Road Supervised Federated Learning with Bug-Aware Sensor Placement

Jianjun Chen[1,2,†], Shuai Wang[3,†], Chenguang Liu[4], Derrick Wing Kwan Ng[5], *Fellow, IEEE*, Chengzhong Xu[6], *Fellow, IEEE*, Qi Hao[1,*], and Haiyan Lu[2,*]

***Abstract*— Federated learning (FL) emerges as a promising solution to enhance autonomous driving (AD) models against out-of-distribution (OOD) data. However, OOD instances often lack labels, rendering conventional FL approaches less effective in AD. This paper proposes road-supervised FL (RSFL), which leverages road sensors' perception results to annotate vehicle sensors' data, providing a fresh perspective on data annotations for FLAD systems. To get deeper insights into RSFL, the information gain of annotating objects with road sensors is derived by leveraging the expected entropy reduction. Furthermore, a bug-aware sensor placement (BASP) algorithm is developed which strategically reduces (increases) the number of sensors in low (high) complexity scenarios. This is in contrast to traditional sensor placements where sensing coverage or road topology is the only consideration. It is shown that BASP approximately maximizes the information gain brought by road supervision. Experiments confirm the superiority of the proposed RSFL framework and BASP algorithm.**

***Index Terms*— Autonomous vehicle, federated learning**

I. Introduction

Detecting out-of-distribution (OOD) data in open scenarios holds significant importance for autonomous driving (AD), since OOD data encapsulates rare cases in driving scenarios and contributes to the enhancement of model generalization [1], [2]. Federated learning AD (FLAD), which updates deep neural networks (DNNs) in a distributed manner whenever OOD data is encountered [3]–[5], is an effective solution to robustify DNNs against OOD data. However, in contrast to existing FL applications (e.g., image classification [6]) where manual data annotation is available, FLAD often needs to operate on *unlabeled* data, due to its high-privacy and high-mobility nature [3]. Consequently, the vanilla FL approaches [6], [7] based on supervised learning become ineffective in AD.

To address the above problem, emerging FL approaches [8], [9] proposed model average (e.g., exponential moving average (EMA) [8]) and logit average (e.g., ensemble attention distillation (EAD) [9]) to generate pseudo labels. These works fail in exploiting the geometric relationship among different data frames. To this end, multi-view fusion distillation (MVFD) has been proposed for generating pseudo labels, e.g., feature-level MVFD [10], box-level MVFD [4]. Nonetheless, leveraging the consensus between vehicles (i.e., fusion) could potentially lead to higher bias.

This paper proposes road supervised FL (RSFL), categorized under the MVFD-type solution, to overcome the bias issue by introducing roadside infrastructures (RSIs). However, RSFL needs to integrate road annotation/placement and FL features for joint optimization, for which the existing MVFD FL algorithms [4], [10] become inefficient, as they ignore the inter-dependency between low-level road designs and high-level FL services. There also exist multi-view fusion and road sensor placement approaches [11]–[14] for non-FL scenarios. While these methods are effective for traffic flow monitoring, they are inefficient for RSFL as they ignore the requirements of FL services.[1]

To satisfy the new requirements of RSFL, we first propose a bug-aware road labeling (BARL) algorithm that generates pseudo labels for the false positives, false negatives, and inaccurate boxes. Then, to gain insights into RSFL with BARL, the information gain of annotating the objects with a road sensor with respect to its placement location is derived by leveraging the *expected entropy reduction*. As such, the problem of maximizing the information gain under road placement constraints is formulated, and a bug-aware sensor placement (BASP) algorithm is proposed based on integer programming. It is shown that BASP is equivalent to optimizing a surrogate function that approximates the total information gain. Experiments in high-fidelity unreal engine simulator show that the proposed RSFL with BASP significantly outperforms the pretrain scheme and the RSFL with topology-based sensor placement (TSP). The main contributions are summarized below.

- Propose an RSFL framework with BARL for AD and derive the information gain brought by road supervision.
- Derive a bug-aware sensor placement (BASP) algorithm based on integer programming.
- Prove that BASP is equivalent to optimizing an approximate function of the total information gain.
- Conduct high-fidelity experiments to verify the superiority of the proposed RSFL with BARL and BASP.

This work was supported by the National Key R&D Program of China (No. 2021YFB3300200), the National Natural Science Foundation of China (Grant No. 62371444), the Shenzhen Science and Technology Program (Grant No. RCYX20231211090206005), the Science and Technology Development Fund of Macao S.A.R (FDCT) (No. 0081/2022/A2), Shenzhen Fundamental Research Program(JCYJ20220818103006012, KJZD20231023092600001), and National Natural Science Foundation of China(62261160654).

*Corresponding author: Qi Hao (hao.q@sustech.edu.cn) and Haiyan Lu (Haiyan.Lu@uts.edu.au). † These authors contribute equally.

[1]Southern University of Science and Technology, Shenzhen, China.

[2]University of Technology Sydney, Sydney, Australia.

[3]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.

[4]Durham University, Durham, UK.

[5]University of New South Wales, Sydney, Australia.

[6]University of Macau, Macau, China.

[1]They follow the road topology (i.e., topology-based), or maximize the sensing coverage (i.e., coverage-based), or maximize the number of visible objects (i.e., object-based).

