Fleet Active Learning: A Submodular Maximization Approach

Anonymous Author(s)
Affiliation
Address
email

Abstract: In multi-robot systems, robots often gather data to improve the performance of their deep neural networks (DNNs) for perception and planning. Ideally, these robots should select the most informative samples from their local data distributions by employing active learning approaches. However, when the data collection is distributed among multiple robots, redundancy becomes an issue as different robots may select similar data points. To overcome this challenge, we propose a fleet active learning (FAL) framework in which robots collectively select informative data samples to enhance their DNN models. Our framework leverages submodular maximization techniques to prioritize the selection of samples with high information gain. Through an iterative algorithm, the robots coordinate their efforts to collectively select the most valuable samples while minimizing communication between robots. We provide a theoretical analysis of the performance of our proposed framework and show that it is able to approximate the NP-hard optimal solution. We demonstrate the effectiveness of our framework through experiments on real-world perception and classification datasets, which include autonomous driving datasets such as Berkeley DeepDrive. Our results show an improvement by up to 25.0% in classification accuracy, 9.2% in mean average precision and 48.5% in the submodular objective value compared to baselines.

1 Introduction

Consider a scenario where a fleet of autonomous vehicles (AVs) operates in various environments, ranging from urban to rural or highway settings. These AVs aim to enhance their machine learning (ML) models employed in perception, prediction, and planning by collecting data and sharing it with a central server. An ideal strategy for these AVs is to gather diverse data, enabling the trained models to generalize effectively across different environments. However, due to limitations such as bandwidth, computational resources, and storage capacity, the AVs can only transmit a limited amount of data to the central server. Notably, one AV can generate a data stream of more than 20-30 Gigabytes (GB) per second, combining video and LiDAR data [1]. In comparison, a standard 5G wireless network offers a bandwidth of only 10 Gbps, which needs to be shared among multiple users [2]. Consequently, it becomes crucial for the AVs to collaborate in selecting the most informative data points that optimize the performance of the trained models.

To illustrate, suppose one AV collects data from a congested urban area during peak hours while another AV captures data from a sparsely populated rural region. Instead of duplicating efforts in either environment, the AVs should collaborate to ensure data diversity in the collected images. By strategically selecting complementary images, the AVs contribute to the creation of a diverse dataset that enhances the robustness of the trained models across various driving scenarios.

In our formulation (Fig.1), AVs communicate with each other using minimal information and exclusively have access to their observed data, ensuring the AVs maintain data privacy and only share selected information. Moreover, we acknowledge that the collected AV data is not independent and identically distributed (i.i.d.), and the incremental value of selecting a particular data point relies on the data points already chosen. Additionally, we do not assume the presence of a perfect labeler capable of labeling data points on the fly; instead, we leverage the outputs of imperfect ML models.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

