Federated Learning for Collaborative Controller Design of Connected and Autonomous Vehicles

Tengchan Zeng, Omid Semiari, Mingzhe Chen, Walid Saad, and Mehdi Bennis

***Abstract*—The deployment of future intelligent transportation systems is contingent upon seamless and reliable operation of connected and autonomous vehicles (CAVs). One key challenge in developing CAVs is the design of an autonomous controller that can make use of wireless connectivity and accurately execute control decisions, such as a quick acceleration when merging to a highway and frequent speed changes in a stop-and-go traffic. However, the use of conventional feedback controllers or traditional machine learning based controllers, solely trained by the CAV's local data, cannot guarantee a robust controller performance over a wide range of road conditions and traffic dynamics. In this paper, a new federated learning (FL) framework enabled by the CAVs' wireless connectivity is proposed for the autonomous controller design of CAVs. In this framework, the learning models used by the controllers are collaboratively trained among a group of CAVs. To capture the varying CAV participation in FL and the diverse local data quality among CAVs, a novel dynamic federated proximal (DFP) algorithm is proposed that accounts for the mobility of CAVs, the wireless channel dynamics, as well as the unbalanced and non-independent and identically distributed data across CAVs. A rigorous convergence study is performed for the proposed algorithm under realistic wireless environments. Then, the impact of varying CAV participation in FL process and diverse local data quality of CAVs on the convergence is explicitly analyzed. Simulation results that use real vehicular data show that the proposed DFP-based controller can accurately track the target speed over time and under different traffic scenarios, and it yields a distance error two times smaller than controllers designed using traditional machine learning solutions trained with the CAV's local data. The results also show that the proposed DFP algorithm is well-suited for the autonomous controller design in CAVs when compared to popular FL algorithms, such as federated averaging (FedAvg) and federated proximal (FedProx) algorithms.**

I. Introduction

As a key component of tomorrow's intelligent transportation systems (ITSs), connected and autonomous vehicles (CAV) are promising solutions to reduce traffic accidents, alleviate road congestions, and increase transportation efficiency. CAVs leverage both sensors and wireless systems to increase their situational awareness and use such awareness for their motion planning and automatic control. However, to operate a full-fledged CAVs, we need to address a number of challenges, ranging from management of wireless resources to reliable controller design. Among these challenges, designing an autonomous controller to achieve target movements for CAVs is important to accomplish the target tasks and achieve the operation safety. In particular, a CAV's controller must accurately execute navigation decisions so that the CAV can quickly adapt to the dynamic road traffic. For example, the controller must generate frequent slow-down and speed-up for CAVs in a stop-and-go traffic, whereas a rapid acceleration will be the target output for the controller when CAVs merge into highways.

There are two common methods to design an autonomous controller for CAVs. The first method uses a conventional feedback controller. In particular, the conventional feedback controller first needs to determine the CAV's dynamic models (e.g., the tire model [1]) and the road conditions (e.g., road slope [2] and slip ratio between the road and tire [3]), and then optimize the controller design based on these settings. However, due to the changes in payload, various types of roads, dynamic road traffic, and varying weather, the vehicle dynamics and road conditions change constantly over time. Hence, the conventional feedback controller cannot guarantee the controller performance over a wide range of environmental parameter changes. To ensure that the CAVs can adapt to the changing vehicle dynamics and road conditions, the second method is to use adaptive controllers, based on machine learning (ML), for the CAV's autonomy. For example, in [4], the authors propose a learning-based model predictive control (MPC) design where the recorded trajectory data is trained to optimize the parameterization of the MPC controller that leads to the optimal closed-loop performance. Also, a database-driven proportional-integral-derivative (PID) controller is proposed in [5] where ML algorithms are used to train the local database to tune control parameters. However, when using learning methods that require training (e.g., neural networks) for the adaptive controller design, the local training data can be insufficient due to the limited on-chip memory available at CAVs [6]. In addition, due to limited storage, only data pertaining to the most recent driving can be stored and such data can be skewed and of poor quality. As a result, when changing to a new traffic environment or less frequently occurred events, e.g., traffic accidents, a controller solely trained by the local data can fail to adapt to the changes. A cooperative training framework among multiple CAVs will be needed for the controller design.

To this end, one can leverage the wireless connectivity in CAVs and use federated learning (FL) to enable CAVs to collaboratively train the learning models used by the controllers [7]. In essence, FL can allow each CAV to train the local model for the controller based on its own data, and it will use the wireless cellular network to share the trained model parameters

This research was supported, in part, by the U.S. National Science Foundation under Grants CNS-1739642 and CNS-1941348, and by the Academy of Finland Projects CARMA, MISSION, and SMARTER, as well as by the INFOTECH Project NOOR.

T. Zeng and W. Saad are with Wireless@VT, Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24061 USA (e-mail: tengchan@vt.edu; walids@vt.edu).