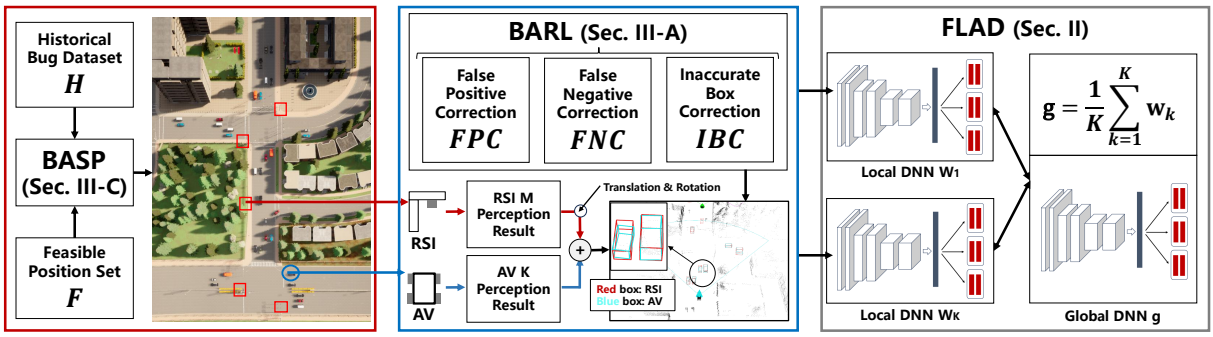


Fig. 1: System architecture of RSFL, which consists of the BASP and BARL modules for FLAD.

II. System Model

We consider an FLAD system with K autonomous vehicles (AVs). The lidar data at the k-th vehicle ($k \in \{1,\ldots,K\}$) at the t-th ($t \in \{1,\ldots,T\}$) lidar time frame, is denoted as ${}^{V}\mathcal{D}_k = \{{}^{V}\mathbf{d}_{k,1}, {}^{V}\mathbf{d}_{k,2}, \cdots\}$, where ${}^{V}\mathbf{d}_{k,t} \in \mathbb{R}^{D_k \times 3}$ is the vector concatenating the coordinates of all points, with D_k being the number of points in each cloud. The DNN parameter vector at the k-th vehicle is $\mathbf{w}_k \in \mathbb{R}^{W_k \times 1}$ with W_k being the dimension of each DNN. The k-th DNN maps ${}^{V}\mathbf{d}_{k,t}$ into a set of bounding boxes

$$ {}^{V}\mathcal{B}_{k,t} = \left\{ {}^{V}\mathbf{b}_{k,t}^{[1]}, {}^{V}\mathbf{b}_{k,t}^{[2]}, \cdots \right\}, \tag{1} $$

where ${}^{V}\mathbf{b}_{k,t}^{[n]}$ represents the n-th object at the t-th lidar frame in the k-th vehicle coordinate system, and its label format is given by

$$ \mathbf{b} = [c, x, y, z, l, w, h, \theta]^T, \tag{2} $$

where c is the category, (x, y, z) is the center position, (l, w, h) is the tuple of length, width, and height, and θ denotes the yaw rotation, respectively. DNN inference at the k-th vehicle can be written as ${}^{V}\mathcal{B}_{k,t} = {}^{V}\Phi_k({}^{V}\mathbf{d}_{k,t}|\mathbf{w}_k)$, where ${}^{V}\Phi_k$ represents the DNN inference function.

The goal of AD perception is to make ${}^{V}\mathcal{B}_{k,t}$ an accurate structured representation of ${}^{V}\mathbf{d}_{k,t}$. However, ${}^{V}\mathcal{B}_{k,t}$ cannot match ${}^{V}\mathbf{d}_{k,t}$ if there exist OOD points in ${}^{V}\mathbf{d}_{k,t}$. Consequently, we need the FLAD procedure for fine-tuning $\mathbf{w}_k$ in a distributed manner. Specifically, the FLAD aims to acquire a global DNN with parameter vector $\mathbf{g}$, by solving the following loss minimization problem:

$$ \min_{\{\mathbf{w}_k\},\mathbf{g}} \underbrace{\frac{1}{\sum_k |{}^{V}\mathcal{D}_k^{\diamond}|} \sum_k \sum_{({}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond}) \in {}^{V}\mathcal{D}_k^{\diamond}} \Theta(\mathbf{g}; {}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond})}_{:=\Lambda(\mathbf{g})} $$

$$ \text{s.t.} \quad \mathbf{w}_1 = \cdots = \mathbf{w}_K = \mathbf{g}, \tag{3} $$

where ${}^{V}\mathcal{B}_{k,t}^{\diamond}$ is the set of pseudo labels[2] obtained from a *teacher model*, $\Theta(\mathbf{g}; {}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond})$ is the loss function corresponding to a single sample $({}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond})$ $(1 \le t \le |{}^{V}\mathcal{D}_k^{\diamond}|)$ in ${}^{V}\mathcal{D}_k^{\diamond} = \{({}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond})\}_t$ with respect to parameter vector $\mathbf{g}$, and $\Lambda(\mathbf{g})$ denotes the global loss function to be minimized.

Now let $\mathbf{g}^{[0]}$ denote the pretrained DNN at the cloud, and let $\mathbf{w}_k^{[i]}(0)$ denote the local DNN parameters at AV k at the beginning of the i-th iteration ($i \ge 0$ and $\mathbf{w}_k^{[i]}(0) = \mathbf{g}^{[i]}$). The FL training of model parameters (i.e., solving (3)) is a distributed and iterative procedure, where each iteration involves the following two steps:

1) The k-th vehicle first minimizes the loss function via the gradient descent approach[3] as

$$ \mathbf{w}_k^{[i]}(\tau+1) = \mathbf{w}_k^{[i]}(\tau) - \frac{\varepsilon}{|{}^{V}\mathcal{D}_k^{\diamond}|} \sum_{({}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond}) \in {}^{V}\mathcal{D}_k^{\diamond}} \nabla\Theta(\mathbf{w}_k^{[i]}(\tau); {}^{V}\mathbf{d}_{k,t}, {}^{V}\mathcal{B}_{k,t}^{\diamond}), \tag{4} $$

where ε is the step-size, $0 \le \tau \le E-1$ (E is the number of local updates) and $\nabla\Theta$ denotes the gradient of Θ.

