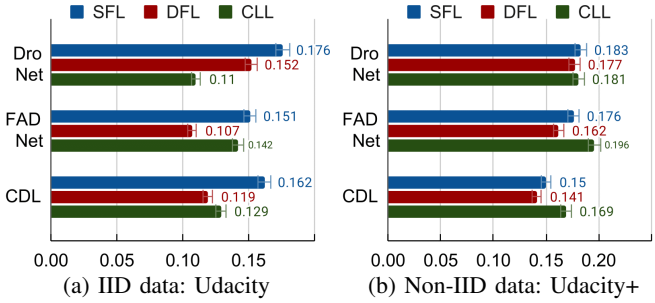
**Convergence Analysis.** Figure 4 illustrates the training results in RMSE of our two baselines DroNet and FADNet as well as our proposed CDL. The results show the convergence ability of mentioned methods in three datasets (Udacity+, Gazebo, Carla), and in Gaia (11 silos), NWS Amazon (22 silos), and Exodus (79 silos) networks. The results indicate that our proposed CDL can reach a better convergence point in comparison with the two baselines. While other methods (Dronet and FADNet) converged with difficulty or do not show good convergence trend, our proposed CDL can get over local optimal points better than other methods and also be less biased into any specific silo.

**CDL on IID Data.** Figure 5 demonstrates the effectiveness of CDL in different data distributions. Although CDL is designed for dealing with the non-IID problem, it also slightly improves the performance of models trained on IID data distribution. Based on the Siamese setup, CDL inherits the behavior and characteristic of triplet loss form. Since triplet loss is proven to be effective in IID data [78], it is clear that CDL can also improve the performance of models when we train them with the IID data.




(a) Udacity+ (b) Gazebo (c) Carla

Fig. 4. The convergence ability of different methods under Gaia topology (first row), NWS topology (second row), and Exodus topology (third row). Note that the results are reported in the train set.




(a) IID data: Udacity (b) Non-IID data: Udacity+

Fig. 5. Performance of different methods on IID dataset (Udacity) and non-IID dataset (Udacity+). Note that all methods are trained on Gaia topology. The number of communication rounds is set to 900 and 1200 on Udacity and Undacity+ datasets, respectively. The results are reported using the RMSE. Best viewed in color.

## VI. Conclusion

This paper has presented a new method to address the non-IID problem during the federated learning of autonomous steering using contrastive divergence loss. Our method directly reduces the effect of divergence factors from transmitted models during the local learning process of each silo. We analyze the proposed contrastive divergence loss in theories, on various autonomous driving scenarios, under multiple network topologies, and with different centralized/distributed learning schemes. The thorough experimental findings on three benchmarking datasets demonstrate that our proposed contrastive divergence loss performs substantially better than current state-of-the-art approaches. In the future, we plan to test our strategy with more data silos and deploy the trained model using an autonomous vehicle on roads. We will also release our source code to encourage further study.

## REFERENCES


[1] S. Grigorescu, B. Trasnea, T. Cocias, *et al.*, "A survey of deep learning techniques for autonomous driving," *JFR*, 2020.

[2] D. Parekh, N. Poddar, A. Rajpurkar, *et al.*, "A review on autonomous vehicles: Progress, methods and challenges," *Electronics*, 2022.

[3] P. Long, T. Fan, *et al.*, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *ICRA*, 2018.

[4] A. Kendall, J. Hawke, D. Janz, *et al.*, "Learning to drive in a day," in *ICRA*, 2019.

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

[6] A. Nguyen, N. Nguyen, K. Tran, *et al.*, "Autonomous navigation in complex environments with deep multimodal fusion network," in *IROS*, 2020.

[7] L. Bergamini, Y. Ye, O. Scheel, *et al.*, "Simnet: Learning reactive self-driving simulations from real-world observations," in *ICRA*, 2021.

[8] X. Du, Z. Wang, M. Cai, *et al.*, "Vos: Learning what you don't know by virtual outlier synthesis," in *ICLR*, 2022.

[9] D. Bogdoll, M. Nitsche, and J. M. Zöllner, "Anomaly detection in autonomous driving: A survey," in *CVPR*, 2022.

[10] P. Li and J. Jin, "Time3d: End-to-end joint monocular 3d object detection and tracking for autonomous driving," in *CVPR*, 2022.

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

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

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

[14] Q. Zhang, S. Hu, J. Sun, *et al.*, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *CVPR*, 2022.

[15] K. Huang and Q. Hao, "Joint multi-object detection and tracking with camera-lidar fusion for autonomous driving," in *IROS*, 2021.

[16] M. Vitelli, Y. Chang, *et al.*, "Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies," in *ICRA*, 2022.

[17] S. Chen, Y. Sun, D. Li, *et al.*, "Runtime safety assurance for learning-enabled control of autonomous driving vehicles," in *ICRA*, 2022.

[18] A. Amini, T.-H. Wang, I. Gilitschenski, *et al.*, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *ICRA*, 2022.

[19] E. Bronstein, M. Palatucci, D. Notz, *et al.*, "Hierarchical model-based imitation learning for planning in autonomous driving," in *IROS*, 2022.

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

[21] X. Liang, Y. Liu, T. Chen, *et al.*, "Federated transfer reinforcement learning for autonomous driving," *arXiv*, 2019.

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

[23] B. X. Nguyen, T. Do, *et al.*, "Multigraph topology design for cross-silo federated learning," *arXiv preprint arXiv:2207.09657*, 2022.

[24] C. Zhang, Y. Xie, H. Bai, *et al.*, "A survey on federated learning," *KBS*, 2021.