O. Semiari is with the Department of Electrical and Computer Engineering, University of Colorado, Colorado Springs, CO, 80918 USA (e-mail: osemiari@uccs.edu).

M. Chen is with the Department of Electrical Engineering, Princeton University, Princeton, NJ, 08544 USA (e-mail: mingzhec@princeton.edu).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland (e-mail:mehdi.bennis@oulu.fi).

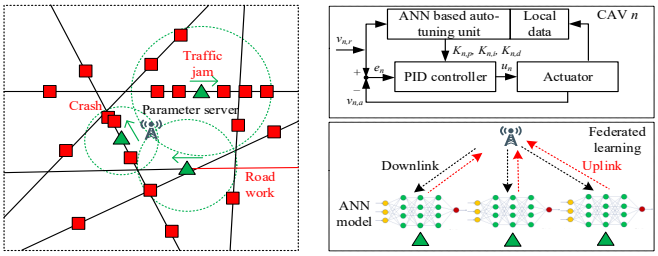
used by its adaptive controller with a parameter server, such as a base station (BS). The parameter server will then aggregate the received model parameters and send the aggregated model parameters back to the CAVs. In this way, the learning model can be collaboratively trained among multiple CAVs, and such a trained model can enable the CAV's controller to adapt to *new traffic scenarios unknown to the particular CAV but experienced by other CAVs*.

For example, as shown in Fig. 1(a), the CAVs participating in FL can learn from each other to operate in a wide range of scenarios, such as crash, traffic jam, and road work areas. Moreover, the FL process is naturally privacy-preserving as the CAVs do not share their local data, e.g., the history trajectory, location, and speed information. To reap all these benefits, we need to address two challenges. First, due to the CAV's mobility and uncertainty of wireless channels, the participation of CAVs in the FL process will vary over time and a good training performance can be challenging to guarantee. Second, because of the unbalanced and non-independent and identically distributed (non-IID) local data across CAVs, the data quality will vary among CAVs and such diverse data quality can impact the FL performance. An effective FL framework for the CAV's controller design must solve these two challenges.

The *main contribution* of this paper is a novel dynamic federated proximal (DFP) framework that can address the aforementioned challenges of FL-based controller design, in presence of CAV mobility, uncertainty of wireless channels, as well as unbalanced and non-IID data. We prove the convergence of the proposed algorithm, under realistic wireless environments, and we theoretically study how the varying CAV participation in the FL process and the presence of diverse local data quality among CAVs affect the convergence of FL. We also extend the results to other case studies, e.g., using the federated averaging (FedAvg) for the controller design and the controller design for electric CAVs with stringent energy constraints. *To the best of our knowledge, this is the first work that develops an FL framework to optimize the autonomous controller design for CAVs, under realistic wireless environments.* Using real data traces, we show that our proposed framework can generate a more accurate speed compared to the controller solely trained by the local data. Furthermore, simulation results show that the proposed algorithm outperforms other popular FL algorithms, i.e., FedAvg [8] and federated proximal (FedProx) algorithms [9], when designing the autonomous controller for CAVs.

II. System Model

Consider a set $\mathcal{N}$ of N CAVs moving along the roads, as shown in Fig. 1(a). To guarantee the safety and achieve the target movement, CAVs will perceive their surrounding environment and accordingly adjust the controller decisions. By collaboratively training the controller via FL, CAVs can automatically change their control parameters, execute the control decisions, and become adapted to the local traffic. Here, we introduce the controller, communication, and learning models used for the FL-based autonomous controller design.



(a) (b)

Fig. 1. Illustration of our system model. The traffic model is presented in (a) where green triangles and red squares, respectively, represent CAVs that do and do not participate in the FL process. The adaptive controller and learning models are shown in (b).

A. Adaptive Longitudinal Controller Model

To perceive the surrounding environment, the CAVs will use sensors and communicate wirelessly with nearby objects, e.g., other CAVs. This perception can enable the longitudinal controller of each CAV to automatically adjust its acceleration or deceleration and maintain a safe spacing and target speed. Here, due to the simplicity and easy implementation, we use the PID controller to control the longitudinal movement of CAVs, and the acceleration $u_n(t)$ of vehicle $n \in \mathcal{N}$ at sample t is defined in a discrete-time manner as follows [10]:

$$u_n(t) = u_n(t-1) + (K_{n,p} + K_{n,i}\Delta t + \frac{K_{n,d}}{\Delta t})e_n(t) + (-K_{n,p} - \frac{2K_{n,d}}{\Delta t})e_n(t-1) + \frac{K_{n,d}}{\Delta t}e_n(t-2), \quad (1)$$

where $K_{n,p}, K_{n,i}$, and $K_{n,d}$ are non-negative coefficients corresponding to the proportional gain, integral time constant, and derivative time constant used by the PID controller in CAV n. Δt is the sampling period and $e_n(t) = v_{n,r}(t) - v_{n,a}(t)$ captures the difference between the target reference speed $v_{n,r}(t)$ and the actual speed $v_{n,a}(t)$ at sample t. Note that the target reference speed is decided by the motion planner in the CAV based on the environmental perception[1].