2) All AVs upload $\{\mathbf{w}_k^{[i]}(E) | \forall k\}$ to the server, which computes an average model

$$ \mathbf{g}^{[i+1]} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k^{[i]}(E), \tag{5} $$

and then broadcast to the AVs for next-round updates.

This completes one federated learning round and we set $i \leftarrow i+1$. The entire procedure stops when $i = I_{\text{FL}}$ with I_{FL} being the number of federated learning rounds.

III. Proposed RSFL Approach

The key to FLAD is to find a proper teacher model such that the pseudo labels in ${}^{V}\mathcal{B}_{k,t}^{\diamond}$ is closer to ground truth. This paper proposes the RSFL approach shown in Fig. 1, which consists of the BARL and BASP modules for the FLAD system presented in Section II. The BARL module leverages roadside infrastructures (RSIs) to annotate data at ego-vehicles. The BASP module determines the positions of RSIs using bug database and integer programming. Below we present the details of the two newly developed modules.

A. Bug Aware Road Labeling

Denote the lidar data at RSI m ($1 \le m \le M$) as ${}^{R}\mathcal{D}_m = \{{}^{R}\mathbf{d}_{m,1}, {}^{R}\mathbf{d}_{m,2}, \cdots\}$, where ${}^{R}\mathbf{d}_{m,t}$ represents a frame of road point cloud. The DNN inference at RSI m at lidar time t is ${}^{R}\mathcal{B}_{m,t} = {}^{R}\Phi_m({}^{R}\mathbf{d}_{m,t}|\mathbf{r}_m)$, where $\mathbf{r}_m \in \mathbb{R}^{R_m \times 1}$ is the DNN parameter vector at RSI.

Now, a direct approach is to label ${}^{V}\mathcal{D}_k$ with ${}^{R}\mathcal{B}_{m,t}$ via the associated rotation and translation matrices between AV

[2]The set of ground truth labels is denoted as ${}^{V}\mathcal{B}_{k,t}^{*}$.

[3]If $|\mathcal{D}_k|$ is large, stochastic gradient descent can be adopted to accelerate the training speed.

Algorithm 1: RSFL with BARL

Input: ${}^{V}\mathcal{D}_k$, ${}^{G}\mathcal{B}_t$, ${}^{V}\mathcal{B}_{k,t}$, $F_{G\to k}$, $F_{k\to G}$, $\mathbf{g}^{[0]}$, I_{FL}, E
Output: $\mathbf{g}^{[I_{\mathrm{FL}}]}$
Function $\mathtt{RS}({}^{G}\mathcal{B}_t, {}^{V}\mathcal{B}_{k,t})$**:**
  Map ${}^{G}\mathcal{B}_t$ to AV k as $\mathcal{C}_{k,t} = F_{G\to k}({}^{G}\mathcal{B}_t)$
  Remove hidden points from $\mathcal{C}_{k,t}$ to acquire $\mathcal{C}^{\diamond}_{k,t}$
  for $\mathbf{c} \in \mathcal{C}^{\diamond}_{k,t}$ **do**
    if $\mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) = 0,\ \forall n$ **then**
      Update ${}^{V}\mathcal{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathcal{B}^{\diamond}_{k,t} \cup \{\mathbf{c}\}$
    if $\delta \le \mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) \le \alpha$ **then**
      Update ${}^{V}\mathcal{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathcal{B}^{\diamond}_{k,t} \setminus \{{}^{V}\mathbf{b}^{[n]}_{k,t}\} \cup \{\mathbf{c}\}$
  Map ${}^{V}\mathcal{B}_{k,t}$ to global as $\mathcal{A}_{k,t} = F_{k\to G}({}^{V}\mathcal{B}_{k,t})$
  Remove hidden points from $\mathcal{A}_{k,t}$ to acquire $\mathcal{A}^{\diamond}_{k,t}$
  for $\mathbf{c} \in \mathcal{A}^{\diamond}_{k,t}$ **do**
    if $\mathrm{IoU}(\mathbf{c}, {}^{G}\mathbf{b}^{[n]}_{t}) = 0$ *for all* n **then**
      Update ${}^{V}\mathcal{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathcal{B}^{\diamond}_{k,t} \setminus \{F_{G\to k}(\mathbf{c})\}$
Function `FL`**:**
  for $k \leftarrow 1$ **to** K **do**
    Execute $\mathsf{RS}({}^{G}\mathcal{B}_t, {}^{V}\mathcal{B}_{k,t})$ to obtain RS dataset ${}^{V}\mathcal{D}^{\diamond}_k = \{({}^{V}\mathbf{d}_{k,1}, {}^{V}\mathcal{B}^{\diamond}_{k,1}), ({}^{V}\mathbf{d}_{k,2}, {}^{V}\mathcal{B}^{\diamond}_{k,2}), \cdots\}$
  for $i \leftarrow 0$ **to** $I_{\mathrm{FL}} - 1$ **do**
    for $k \leftarrow 1$ **to** K **do**
      for $\tau \leftarrow 0$ **to** $E - 1$ **do**
        Update local model using (4)
    Update global model using (5)