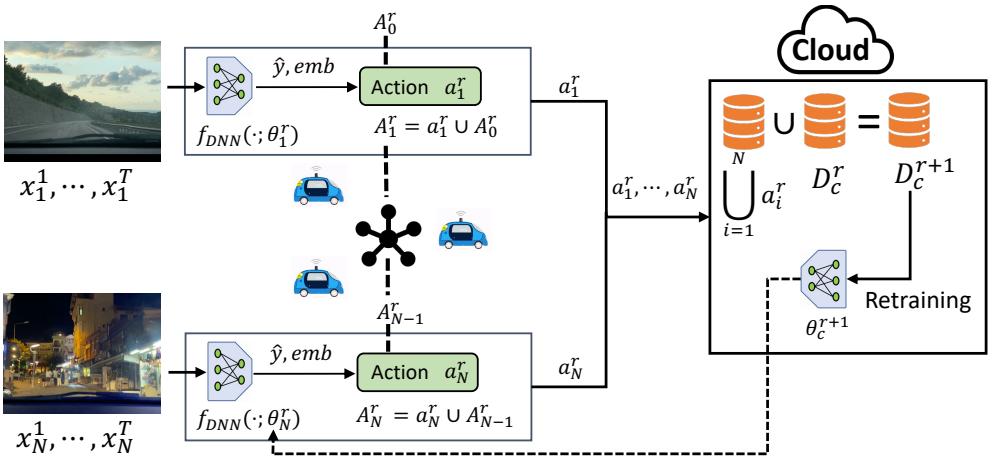

Figure 1: **Fleet Active Learning Framework:** Each robot i observes a stream of data points $x_i^1, \ldots, x_i^T$ and processes them using its neural network $f_{\text{DNN}}(\cdot;\theta_i^r)$ to obtain predictions $\hat{y}$ and embeddings emb. Then, robots utilize these embeddings and predictions to select an action a_i^r to sample data points, maximizing a submodular function f while considering the previous robots' actions A_{i-1}^r. The aggregated action $A_i^r = a_i^r \cup A_{i-1}^r$ is passed to the next robot, repeating the process until all robots have taken an action. At the end of each round, the actions are shared with the cloud, which labels the newly acquired data points and updates the training dataset $\mathcal{D}_c^r$ with newly acquired data points $\bigcup_{i=1}^{N} a_i^r$, resulting in the creation of a new dataset $\mathcal{D}_c^{r+1}$. The model is retrained with the new dataset, yielding θ_c^{r+1}. Finally, the cloud shares the updated model weights with all robots, and each robot updates its model parameters accordingly.

To create a framework that can be applied to a variety of data collection problems, we formulated this problem as a distributed submodular maximization problem. The reasoning behind this comes from the fact that the gain in collecting data points decreases over time, and the value of the data points is not independent but rather depends on the previously selected data points. The submodularity property captures the diminishing return in collecting data points and the dependence of the value of the data points on the previously selected data points. The main contributions of our work can be summarized as follows:

1. We present a framework to scale active learning in a multi-robot setting by formulating the data collection problem in a networked AV system as a distributed submodular maximization problem.
2. We propose an interactive algorithm that iteratively updates the actions of the AVs and provide a rigorous proof that it achieves a solution within the $1/2$ optimality bound.
3. We empirically show that our interactive algorithm performs similarly to a centralized algorithm while outperforming a fully decentralized algorithm. Our algorithm shows an improvement of up to 48.5% in the submodular objective, 25.0% in classification accuracy, and 9.2% in mean average precision (mAP) compared to the fully decentralized approach in real-world datasets including the Berkeley DeepDrive autonomous driving dataset [3].

We also utilize embeddings generated by vision and language models such as CLIP [4], which are trained on large-scale datasets to further enhance the data point selection process. These embeddings aid in selecting data points to effectively cover the entire data distribution.

2 Related Work

Data collection is a well-studied problem in the robotics and machine learning literature, closely related to active learning [5–9] and cloud robotics [10–20]. However, existing approaches typically involve robots sharing *all* their data without considering other robots' input during the data collection process. The closest work to ours is [19], which addresses data collection in a multi-robot setting with the interaction between robots. However, their focus is on minimizing a convex loss function on dataset statistics, with an equal value assigned to each class point. In contrast, our framework

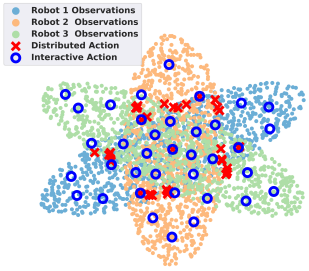

Figure 2: **Enhancing Information Coverage through Cooperation:** We illustrate a toy example involving 3 robots observing data mapped into a 2D space. Each point represents an embedding of a data point. The objective is to maximize the coverage of the observed data points through selected points. In the distributed setting (red), each robot selects a subset of data points independently, leading to overlaps and redundancy. However, when the robots cooperate using an iterative policy (blue), they consider previous actions, resulting in more diverse data collection that covers a larger portion of the data points.

provides a more general approach, considering cases where data points are unequally valued and utilizing submodular maximization for selecting the most informative samples.