According to (1), we can calculate the actual speed at sample $t+1$ as $v_{n,a}(t+1) = v_{n,a}(t) + u_n(t)\Delta t$ and the distance traversed between samples t and $t+1$ as $d_{n,p} = \frac{v_{n,a}(t+1)+v_{n,a}(t)}{2}\Delta t$. It is clear that achieving the target speed and safe spacing will depend on the control parameter setting of each CAV's PID controller, and it is imperative to adjust these control parameters adaptively to deal with varying traffic dynamics and road conditions. In particular, instead of using a time-consuming manual tuning for the control parameters $K_{n,p}, K_{n,i}$, and $K_{n,d}, n \in \mathcal{N}$, we assume that the CAV will use an adaptive PID controller enabled by an artificial neural network (ANN)-based auto-tuning unit, as shown in Fig. 1(b). In this case, in order to adapt to various traffic conditions, the CAVs will train the auto-tuning unit using its own history of speed data and then adjust the control parameters accordingly.

[1]As the motion planner design has been extensively studied by the prior art and is not the main scope of this work, we omit details about the process of choosing the target speed and we refer readers to [11].

B. FL Model

The ANN based auto-tuning unit in Fig. 1(b) can adaptively tune the PID control parameters to achieve the target speed. However, the CAV's local training data, i.e., the speed history, is constrained by the onboard memory of the CAV, and thus, is limited to a few of traffic scenarios. For example, vehicles on the highway will mostly drive with a high speed, whereas in urban settings, the speed of CAVs will vary constantly in a lower speed region due to frequent stops and accelerations. Hence, by solely training the local data for the auto-tuning unit, the controller can only be used in limited traffic scenarios. To this end, we can use the wireless connectivity of CAVs to build a cooperative training framework, i.e., FL, among multiple CAVs for the controller design.

Here, we consider CAVs will involve in an FL process to collaboratively train the ANN auto-tuning units for the adaptive controller design. In particular, the BS, operating as a parameter server, will first generate an initial global ANN model parameter $\boldsymbol{w}_0$ for the auto-tuning unit and send it wirelessly to all CAVs over a broadcast downlink channel. Then, in the first communication round, all CAVs will use the received model parameters $\boldsymbol{w}_0$ to independently train the model based on their local data for I iterations. In the uplink, the CAVs transmit their trained model parameters to the BS. Next, the BS will aggregate all the received local model parameters to update the global model parameters which is then sent to all CAVs. This FL process is repeated over uplink-downlink channels and the local and global ANN models are sequentially updated. Ultimately, the ANN model parameters used by the CAVs will converge to the optimal model as follows [8]:

$$\underset{\boldsymbol{w}^{(1)},\ldots,\boldsymbol{w}^{(N)}\in\mathbb{R}^d}{\arg\min} \sum_{n=1}^{N}\sum_{i=1}^{s_n}\frac{1}{s_N}f_n(\boldsymbol{w}^{(n)},\xi_i), \tag{2}$$

$$\text{s.t.} \quad \boldsymbol{w}^{(1)}=\boldsymbol{w}^{(2)}=\ldots=\boldsymbol{w}^{(N)}=\boldsymbol{w}, \tag{3}$$

where $s_N=\sum_{n\in\mathcal{N}}s_n$ is the size of the entire training data of all CAVs with s_n being the size of the local data at CAV n. $f_n(\boldsymbol{w}^{(n)},\xi_i)$ captures the loss function of CAV n when using the ANN model parameters $\boldsymbol{w}^{(n)}$ in the auto-tuning unit for the selected data ξ_i. Note that, the loss function plays a pivotal role in determining the performance of the trained auto-tuning unit. The loss function used for the controller design can be either convex [12] or non-convex [13]. We assume $f(\boldsymbol{w})$ to be the value of objective function (2) when $\boldsymbol{w}^{(n)}=\boldsymbol{w}, n\in\mathcal{N}$.

When training the local ANN models at CAVs, we can calculate the computing delay as $t_{n,\text{comp}}=I\frac{\bar{s}c}{\phi_n}$, where $\bar{s}$ is the size of randomly selected data at each iteration, c is the number of computing cycles needed per data bit, and ϕ_n is the frequency of the central processing unit (CPU) clock of CAV $n\in\mathcal{N}$. However, due to the mobility of CAVs and wireless fading channels, it is possible that not every CAV in the set $\mathcal{N}$ can transmit the trained model parameters to the BS within the duration $\bar{t}$ of each communication round. With this in mind, next, we present the communication model used to determine whether the locally trained model at a particular CAV can be used in the model aggregation.