k and RSI m. However, RSI m also suffers from occlusion, and its perception performance of a certain object could be even worse than that at AV k. To improve the quality of pseudo labels, we propose to merge the perception results of all RSIs into a global object list ${}^{G}\mathcal{B}_t = \{{}^{G}\mathbf{b}^{[1]}_t, {}^{G}\mathbf{b}^{[2]}_t, \cdots\}$. This is realized by exploiting the late fusion cooperative perception algorithm developed in our previous work [5], [11]. After fusion, the road server executes $\mathsf{BARL}({}^{G}\mathcal{B}_t, {}^{V}\mathcal{B}_{k,t})$ to generate the pseudo label dataset ${}^{V}\mathcal{D}^{\diamond}_k = \{({}^{V}\mathbf{d}_{k,1}, {}^{V}\mathcal{B}^{\diamond}_{k,1}), ({}^{V}\mathbf{d}_{k,2}, {}^{V}\mathcal{B}^{\diamond}_{k,2}), \cdots\}$, which is fed to the FLAD. The entire procedure of BARL is summarized in Algorithm 1.

In particular, for the k-th vehicle at time t, we first initialize ${}^{V}\mathcal{B}^{\diamond}_{k,t} = {}^{V}\mathcal{B}_{k,t}$ and then consider three cases to update ${}^{V}\mathcal{B}^{\diamond}_{k,t}$.

1) **False Negative Correction (FNC)**. We map ${}^{G}\mathcal{B}_t$ to the vehicle coordinate system as $\mathcal{C}_{k,t} = F_{G\to k}({}^{G}\mathcal{B}_t)$, where $F_{G\to k}$ is the function mapping coordinates from global frame to local frame. We then perform view frustum culling and hidden point removal on $\mathcal{C}_{k,t}$ with respect to the FoV of the k-th vehicle, which yields $\mathcal{C}^{\diamond}_{k,t}$. For any box $\mathbf{c} \in \mathcal{C}^{\diamond}_{k,t}$, if $\mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) = 0$ for all n and $\mathbf{c}$ is learnable for the AV[4], then we update ${}^{V}\mathcal{B}^{\diamond}_{k,t} = {}^{V}\mathcal{B}^{\diamond}_{k,t} \cup \{\mathbf{c}\}$.
2) **False Positive Correction (FPC)**. We map ${}^{V}\mathcal{B}_{k,t}$ to the global coordinate system as $\mathcal{A}_{k,t} = F_{k\to G}({}^{V}\mathcal{B}_{k,t})$, where $F_{k\to G}$ is a reverse mapping of $F_{G\to k}$. We then perform view frustum culling and hidden point removal on $\mathcal{A}_{k,t}$ with respect to the FoV of RSIs, which yields $\mathcal{A}^{\diamond}_{k,t}$. For any box $\mathbf{c} \in \mathcal{A}^{\diamond}_{k,t}$, if $\mathrm{IoU}(\mathbf{c}, {}^{G}\mathbf{b}^{[n]}_{t}) = 0$, $\forall n$, where IoU is intersection over union function [11], then we update ${}^{V}\mathcal{B}^{\diamond}_{k,t} = {}^{V}\mathcal{B}^{\diamond}_{k,t} \setminus \{F_{G\to k}(\mathbf{c})\}$.
3) **Inaccurate Box Correction (IBC)**. For any box $\mathbf{c} \in \mathcal{C}^{\diamond}_{k,t}$, if there exists some n such that $\delta \le \mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) < \alpha$, where (δ, α) are predefined thresholds, e.g., we set $(\delta, \alpha) = (0.05, 0.7)$, then we update ${}^{V}\mathcal{B}^{\diamond}_{k,t} = {}^{V}\mathcal{B}^{\diamond}_{k,t} \setminus \{{}^{V}\mathbf{b}^{[n]}_{k,t}\} \cup \{\mathbf{c}\}$.

After iterating all the elements in ${}^{V}\mathcal{B}_{k,t}$ and ${}^{G}\mathcal{B}_t$, the resultant ${}^{V}\mathcal{B}^{\diamond}_{k,t}$ is our desired pseudo label set. By concatenating the results of all time frames, the RS dataset ${}^{V}\mathcal{D}^{\diamond}_k$ at the k-th vehicle is obtained.

B. Information Gain of BARL

To quantify the benefit brought by BARL, we first consider the single-RSI case and evaluate the information gain of annotating the objects with the road sensor. Specifically, consider the i-th object at the k-th vehicle at the t-th frame, and denote its points as $\mathbf{p}_i \in {}^{V}\mathbf{d}_{k,t}$. The associated ground truth box is $\mathbf{b}^{*}_i$, with its format given by (2). However, the generated box $\hat{\mathbf{b}}$ at the AV and box $\mathbf{b}^{\diamond}$ at the RSI involve uncertainties. To quantify their uncertainties, we model the elements in $\hat{\mathbf{b}}$ and $\mathbf{b}^{\diamond}$ as random variables and compute the associated entropy.

For $\hat{\mathbf{b}}_i = [\hat{c}_i, \hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{l}_i, \hat{w}_i, \hat{h}_i, \hat{\theta}_i]^T$, it contains 1 discrete variable (i.e., category $\hat{c}_i$) and 7 continuous variables (i.e., position $(\hat{y}_i, \hat{z}_i, \hat{l}_i)$, size $(\hat{l}_i, \hat{w}_i, \hat{h}_i)$, and orientation $\hat{\theta}_i$). The probability mass function (pmf) of $\hat{c}_i$ is $p_{\hat{c}_i}$. For instance, in a binary classification, $\hat{c}_i = 0$ represents car and $\hat{c}_i = 1$ represents person. Then, the pmf $[p_{\hat{c}_i}(\hat{c}_i = 0), p_{\hat{c}_i}(\hat{c}_i = 1)] = [0.2, 0.8]$ means that the object is classified as a person with a probability of 0.8. On the other hand, since $(\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{l}_i, \hat{w}_i, \hat{h}_i, \hat{\theta}_i)$ are continuous random variables, we model their uncertainties using probability density functions (pdfs), which are given by $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i}, p_{\hat{l}_i}, p_{\hat{w}_i}, p_{\hat{h}_i}, p_{\hat{\theta}_i})$. For instance, for box position, the mean of $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i})$ is the output box and the variance of $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i})$ is the perturbation. Based on the above pmf and pdf models, the entropy of $\hat{\mathbf{b}}_i$ is given by