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

[26] N. Amelina, A. Fradkov, and K. Amelin, "Approximate consensus in multi-agent stochastic systems with switched topology and noise," in *CCTA*, 2012.

[27] W. Hammedi, B. Brik, and S. M. Senouci, "Toward optimal mec-based collision avoidance system for cooperative inland vessels: a federated deep learning approach," *T-ITS*, 2022.

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

[29] N. Majcherczyk, N. Srishankar, and C. Pinciroli, "Flow-fl: Data-driven federated learning for spatio-temporal predictions in multi-robot systems," in *ICRA*, 2021.

[30] C. Wang, X. Chen, *et al.*, "Atpfl: Automatic trajectory prediction model design under federated learning framework," in *CVPR*, 2022.

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

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

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

[34] B. Li, Y. Jiang, Q. Pei, *et al.*, "Feel: Federated end-to-end learning with non-iid data for vehicular ad hoc networks," *T-ITS*, 2022.

[35] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *IJCNN*, 2021.

[36] S. Jung, J. Lee, D. Gwak, *et al.*, "Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation," in *ICCV*, 2021.

[37] F. Heidecker, A. Hannan, *et al.*, "Towards corner case detection by modeling uncertainty of instance segmentation networks," in *IV*, 2021.

[38] G. Garimella, J. Funke, C. Wang, *et al.*, "Neural network modeling for steering control of an autonomous vehicle," in *IROS*, 2017.

[39] A. Amini, L. Paull, T. Balch, *et al.*, "Learning steering bounds for parallel autonomous systems," in *ICRA*, 2018.

[40] M. Bojarski, A. Choromanska, *et al.*, "Visualbackprop: Efficient visualization of cnns for autonomous driving," in *ICRA*, 2018.

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

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

[43] Z. Du, C. Wu, T. Yoshinaga, *et al.*, "Federated learning for vehicular internet of things: Recent advances and open issues," *OJ-CS*, 2020.

[44] H. Zhang, J. Bosch, and H. H. Olsson, "Real-time end-to-end federated learning: An automotive case study," *arXiv*, 2021.

[45] Y. He, K. Huang, G. Zhang, *et al.*, "Bift: A blockchain-based federated learning system for connected and autonomous vehicles," *IoT-J*, 2021.

[46] M. A. Khan, H. El Sayed, *et al.*, "A journey towards fully autonomous driving-fueled by a smart communication system," *VC*, 2022.

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

[48] R. Parekh, N. Patel, *et al.*, "Gefl: Gradient encryption-aided privacy preserved federated learning for autonomous vehicles," *Access*, 2023.

[49] S. Doomra, N. Kohli, and S. Athavale, "Turn signal prediction: A federated learning case study," *arXiv*, 2020.

[50] N. Le, K. Nguyen, *et al.*, "Uncertainty-aware label distribution learning for facial expression recognition," in *WACV*, 2023.

[51] Q. Li *et al.*, "Model-contrastive federated learning," in *CVPR*, 2021.

[52] M. Mendieta, T. Yang, P. Wang, *et al.*, "Local learning matters: Rethinking data heterogeneity in federated learning," in *CVPR*, 2022.

[53] X. Mu, Y. Shen, K. Cheng, *et al.*, "Fedproc: Prototypical contrastive federated learning on non-iid data," *FGCS*, 2023.

[54] T. Nguyen, M. N. Vu, *et al.*, "Open-vocabulary affordance detection in 3d point clouds," *arXiv preprint arXiv:2303.02401*, 2023.

[55] K. Georgiev and P. Nakov, "A non-iid framework for collaborative filtering with restricted boltzmann machines," in *ICML*, 2013.

[56] V. Tsouvalas, A. Saeed, and T. Ozcelebi, "Federated self-training for semi-supervised audio recognition," *TECS*, 2022.

[57] Y. Tan, G. Long, J. Ma, *et al.*, "Federated learning from pre-trained models: A contrastive learning approach," in *ICMLW*, 2022.

[58] T. Qi, F. Wu, C. Wu, *et al.*, "Fairvfl: A fair vertical federated learning framework with contrastive adversarial learning," in *NIPS*, 2023.

[59] H. Kassem, D. Alapatt, P. Mascagni, *et al.*, "Federated cycling (fedcy): Semi-supervised federated learning of surgical phases," *TMI*, 2022.

[60] S. Hu, L. Feng, X. Yang, *et al.*, "Fedssc: Shared supervised-contrastive federated learning," *arXiv*, 2023.

[61] P. S. Bouzinis *et al.*, "Wireless federated learning (wfl) for 6g networks part i: Research challenges and future trends," *COMML*, 2021.

[62] M. Duan, D. Liu, X. Chen, *et al.*, "Self-balancing federated learning with global imbalanced data in mobile systems," *TPDS*, 2020.

[63] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICMLW*, 2015.

[64] J. M. Joyce, "Kullback-leibler divergence," in *International encyclopedia of statistical science*, 2011.

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

[66] A. L. Yuille, "The convergence of contrastive divergences," *NIPS*, 2004.

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

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

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

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

[71] C. Szegedy, V. Vanhoucke, S. Ioffe, *et al.*, "Rethinking the inception architecture for computer vision," in *CVPR*, 2015.

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

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.
[74] B. McMahan, E. Moore, D. Ramage, *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.
[75] T. Li, A. K. Sahu, M. Zaheer, *et al.*, "Federated optimization in heterogeneous networks," *arXiv*, 2018.
[76] J. Wang, A. K. Sahu, Z. Yang, *et al.*, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *ICC*, 2019.
[77] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, 1957.
[78] T. Wang *et al.*, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*, 2020.