C. Communication Model

For the uplink from local CAVs to the BS, we consider an orthogonal frequency-division multiple access (OFDMA) scheme where each CAV in set $\mathcal{N}$ will use a unique and orthogonal resource block to transmit the trained ANN model parameters to the BS. The uplink data rate from CAV $n\in\mathcal{N}$ to the BS will be

$$r_n=B\log_2\left(1+\frac{P_nh_nd_n^{-\alpha}}{\rho_n+BN_0}\right), \tag{4}$$

where B is the bandwidth of each orthogonal resource block, P_n is the transmit power of CAV n, and h_n denotes the Rayleigh fading channel gain. Moreover, d_n is the distance between CAV n and the BS, α is the path-loss exponent, and N_0 is the noise power spectral density. In addition, $\rho_n=\sum_{j\notin\mathcal{N}}P_jh_jd_j^{-\alpha}$ is the received interference power generated by other CAVs outside of set $\mathcal{N}$ and sharing the same resource block with CAV n. From (4), the uplink transmission delay for CAV $n\in\mathcal{N}$ can be calculated as $t_{n,\text{comm}}=\frac{s(\boldsymbol{w}^{(n)})}{r_n}$, where $s(\boldsymbol{w}^{(n)})$ is the size of the data packet that depends on the trained model parameters $\boldsymbol{w}^{(n)}$ transmitted by CAV n.

In the downlink, since the BS can have a much higher transmit power and larger bandwidth for the broadcast channel, the downlink transmission delay is considered to be negligible compared to the uplink transmission delay. In addition, given the higher computing power of BSs, the computing delay at the BS can be ignored in contrast to its counterpart at CAVs. Hence, to identify whether the local learning model update from CAV $n\in\mathcal{N}$ can be used for the model aggregation in the BS, we can compare the time for uplink transmission and local computing at the CAV with the duration $\bar{t}$ of the communication round. In this case, the probability that CAV $n\in\mathcal{N}$ participates in FL (the locally trained model at CAV n is used in the model aggregation) can be thereby calculated as $p_n^t=\mathbb{P}(t_{n,\text{comp}}+t_{n,\text{comm}}\leq\bar{t})$.

When designing the FL framework for the CAV's controller design, it is important to explicitly consider the following two factors. The first factor is that the BS can only aggregate a subset $\mathcal{N}_t\subseteq\mathcal{N}$ to update the global model at each communication round. This is because the mobility of CAVs and uncertainty of wireless fading channels make the participation probability p_n^t, $n\in\mathcal{N}$, to be a time-varying parameter that is different for each CAV. It will be challenging to guarantee a good training performance for the controller design when the participation of CAVs in the FL process varies over time [7]. The second factor is the fact that the distribution and size of the local dataset can largely vary among CAVs because of the heterogeneity of the CAVs' local data (generated over different trajectories, traffic, and road incidents). As such, the local datasets at different CAVs will be unbalanced and non-IID distributed. This non-IID and unbalanced local data can adversely affect the convergence of the FL model [9]. To sum up, when designing the FL framework for a CAV, we must account for the varying CAV participation in FL and the diverse local data quality across CAVs.

III. Dynamic Federated Averaging Algorithm for CAV Controller Design

To consider the aforementioned two factors in FL, we now propose a new DFP algorithm customized for the autonomous controller design of CAVs. In particular, we study the impact of the varying participation of CAVs in the FL process and non-IID and unbalanced data on the learning model convergence. We will first introduce the proposed DFP algorithm and then study its convergence performance.

A. Proposed Dynamic Federated Averaging Algorithm

The proposed learning algorithm is summarized in Algorithm 1. In particular, instead of solely optimizing the loss function using the whole training dataset at the CAVs, we assume that the CAVs will run I iterations of stochastic gradient descent (SGD) at each communication round. In each iteration of SGD, the model update at CAV $n \in \mathcal{N}$ will solve the following optimization problem that minimizes the sum of the loss of a randomly selected local training sample $\xi \in \mathcal{S}_n$ and an L_2 regularizer:

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^n} f_n(\boldsymbol{w},\xi) + \frac{\gamma_t}{2}||\boldsymbol{w}-\boldsymbol{w}_t||^2, \xi \in \mathcal{S}_n, \tag{5}$$

where γ_t is the coefficient for the regularizer and $\boldsymbol{w}_t$ captures the received learning model parameters from the BS at communication round t. The purpose of introducing the L_2 regularizer is to guarantee that the trained model parameters $\boldsymbol{w}$ of CAV $n \in \mathcal{N}$ will be close to $\boldsymbol{w}_t$, reducing the variance introduced by the non-IID and unbalanced data. In particular, after I iterations of SGD at the communication round t, we can obtain the trained model parameters of CAV n as follows:

$$f_n(\boldsymbol{w}_{t+1,I}^{(n)}) = \boldsymbol{w}_t + \eta_t \sum_{i=0}^{I-1} \left(\nabla f_n(\boldsymbol{w}_{t,i}^{(n)}, \xi_i) + \gamma_t(\boldsymbol{w}_{t,i}^{(n)} - \boldsymbol{w}_t) \right).$$

Then, due to the uncertainty of wireless channels and CAVs' mobility, the BS will aggregate the trained model parameters from a subset $\mathcal{N}_t$ of N_t CAVs that is able to finish local computing and communication within time constraint $\bar{t}$.