$$H(\hat{\mathbf{b}}_i) = -\sum_{t} p_{\hat{c}_i}(t)\log(p_{\hat{c}_i}(t)) - \sum_{u\in\{\hat{x}_i,\hat{y}_i,\hat{z}_i,\hat{l}_i,\hat{w}_i,\hat{h}_i,\hat{\theta}_i\}} \int p_u(t)\log(p_u(t))\,dt. \quad (6)$$

Similarly, the entropy of $\mathbf{b}^{\diamond}_i$ is $H(\mathbf{b}^{\diamond}_i)$.

Now we consider two cases.

- If the IoU between $\hat{\mathbf{b}}_i$ and $\mathbf{b}^{\diamond}_i$ is larger than α, then the RSI would agree with the ego-vehicle and would not

[4]A learnable object should contain a sufficient number of points observed by the AV.

change the bounding box. In this case, the entropy is unchanged and we denote this event as $\mathbf{b}_i^+$.

- If the IoU between $\hat{\mathbf{b}}_i$ and $\mathbf{b}_i^\diamond$ is smaller than α, then the RSI would change the bounding box. In this case, the entropy is reduced and we denote this event as $\mathbf{b}_i^-$.

According to [15], the expected information gain (IG) of annotating $\mathbf{p}_i$ by RSI is given by

$$IG_i = P\left(\mathbf{b}_i^-\right)\left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond\right)\right]. \tag{7}$$

The total IG of annotating all data is given by

$$IG_{\text{sum}} = \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{\mathbf{p}_i\in{}^V\mathbf{d}_{k,t}} P\left(\mathbf{b}_i^-\right)\left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond\right)\right]. \tag{8}$$

C. Bug-Aware Sensor Placement

The IG_{sum} in (8) is defined with respect to a single RSI. However, in the multi-RSI case, IG_{sum} would also depend on the sensor placement vector $\mathbf{v} = [v_1, \cdots, v_M]^T \in \{0,1\}^M$ (M is the number of candidate locations), where $v_i = 1$ represents that the i-th position is selected as a placement site and $v_i = 0$ denotes that the i-th position is abandoned. The number of deployed sensors should satisfy $\sum_{m=1}^{M} v_m = L$. Furthermore, since RSIs can only be attached to utility poles, the set of candidate locations, denoted as $\mathcal{F}$ (with $|\mathcal{F}| = M$), is known, and we denote $\mathbf{r}_m = [r_{m,x}, r_{m,y}, r_{m,z}]^T \in \mathcal{F}$ as the location of the m-th feasible position to attach sensors.

Based on the above sensor placement model, we rewrite (8) into its multi-RSI form:

$$\Psi(\mathbf{v}) = \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{\mathbf{p}_i\in{}^V\mathbf{d}_{k,t}} P\left(\mathbf{b}_i^-|\mathbf{v}\right)\left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right)\right] \tag{9}$$

where the labeling probability $P\left(\mathbf{b}_i^-|\mathbf{v}\right)$ and entropy of RSI's detection $H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right)$ are now dependent on $\mathbf{v}$. The problem of maximizing IG for all data with respect to the sensor placement $\mathbf{v}$ is

$$(\mathsf{P0}) \quad \max_{\mathbf{v}} \quad \Psi(\mathbf{v}) \tag{10a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} v_m = L, \quad v_m \in \{0,1\},\ \forall m. \tag{10b}$$

In practice, however, it is challenging solve P0 since $H(\hat{\mathbf{b}}_i)$ and $H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right)$ have no explicit forms. To this end, we propose to approximate $\Psi(\mathbf{v})$ in (10b) using a surrogate function $\Psi'(\mathbf{v})$. In particular, no matter what values $H(\hat{\mathbf{b}}_i)$ and $H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right)$ take, we always have $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right) > 0$ if $\mathbf{b}_i^-$ happens and $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right) = 0$ otherwise. By setting $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^\diamond|\mathbf{v}\right) = G > 0$ for all i, P0 is approximated as

$$(\mathsf{P1}) \quad \max_{\mathbf{v}} \ \underbrace{G\sum_i P\left(\mathbf{d}_i^-|\mathbf{v}\right)}_{:=\Psi'(\mathbf{v})}, \quad \text{s.t. (10b)}. \tag{11}$$

It can be seen that P1 is equivalent to maximizing the expected number of pseudo labels, which can be estimated by Monte-Carlo sampling. Specifically, we deploy the pretrained DNN $\mathbf{g}^{[0]}$ on AVs and test them inside our region of interest. If any AV generates a bug object (including the inaccurate box, false negative, and false positive) within its FoV, the 3D position of this bug $\mathbf{e}_j$ is registered into a database. After a sufficiently long simulation time, the database becomes $\mathcal{H} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots\}$, which contains all registered error items. Therefore, maximizing the expected number of pseudo labels is equivalent to maximizing the cardinality $|\mathcal{Z}|$, where $\mathcal{Z} = \{\mathbf{e} \in \mathcal{H} | \exists m : v_m = 1, \|\mathbf{e} - \mathbf{r}_m\| \le R\}$ and R in meter is the detection range of RSIs.[5]