Another related topic is federated learning (FL) [21–29]. FL aims to train a model using data distributed among multiple robots, where robots train the model locally and share parameters with each other. However, our problem setting fundamentally differs as the robots lack access to perfectly labeled data and need to transmit their data to the cloud to obtain labels. Additionally, the models we consider are considerably large, making local training infeasible.

Our work predominantly falls into the category of pool-based active learning [7, 30], where the objective is to select the most informative data points from an unlabeled pool. Various methods have been proposed for selecting these data points in general machine learning settings [31–37] and robotics [38–41], each capturing informativeness in different ways and also requiring access to all unlabeled data points. In our case, however, we consider a multi-robot setting where robots only have access to their local data distribution. Our work extends these methods naturally to be applicable in a multi-robot context.

The problem of active learning is closely related to submodular maximization [42, 43], as submodular functions capture the diminishing returns property and the marginal gain of selecting a data point depends on the previously selected points. Numerous works address submodular maximization in both decentralized [42, 44–47] and centralized settings [43, 48–54]. Specifically, the works such as [46, 47] investigate a similar problem to ours. They focused on selecting subsets of elements in a distributed manner but with the constraint of choosing only one element in each robot. In our work, we extend this approach to handle a broader range of scenarios, enabling the selection of multiple elements simultaneously in a distributed manner. Our extension is essential since AVs upload a batch of sampled data to labelers over bandwidth-limited networks.

3 Problem Formulation

We consider a practical problem that involves data collection from a distributed fleet of robots for training a robust machine learning model in the cloud. Our goal is to select the most informative samples from each robot in the fleet, such that when these data points are added to the training dataset, the accuracy of the trained model is maximized. This problem formulation can be seen as a generalized version of the active learning problem in the context of a distributed fleet of robots.

Robot Perception Model: We assume a fleet of N_{robot} robots operating in diverse environments, each performing a computer vision task such as classification or object detection. Robot i observes a set of data points $\{x_i^1, \ldots, x_i^T\} = X_i^r$, which represents the accumulated data available for sampling in round r. A round represents a data collection period, like a day, during which robots operate and gather data. In each round, the robots sample N^{cache} data points from their accumulated data X_i^r and send them to the cloud to retrain an ML model. The perception model used by robot i is a DNN represented as $f_{\text{DNN}}(x; \theta_i^r) = \hat{y}, emb$, where θ_i^r denotes the parameters of the perception model. The perception model provides predictions $\hat{y}$ and embeddings emb for the input data x.

Embeddings are crucial in our approach as they capture the similarity between data points. Unlike raw data, which is high-dimensional and ineffective at summarizing information, embeddings provide a more efficient and accurate representation. By leveraging embeddings generated by foundation models like CLIP [4] or other active learning methods [32–34, 55], we enhance our ability to capture data similarity effectively.

Assumption 1 (Robots' Accumulated Data Points are Disjoint)**.** *Each robot i stores its observed data X_i^r separately, resulting in disjoint sets of accumulated data points for selection in each round r. That is, $X_i^r \cap X_j^r = \emptyset$ for all $i \neq j$.*

This assumption reflects the fact that each robot maintains its own storage of observed data points. As a result, the sets of accumulated data available for selection in each round are disjoint. Then for each round r, we can define the set of all observed images as $X^r = \bigcup_{i=1}^{N_{\text{robot}}} X_i^r$.

Robotic Fleet: The fleet of N_{robot} robots operate in diverse environments, observing different data distributions X_i^r in each round r. In each round, each robot i selects at most N_i^{cache} data points from its observed images X_i^r and sends them to the cloud for retraining the ML model. The size of the selected samples N_i^{cache} is determined based on factors such as communication bandwidth and labeling budget. The samples chosen by robot i in round r are denoted as action a_i^r, where $a_i^r \subset X_i^r$ and $|a_i^r| \leq N_i^{\text{cache}}$, indicating that the number of selected samples is smaller than or equal to the cache size N_i^{cache}. Additionally, we assume that the robotic fleet is connected via a wireless network, enabling communication among the robots.