B. Convergence of the Proposed DFP Algorithm

To prove the convergence of the proposed scheme in Algorithm 1, we make the following standard assumptions:

- The gradient $\nabla f_n(\boldsymbol{w}), n \in \mathcal{N}$, is uniformly Lipschitz continuous with positive parameter L.
- The variance of SGD with respect to the full gradient descent is upper bounded for CAV $n \in \mathcal{N}$ and $\boldsymbol{w} \in \mathbb{R}^d$, which is given by $\mathbb{E}_{\xi\in\mathcal{S}_n}||\nabla f_n(\boldsymbol{w},\xi) - \nabla f_n(\boldsymbol{w})||_2^2 \leq \sigma^2, \forall n \in \mathcal{N}, \forall \boldsymbol{w} \in \mathbb{R}^d$, where σ^2 is the upper bound.

Both assumptions are commonly used by many literature, like [14]. The first constraint can be satisfied by some popular loss functions used in control theory, such as the squared error loss function. The second constraint is often adopted in stochastic optimization where the gradient estimator is always assumed to have a bounded variance. Using these two assumptions, we can bound the expected loss function at communication round $t+1$ as shown by the following theorem.

Theorem 1. Given that the BS sends the global learning model parameters $\boldsymbol{w}_t$ to all CAVs at communication round t, an upper bound for the expected loss function at communication round $t+1$ can be written as

$$\mathbb{E}_{\xi,n}(f(\boldsymbol{w}_{t+1})) \leq f(\boldsymbol{w}_t) - \frac{(\eta_t+\gamma_t\eta_t)\sum_{n=1}^N p_n^t s_n^2 I||\nabla f_n(\boldsymbol{w}_t)||_2^2}{2s_N\sum_{j=1}^N p_j^t s_j}$$
$$+\left(\frac{\eta_t}{2s_N}L\eta_t^2I^2 + \frac{\eta_t\gamma_t}{2s_N}(I+I^2(1+\eta_t)^2) + L\eta_t^2 I\right)\frac{\sum_{n=1}^N p_n^t s_n^2}{\sum_{j=1}^N p_j^t s_j}\sigma^2, \tag{6}$$

if the following two conditions are satisfied:

$$L^2\eta_t^2 I^2 + \gamma_t I^2(1+\eta_t)^2 + 2s_N L\eta_t I \leq 1, \tag{7}$$

$$L^2\eta_t^2\gamma_t I^2 + \gamma_t^2\eta_t^2 I^2 + 2s_N\eta_t\gamma_t L I \leq 1, \tag{8}$$

where $p_n^t = \exp\left(-\frac{I_n+BN_0}{P_n d_n^{-\alpha}}\left(2^{\frac{s(\boldsymbol{w}_{n,t})}{B\left(\Delta t - I\frac{\bar{s}c}{\phi_n}\right)}} - 1\right)\right)$.

Proof: The proof is provided in the journal extension [15]. ■

Algorithm 1 Dynamic Federated Proximal (DFP) Algorithm

Iutput: $\mathcal{N}$, $\mathcal{N}_t$, $\mathcal{S}_n$, η_t, $\boldsymbol{w}_0$, I, u_t, γ_t, $s_n, n = 1, ..., N$

Output: ANN-based auto-tuning unit for the CAV's controller

for $t = 0...$ **do**

1. The BS sends $\boldsymbol{w}_t$ to all N CAVs.
2. CAV $n \in \mathcal{N}$ updates its trained model parameters $\boldsymbol{w}_t$ for I iterations of SGD on the local dataset $\mathcal{S}_n$ with a step size as η_t of (5) and obtain $\boldsymbol{w}_{t+1,I}^{(n)}$ which will be sent to the BS.
3. Due to the mobility and time-varying channels, the BS can only aggregate the trained model parameters from a subset $\mathcal{N}_t$ of N_t CAVs and update the global model parameters as $\boldsymbol{w}_{t+1} = \sum_{n\in\mathcal{N}_t}\frac{s_n}{s_{N_t}}\boldsymbol{w}_{t+1,I}^{(n)}$ with $s_{N_t} = \sum_{n\in\mathcal{N}_t} s_n$.

end

From Theorem 1, we can calculate how much the total loss decreases between two consecutive communication rounds and determine the speed with which the model converges to the optimal auto-tuning model in (2). In particular, as observed from Theorem 1, the convergence speed depends on the participation probability p_n^t, $n \in \mathcal{N}$, captured by the mobility of CAVs and the quality of wireless fading channel. In addition, to identify how the participation of a particular CAV in FL can impact the convergence in Theorem 1, we also need to consider the size and distribution of local data at CAVs. To do so, in the following corollary, we will first mathematically define the local data quality of CAVs and study the impact of local data quality on the convergence of learning models.