Algorithm 2: BSAP

Input: $\mathcal{F}$, $\mathcal{H}$, L
Output: Optimal sensor placement vector $\mathbf{v}^*$
Function BASP($\mathcal{F}$, $\mathcal{H}$, L):
  Construct bipartite graph $(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$
  Calculate $E_{m,j}$ for all m, j
  Form the observability matrix $\mathbf{E} \in \{0,1\}^{M\times J}$
  Solve problem (13) and obtain its solution $\mathbf{v}^\diamond$
  Return solution $\mathbf{v}^* = \mathbf{v}^\diamond$

To further derive $\mathcal{Z}$, consider a bipartite graph $(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$, where $\mathcal{V}_1 = \{a_1, \cdots, a_M\}$ is the set of M positions, $\mathcal{V}_2 = \{b_1, \cdots, b_J\}$ is the set of J bug data, and $\mathcal{E}$ is the set of edges with $E_{m,j}$ representing the observability of bug data j from position m. Specifically, if $\|\mathbf{e}_j - \mathbf{r}_m\| \le R$, we set $E_{m,j} = 1$; otherwise, we set $E_{m,j} = 0$. Weights $\{E_{m,j}\}$ are stacked into a matrix $\mathbf{E} = [E_{1,1}, \cdots, E_{1,j}; \cdots; E_{M,1}, \cdots, E_{M,J}] \in \{0,1\}^{M\times J}$. The nonzero elements in the vector $\mathbf{v}^T\mathbf{E}$ represents the expected number of false detections at AVs that could be supervised by the RSIs. That is, $\mathcal{Z} = \|\mathbf{v}^T\mathbf{E}\|_0$. Based on the above derivation, P1 is reformulated as

$$\max_{\mathbf{v}} \quad \|\mathbf{v}^T\mathbf{E}\|_0, \quad \text{s.t. (10b)}. \tag{12}$$

The above problem is a nonconvex integer programming problem due to the nonconvexity of l_0 norm and discontinuity of $\mathbf{v}$. To this end, a slack variable $\mathbf{q} \in \{0,1\}^J$ with $\mathbf{q} \preceq \mathbf{v}^T\mathbf{E}$ is introduced, and the following surrogate problem of (12) is considered:

$$\max_{\mathbf{v},\mathbf{q}} \sum_{j=1}^{J} q_j, \ \text{s.t. (10b)}, \ \mathbf{v}^T\mathbf{E} \succeq \mathbf{q}, \ q_j \in \{0,1\}, \ \forall j. \tag{13}$$

Denoting the optimal solutions to problem (12) and (13) as $\mathbf{v}^*$ and $(\mathbf{v}^\diamond, \mathbf{q}^\diamond)$, respectively, the following proposition is established.

Proposition 1. *The optimal* $\mathbf{v}^*$ *to problem* (12) *and the optimal* $\mathbf{v}^\diamond$ *to problem* (13) *satisfies* $\|\mathbf{v}^{\diamond T}\mathbf{E}\|_0 = \|\mathbf{v}^{* T}\mathbf{E}\|_0$.

Proof: Assume that $\|\mathbf{v}^{\diamond T}\mathbf{E}\|_0 \neq \|\mathbf{v}^{* T}\mathbf{E}\|_0$. Then we can always find a feasible solution $\mathbf{v}'$ to (12) such that $\|\mathbf{v}'^T\mathbf{E}\|_0 < \|\mathbf{v}'^T\mathbf{E}\|_0 \le \|\mathbf{v}^{* T}\mathbf{E}\|_0$. Construct $(\mathbf{v}', \mathbf{q}')$ with $[\mathbf{q}']_j = 0$ if $[\mathbf{v}'^T\mathbf{E}]_j = 0$ and $q'_j = 1$ if $[\mathbf{v}'^T\mathbf{E}]_j \neq 0$, where $[\mathbf{a}]_j$ represents the j-th element of vector $\mathbf{a}$. It can be shown that $(\mathbf{v}', \mathbf{q}')$ is a feasible solution to (13).

[5]It is assumed that the objects are accurately detected within the range of R; otherwise we can always decrease R to make this assumption holds.

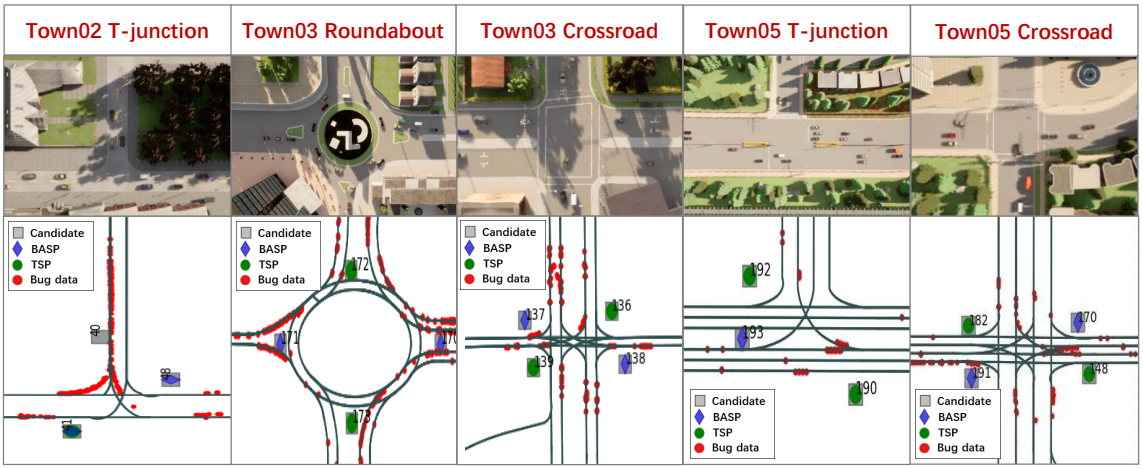


Fig. 2: Comparison between BASP and TSP in 5 different scenarios.