Definition 1 (Robots' Feasible Action Space)**.** *In round r, each robot i can select at most N_i^{cache} samples from the observed datapoints X_i^r such that:*

$$\mathcal{A}_i^r = \{a_i^r \subset X_i^r : |a_i^r| \leq N_i^{\text{cache}}\}, \quad \mathcal{A}^r = \{ \bigcup_{i=1}^{N_{\text{robot}}} a_i^r : a_i^r \in \mathcal{A}_i^r \ , \forall i = 1, \ldots, N_{\text{robot}}\}.$$

Here, $\mathcal{A}_i^r$ represents the set of all feasible actions for robot i, and $\mathcal{A}^r$ represents the combined feasible actions of all robots.

Data Collection Problem: The goal of the data collection process is to select the most informative samples from each robot in the fleet, such that when these samples are added to the training set, the accuracy of the trained model is maximized. However, there is no proven function that reliably represents dataset quality, as it depends on various factors such as data diversity and image quality [56]. Instead, heuristic methods are used to measure dataset quality. To provide a general framework for the data collection problem, we will assume two properties about the dataset quality.

First, we will assume that the quality of the dataset always increases with the addition of new samples [57]. This is a reasonable assumption since, unless the training data points are intentionally manipulated, adding more data points improves the coverage of the training set of the real data distribution. Thus, we can safely make the following assumption:

Assumption 2 (Dataset Quality Function is Monotone)**.** *The dataset quality function $f : 2^X \to \mathbb{R}$ is a monotone function, meaning that for all $A \subseteq B \subseteq X, \quad f(A) \leq f(B)$.*

Secondly, we assume that the marginal gain of adding a new sample to the training dataset exhibits a diminishing returns. This is also a reasonable assumption because if the training dataset already covers a significant portion of the real data distribution, adding more samples may not increase the accuracy of the trained model significantly. We verified this on four real-world datasets, confirming a diminishing returns relationship between test accuracy and the percentage of training data (App. A.5). This property, known as submodularity, is exhibited by many problems, such as set cover, facility location, and sensor placement [48]. Formally, we state the following assumption:

Assumption 3 (Dataset Quality Function is Submodular)**.** *The dataset quality function $f : 2^X \to \mathbb{R}$ is submodular, i.e., for all $A \subseteq B \subseteq X$ and $x \in X \setminus B$ it holds that $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$.*

Finally, we define the data collection problem: given a fleet of N_{robot} robots sharing a common dataset $\mathcal{D}_c^r$ in the cloud, the objective is to select a maximum of N_i^{cache} samples from each robot i, maximizing the quality of the dataset. Formally, we can define the data collection problem as:

$$\begin{aligned} \max_{a_1^r, \ldots, a_{N_{\text{robot}}}^r} \quad & f(\mathcal{D}_c^r \cup \bigcup_{i=1}^{N_{\text{robot}}} a_i^r) \\ \text{subject to:} \quad & a_i^r \subseteq X_i^r \qquad \forall i = 1, \ldots, N_{\text{robot}} \\ & |a_i^r| \leq N_i^{\text{cache}} \quad \forall i = 1, \ldots, N_{\text{robot}}. \end{aligned} \tag{1}$$

This problem is a combinatorial optimization problem, and without any further assumptions, solving for the optimal solution is NP-hard. Thus, we use algorithms that provide approximation guarantees to the optimal solution. For such problems, the greedy algorithm is shown to provide a good approximation to the optimal solution [50].

There are various submodular functions that can be used; such examples are facility location, mutual information, or the set cover problem [48]. In our experiments, we used a facility location function, which assesses how well a subset A represents the entire dataset X [58]. It accomplishes this by calculating the sum of similarities $M_{x,a}$ between each element $x \in X$ and its closest element $a \in A$. The facility location function is known to be submodular for non-negative similarity values $M_{x,a} \geq 0$ [59]. Fig. 2 shows an example of how our facility location submodular objective encourages covering the data distribution. In formal terms, the facility location function is defined as follows:

$$f(A) = \sum_{x \in X^r} \max_{a \in A} M_{x,a}. \tag{2}$$

Centralized Action Policy: We now introduce a centralized action policy, named CENTRALIZED, described in Alg. 1. In this policy, a centralized cloud has access to all the data points observed by the robots X^r (line 4). Starting from an empty solution set (line 3), the central cloud iteratively evaluates all combined data points X^r, selects the data point within the feasible action space that maximizes the submodular function f (line 6) and adds it to the action (line 7). This process continues until no more points can be added to the action (lines 5 - 8). The action of the CENTRALIZED policy, denoted as a^C, is then appended to the cloud dataset $\mathcal{D}_c^r$ (line 9), and this algorithm is repeated for each round r.