Corollary 1. When the local data quality of CAV $n \in \mathcal{N}$ is defined as $\beta_n = s_n^2[\left(\frac{\eta_t}{2s_N} + \frac{\gamma_t\eta_t}{2s_N}\right)I||\nabla f_n(\boldsymbol{w}_t)||_2^2 - \left(\frac{\eta_t}{2s_N}L\eta_t^2I^2 + L\eta_t^2 I\right)\sigma^2 + \frac{\eta_t\gamma_t}{2s_N}(I+I^2(1+\eta_t)^2)\sigma^2]$, the set $\mathcal{N}$ can be divided into two subsets $\mathcal{N}_{(1)}$ and $\mathcal{N}_{(2)}$ with the negative and positive data quality, respectively. In this case, the results in (6) can be simplified as

$$f(\boldsymbol{w}_t) - \mathbb{E}_{\xi,n}(f(\boldsymbol{w}_{t+1})) \geq \frac{\sum_{n\in\mathcal{N}_{(1)}} p_n^t\beta_n}{\sum_{j=1}^N p_j^t s_j} + \sum_{n\in\mathcal{N}_{(2)}}\frac{p_n^t\beta_n}{s_N}.$$

Proof: The proof is provided in the journal extension [15]. ■

From Corollary 1, it is clear that the participation of CAVs within the subset $\mathcal{N}_{(1)}$ in FL will impede the FL convergence while the CAVs in subset $\mathcal{N}_{(2)}$ will improve the FL convergence. In other words, depending on the value of data quality

$\beta_n, n \in \mathcal{N}$, the convergence gain contributed by c
CAVs can be negative or positive. In the following cc
we also extend Theorem 1 to the case where the vanilla
is used for the autonomous controller design.

Corollary 2. When using the vanilla FedAvg algorithm
L_2 regularizer in each SGD, we can replace $\gamma_t = 0$ in T
1 and obtain the following upper bound for the expected loss:

$$\mathbb{E}_{\xi,n}(f(\boldsymbol{w}_{t+1})) \leq f(\boldsymbol{w}_t) - \frac{\eta_t}{2s_N} \frac{\sum_{n=1}^{N} p_n^t s_n^2 I ||\nabla f_n(\boldsymbol{w}_t)||_2^2}{\sum_{j=1}^{N} p_j^t s_j} + \left(\frac{\eta_t}{2s_N} L\eta_t^2 I^2 + L\eta_t^2 I \right) \frac{\sum_{n=1}^{N} p_n^t s_n^2}{\sum_{j=1}^{N} p_j^t s_j} \sigma^2,$$

if $L^2\eta_t^2 I^2 + 2s_N L\eta_t I \leq 1$.

By comparing Theorem 1 and Corollary 2, we can prove that, when both constraints (7) and (8) are satisfied, the proposed DFP algorithm can achieve a smaller upper bound for the expected loss than the vanilla FedAvg algorithm. In other words, the proposed DFP can achieve a faster convergence for the controller design in comparison to the FedAvg algorithm, leading to a quick adaption to the traffic dynamics for CAVs.

IV. Simulation Results

To show the performance of the proposed DFP algorithm, we use two real datasets: The Berkeley deep drive (BDD) data [16] and the dataset of annotated car trajectories (DACT) [17]. BDD data is a large-scale driving video dataset with extensive annotations for heterogeneous tasks, and such dataset is collected under diverse geographic, environmental, and weather conditions across the United States. DACT data is a collection of trajectories collected in the city of Columbus, Ohio, where each trajectory records more than 10 minutes of driving data and can be divided into multiple segments annotated by the operating pattern, like speed-up and slow-down. In terms of the traffic model, we consider a 2 km $\times$2 km square area with 20 lanes randomly located around the center of the square area. When using BDD data and the DACT data, we assume that CAVs are randomly assigned to these 20 lanes and all the training data is randomly split among CAVs. Similar to [18], the CAVs' velocity is determined by the headway distance to the preceding CAVs. Other simulation parameters include $\eta = 0.01, \gamma = 0.1, I = 20, P = 0.6$ W, $\Delta t = 1$ s, $\kappa = 10^{-28}$, $c = 10^3$, $\phi = 10^9$ cycles/s, and $N_0 = -174$ dBm/Hz.