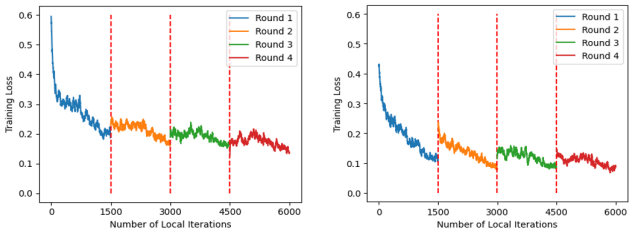
Furthermore, $\sum_{j=1}^{J} q_j' = \|\mathbf{v}'^T\mathbf{E}\|_0 > \|\mathbf{v}'^T\mathbf{E}\|_0 = \sum_{j=1}^{J} q_j^\diamond$. This contradicts to the optimality of $(\mathbf{v}^\diamond, \mathbf{q}^\diamond)$.

Based on **Proposition 1**, we can solve (13) instead of (12). Moreover, problem (13) is an integer linear programming (ILP) problem, which can be optimally solved by off-the-shelf software packages, such as CVXPY [16]. The entire procedure of BASP is summarized in Algorithm 2.

IV. Experiments

We implement the proposed RSFL system and BASP algorithms using Python in the high-fidelity CARLA simulator (driven by unreal engine) [17] on a Ubuntu workstation with a 3.7 GHz AMD Ryzen 9 CPU and an NVIDIA 3090 Ti GPU. We simulate 5 scenarios (i.e., 2 T-juctions, 2 cross roads, and 1 round-about), and their bird-eye-view pictures are shown in the first row of Fig. 2. In the considered scenarios, we have $M = 18$ feasible positions and $L = 9$ RSIs, where the feasible positions are marked as squares in the second row of Fig. 2. To collect the RSFL dataset, we generate $K = 4$ AVs, each equipped with a 64-line lidar and a SECOND DNN model [18] for 3D object detection, to navigate in all these scenarios. Each AV collects 7000 frames of point cloud data at a frequency of 10 Hz, where objects within the range of any RSI are labeled with road detected bounding boxes and are unlabelled otherwise. Note that the pretrained SECOND at all AVs is obtained by training SECOND with 9000 frames in CARLA Town02, Town03, Town05 maps for 50 epochs. All the DNN models are tested on a common dataset with 4000 samples collected in CARLA Town02, Town03, Town05 maps.

We compare the following schemes: 1) **Pretrain**, which directly adopts the pretrained SECOND; 2) **RSFL with TSP** [12], which places RSIs based on the complexity of road topology; 3) **RSFL with BASP**, which places the RSIs by solving (13). First, as seen from the second row of Fig. 2, in contrast to existing TSP that places more RSIs (marked as green circles) at roads with more lanes, the proposed RSFL automatically identifies the critical scenarios and places more RSIs (marked as blue diamonds) in areas with more bug data. As such, the number of false detections (marked as red circles) that can be calibrated by the accurate road detections



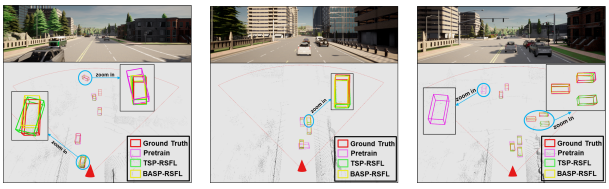
(a) AV1. (b) AV2.

Fig. 3: Training loss versus the number of iterations.

under the BASP scheme is larger than that by TSP. For instance, the BASP places more RSIs at narrow T-junctions (i.e., the first scenario) instead of wide T-junctions (i.e., the forth scenario), which is in shapely contrast to TSP. This is because a narrow road would lead to a high occlusion probability. As for the roundabout scenario (i.e., the second scenario), the positions generated by BASP and TSP are also different, where BASP places two lidars at west and east sites, but TSP places two lidars at north and south sites. Indeed, we assign a higher traffic density to the traffic flow from west to east, and the proposed BASP automatically recognizes such patterns. The above observations imply that the proposed BASP method serves as a better sensor placement strategy for FL than existing methods, producing automatic annotations with a higher probability.

Next, the training loss versus the number of iterations at two AVs is shown in Fig. 3. It can be seen that the global model aggregation is executed every 1500 local iterations. The total number of local iterations is 6000, corresponding to 4 FL rounds. The training losses decrease for both vehicles as the number of iterations or FL rounds increases. This demonstrates the convergence property of the proposed RSFL framework.

Furthermore, the performance of SECOND models trained by different schemes is presented in Fig. 4. The detection results of ground truth, pretrained model, RSFL model and RSFL+BASP model are marked as red, pink, green and yellow, respectively. First, it can be seen from Fig. 4 that the pretrained scheme without FL (marked in pink) generates inaccurate boxes in Fig. 4a, false negatives in Fig. 4b,



(a) Case I. (b) Case II. (c) Case III.

Fig. 4: Detection results. Red, pink, green, and yellow boxes represent results of ground truth, pretrain, TSP-RSFL, and BASP-RSFL, respectively.

		mAP	
		IoU=0.7	IoU=0.5
Crossroad (Town03)	pretrained	48.87%	68.25%
	RSFL+TSP	51.92% (↑ 3.05%)	68.3% (↑ 0.11%)
	RSFL+BASP	51.09% (↑ 2.21%)	70.96% (↑ 2.71%)
T-junction (Town03)	pretrained	49.75%	71.02%
	RSFL+TSP	52.61% (↑ 2.86%)	72.03% (↑ 1.02%)
	RSFL+BASP	52.94% (↑ 3.19%)	76.33% (↑ 5.31%)
Roundabout (Town03)	pretrained	43.07%	71.74%
	RSFL+TSP	47.36% (↑ 4.29%)	73.89% (↑ 2.15%)
	RSFL+BASP	49.38% (↑ 6.31%)	75.69% (↑ 3.95%)
Crossroad (Town05)	pretrained	46.65%	69.73%
	RSFL+TSP	47.12% (↑ 0.47%)	73.73% (↑ 4.00%)
	RSFL+BASP	63.09% (↑ 16.43%)	76.30% (↑ 6.57%)
T-junction (Town05)	pretrained	43.87%	71.13%
	RSFL+TSP	49.61% (↑ 5.74%)	74.73% (↑ 3.60%)
	RSFL+BASP	51.94% (↑ 8.07%)	76.54% (↑ 5.41%)

TABLE I: Comparison of mAPs for different schemes.

and false positives in Fig. 4c. Next, by employing RSFL, the bounding boxes (marked in yellow) become closer to the ground truth in Fig. 4a. The RSFL also corrects the false negative in Fig. 4b, and recovers a false positive in Fig. 4c. Lastly, with the proposed RSFL with BASP, all the objects (marked in red) in all scenarios are successfully detected. This demonstrates the benefits brought by the bug-awareness feature and corroborates the entropy reduction theory described in Section III.

Finally, we compare the mAPs of different schemes. Table I summarises the mAPs at $\text{IoU} = 0.5$ and $\text{IoU} = 0.7$ in cross-road, T-junction, and roundabout scenarios of Town03 and Town05. The experimental results show that even with TSP-RSFL, the mAP is significantly increased compared to the pretrained SECOND, i.e., with up to 5.74% and 4.00% improvements at $\text{IoU}= 0.7$ and $\text{IoU}= 0.5$, respectively. This is because the RSFL fine-tunes the pretrained model using the OOD data and the road supervision at the adversarial scenarios, thus enhancing the model generalization capability. Furthermore, with BASP, the mAP performance is further boosted. The mAP of RSFL with BASP outperforms RSFL with TSP in almost all test scenarios, and leads to an mAP improvement of more than 15% at $\text{IoU}= 0.7$ in the crossroad scenario of Town05. This is because under BASP, the RSI pose is not only determined by the road topology or traffic flows, but also determined by the bug data distribution, which accounts for the learning requirements at AVs in complex urban scenarios.

V. Conclusion

This paper presented an RSFL framework for AVs to learn from unlabeled data in open scenarios. To maximize the expected amount of road-supervised OOD data, we further proposed a BASP algorithm based on graph model and integer optimization. Experiments showed that RSFL fine-tunes the pretrained model towards better generalization. Moreover, the BASP algorithm can enhance automatic road data annotations for FLAD.

References

[1] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, 2020.

[2] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena, "Out-of-distribution detection for automotive perception," in *IEEE Int. Intell. Transport. Syst. Conf.*, 2021, pp. 2938–2943.

[3] S. Wang, C. Li, D. W. K. Ng, Y. C. Eldar, H. V. Poor, Q. Hao, and C. Xu, "Federated deep learning meets autonomous vehicle perception: Design and verification," *IEEE Netw.*, vol. 37, no. 3, pp. 16–25, 2023.

[4] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *IEEE Int. Conf. Robot. Automat.*, 2021, pp. 953–959.

[5] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, 2019.

[6] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, 2021.

[7] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mob. Comput.*, vol. 21, no. 8, pp. 2818–2832, 2020.

[8] T. Kim, E. Lin, J. Lee, C. Lau, and V. Mugunthan, "Navigating data heterogeneity in federated learning: A semi-supervised approach for object detection," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[9] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Ensemble attention distillation for privacy-preserving federated learning," in *IEEE/CVF Int. Conf. Comput. Vis*, 2021, pp. 15 076–15 086.

[10] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, "Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," in *Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–15.

[11] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1852–1864, 2020.

[12] X. Cai, W. Jiang, R. Xu, W. Zhao, J. Ma, S. Liu, and Y. Li, "Analyzing infrastructure lidar placement with realistic lidar simulation library," in *IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5581–5587.

[13] J.-J. Gonzalez-Barbosa, T. Garcia-Ramirez, J. Salas, J.-B. Hurtado-Ramos *et al.*, "Optimal camera placement for total coverage," in *IEEE Int. Conf. Robot. Automat.*, 2009, pp. 844–848.

[14] R. Vijay, J. Cherian, R. Riah, N. De Boer, and A. Choudhury, "Optimal placement of roadside infrastructure sensors towards safer autonomous vehicle deployments," in *IEEE Int. Intell. Transp. Syst. Conf.* IEEE, 2021, pp. 2589–2595.

[15] W. Zhang, Y. Wang, Z. You, M. Cao, p. Huang, J. Shan, Z. Yang, and B. Cui, "Information gain propagation: a new way to graph active learning with soft labels," in *Int. Conf. Learn. Represent. (ICLR)*, 2022.

[16] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control and Decis.*, vol. 5, no. 1, pp. 42–60, 2018.

[17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conf. Robot Learn. (CoRL)*, vol. 78, 2017, pp. 1–16.

[18] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.