Fig. 2 shows the velocity tracking performance comparison between the autonomous controllers solely trained by the local data (i.e., smooth slow-down) and trained by our proposed DFP algorithm under different traffic scenarios. In this simulation, we consider three traffic scenarios from the DACT dataset. In particular, we choose a use case with a dramatic speed decline to represent a harsh brake in a traffic accident, the speed variations around zero as the stop-and-go traffic in a congestion, and the change of the average speed as the speed limit changes in a roadwork zone. As shown in Fig. 2, the controller trained by our proposed DFP algorithm can accurately execute the control decisions and track the target speed under all three traffic scenarios. However, when using the

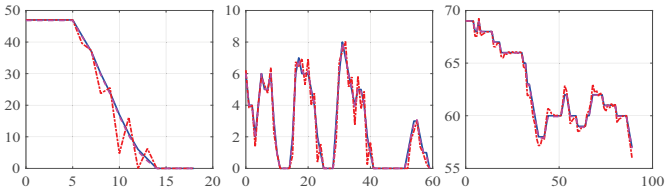


Fig. 2. Velocity variations over different traffic scenarios where x-axis is the time with unit (s) and y-axis is the velocity with unit (miles/hour). The figures from left to right, respectively, refer to harsh brake in a traffic accident, stop-and-go traffic in a congestion, and speed limit changes in a work zone. The blue solid line, magenta dash line, and red dash-dot line, respectively, denote target reference speed, actual speed trained by the DFP algorithm, and actual speed trained only by the local data.

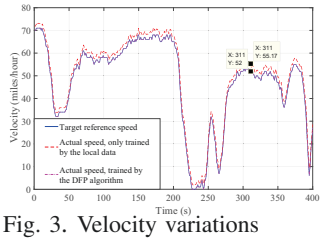


Fig. 3. Velocity variations over time.

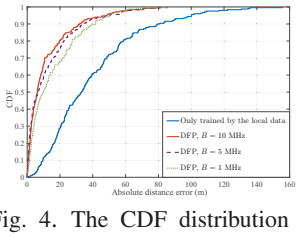


Fig. 4. The CDF distribution of absolute distance errors.

controller trained with the local data, we can face large speed variations around the target values. For example, as shown in the left plot of Fig. 2, to achieve a harsh brake, the controller trained by the local data will generate sequential deceleration and acceleration instead of a constant deceleration as done by the controller trained by our proposed DFP algorithm. In the traffic congestion and roadwork zone in Fig. 2, the controller trained by the local data will have a more frequent switch between acceleration and deceleration than the target speed traces, adversely impacting the driving experience of the passengers. Also, in Fig. 2, the controller trained by the local data can make aggressive deceleration and acceleration and such behaviors will not only increase the CAVs' maintenance costs, but it will also endanger following or preceding CAVs especially when the spacing is small.

Fig. 3 shows the velocity tracking performance comparison between the autonomous controllers solely trained by the local data (i.e., smooth speed-up) and trained by our proposed DFP algorithm. In this simulation, the trajectory data in the DACT dataset is randomly assigned to the CAVs. Fig. 3 shows that the DFP-based controller design can accurately track the target velocity over time. However, the actual velocity generated by the controller trained with local data can deviate from the target value. In particular, at time $t = 311$ s, the error between the actual and target velocities can be as large as 3.17 miles/hour (1.42 meters/second), violating the commonly used two design criteria for the vehicle's controller, i.e., 0.5 meters/second error upper bound [19] and 5% maximal allowable error [20]. Hence, the autonomous controller based on the proposed DFP algorithm outperforms the baseline scheme that solely relies on the local data for training.

Fig. 4 shows the cumulative distribution function (CDF) when the controllers tracks the DACT dataset. In particular, the

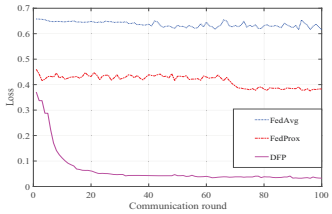


Fig. 5. Comparison between the proposed DFP, FedAvg, and FedProx algorithms.

autonomous controllers are trained, respectively, by local data and by our proposed DFP algorithm with different bandwidth. Also, the absolute distance error is calculated by the absolute difference between the target distance in the DACT dataset and the actual distance traversed by the CAV with the designed controller at the end of each trajectory. As observed from Fig. 4, the controller trained by the proposed DFP algorithm yields a much smaller distance error compared with the case in which the CAVs only use their local data to train the controller model. In particular, with a 0.90 probability, the controller solely trained with local data will generate an absolute distance error of less than 80 m, two times larger than the error resulting from the DFP-based autonomous controller. Moreover, as shown in Fig. 4, for a larger bandwidth, the proposed DFP-based controller design will more likely yield a smaller distance error. For example, when the bandwidth $B = 10$ MHz, the probability that the distance error generated by DFP-based controller remains below 20 m is around 0.80, while the counterpart for the case with a bandwidth $B = 1$ MHz is around 0.68. That is because with a larger bandwidth, more CAVs can meet the time constraint $\bar{t}$ and participate in the FL, leading to a better training performance. As shown in Figures 2-4, it is clear that the autonomous controller based on the proposed DFP algorithm outperforms the baseline scheme that solely relies on the local data for training.

Fig. 5 compares the proposed DFP with the vanilla FedAvg [8] and FedProx [9]. To test the ability of dealing with unbalanced and non-IID data for these three algorithms, we choose a larger BDD dataset. As observed from Fig. 5, when faced with unbalanced and non-IID training data, FedAvg and FedProx fail to converge near zero loss over 100 communication rounds. In particular, after 100 communication rounds, the loss values for the vanilla FedAvg and FedProx are near, respectively, 0.62 and 0.38. However, as shown in Fig. 5, our proposed DFP algorithm only needs around 20 communication rounds (i.e., 0.2 s) to achieve convergence, much faster than the counterparts of FedAvg and FedProx. Hence, using the proposed DFP algorithm, the CAV can quickly adapt to the traffic dynamics and correctly track the target speed.

V. Conclusions

In this paper, we have developed an FL framework to enable collaborative training of the autonomous controller model among CAVs. In particular, we have proposed a new DFP algorithm for the FL that can account for the varying participation of CAVs in FL process as well as diverse data quality across CAVs. We have performed a rigorous theoretical convergence analysis for the proposed algorithm and have explicitly studied the impact of CAVs' mobility, uncertainty of wireless channels, as well as unbalanced and non-IID local data on the overall convergence performance. Simulation results from using the real traces have shown that the autonomous controller designed by the proposed algorithm can track the target speed more accurately than the adaptive controller trained by the local data and the FDA algorithm can lead to a better controller in comparison to the FedAvg and FedProx algorithms. As future extension of the proposed approach, DFP algorithm can be studied for lateral controller design and MPC design in CAVs.

References

[1] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. of IEEE Intelligent Vehicles Symposium*, Seoul, South Korea, Jun. 2015.

[2] M. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, Sept. 2011.

[3] K. Nam, Y. Hori, and C. Lee, "Wheel slip control for improving traction-ability and energy efficiency of a personal electric vehicle," *Energies*, vol. 8, no. 7, pp. 6820–6840, Jul. 2015.

[4] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, May 2020.

[5] S. Wakitani, T. Yamamoto, and B. Gopaluni, "Design and application of a database-driven PID controller with data-driven updating algorithm," *Industrial & Engineering Chemistry Research*, vol. 58, no. 26, pp. 11 419–11 429, May 2019.

[6] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *Proc. of ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Williamsburg, VA, USA, Mar. 2018.

[7] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Communications Magazine, Special Issue on Communication Technologies for Efficient Edge Learning*, to appear, 2020.

[8] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[9] T. Li, A. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. of ACM Conference on Machine Learning and Systems*, Austin, TX, USA, Mar. 2020.

[10] L. D. S. Coelho and D. L. D. A. Bernert, "An improved harmony search algorithm for synchronization of discrete-time chaotic systems," *Chaos, Solitons & Fractals*, vol. 41, no. 5, pp. 2526–2532, Sept. 2009.

[11] B. Paden, M. Čáp, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[12] Q. Song, J. Spall, Y. Soh, and J. Ni, "Robust neural network tracking controller using simultaneous perturbation stochastic approximation," *IEEE Transactions on Neural Networks*, vol. 19, no. 5, pp. 817–835, May 2008.

[13] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, Apr. 2014.

[14] L. Bottou, F. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, May 2018.

[15] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road: Autonomous controller design for connected and autonomous vehicles," *arXiv preprint arXiv:2102.03401*, 2021.

[16] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Jun. 2020.

[17] S. Moosavi, B. Tehrani, and R. Ramnath, "Trajectory annotation by discovering driving patterns," in *Proc. of ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, Redondo Beach, CA, USA, Nov. 2017.

[18] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7907–7922, Nov. 2019.

[19] S. Xiong, H. Xie, K. Song, and G. Zhang, "A speed tracking method for autonomous driving via adrc with extended state observer," *Applied Sciences*, vol. 9, no. 16, pp. 1–21, Aug. 2019.

[20] G. Tagne, R. Talj, and A. Charara, "Higher-order sliding mode control for lateral dynamics of autonomous vehicles, with experimental validation," in *Proc. of IEEE Intelligent Vehicles Symposium*, Gold Coast, QLD, Australia, Jun. 2013.