This algorithm has been proven to provide $1/2$ optimality bound to the optimal solution [50]. Additionally, randomized approximations exist that can achieve a $1 - 1/e$ optimality bound [49]. We have chosen this version of the centralized policy because of its simplicity and generality. Since solving the combinatorial optimization problem is NP-hard, we will use this centralized policy as a target benchmark to compare the performance of our framework.

1 **Input:** $\mathcal{D}_c^r, X_i^r, f$;
2 **Output:** $\mathcal{D}_c^{r+1}$;
3 Initialize $a^C = \emptyset$;
4 $X^r = \bigcup_{i=1}^{N_{\text{robot}}} X_i^r$;
5 **for** $j = 1$ to $\sum_{i=1}^{N_{\text{robot}}} N_i^{\text{cache}}$ **do**
6 $\quad x_j = \underset{x: a^C \cup \{x\} \in \mathcal{A}^r}{\text{argmax}} \; f(\mathcal{D}_c^r \cup a^C \cup \{x\})$;
7 $\quad a^C = a^C \cup \{x_j\}$;
8 **end**
9 $\mathcal{D}_c^{r+1} = \mathcal{D}_c^r \cup a^C$;
10 **return** $\mathcal{D}_c^{r+1}$;

Algorithm 1: CENTRALIZED Policy

1 **Input:** $\mathcal{D}_c^r, X_i^r, f$;
2 **Output:** $\mathcal{D}_c^{r+1}$;
3 **for** $i = 1$ to N_{robot} **do**
4 $\quad$ Initialize $a_i^D = \emptyset$;
5 $\quad$ **for** $j = 1$ to N_i^{cache} **do**
6 $\quad\quad x_j = \underset{x: a_i^D \cup \{x\} \in \mathcal{A}_i^r}{\text{argmax}} \; f(\mathcal{D}_c^r \cup a_i^D \cup \{x\})$;
7 $\quad\quad a_i^D = a_i^D \cup \{x_j\}$;
8 $\quad$ **end**
9 **end**
10 $\mathcal{D}_c^{r+1} = \mathcal{D}_c^r \cup \bigcup_{i=1}^{N_{\text{robot}}} a_i^D$;
11 **return** $\mathcal{D}_c^{r+1}$;

Algorithm 2: DISTRIBUTED Policy

Distributed Action Policy: In another scenario, we can solve this problem without considering any communication between robots, referred to as DISTRIBUTED . This policy is described in Alg. 2. In the DISTRIBUTED policy, each robot independently runs the greedy selection in its observed images X_i^r in parallel (lines 3 - 9). Starting with an empty solution (line 4), each robot adds data points that maximize the submodular function f from its observed data points (line 5). At the end of round r, the selected data points a_i^D are shared with the cloud, which adds them to the cloud dataset $\mathcal{D}_c^r$ (line 11). Although the DISTRIBUTED policy can be run in parallel and preserves the distributed nature of the problem, it does not consider other robots' actions, potentially resulting in repeatedly selecting similar elements. Consequently, the solutions obtained through the DISTRIBUTED policy can be a $\Omega(N_{\text{robot}})$ factor worse than the centralized solution [42].

These algorithms offer different approaches for sample selection from distributed robots, with the CENTRALIZED policy utilizing global information and the DISTRIBUTED policy relying on local observations of each robot. The CENTRALIZED policy needs $O(\sum_{i=1}^{N_{\text{robot}}} N_i^{\text{cache}})$ iterations over all robots and their samples, which may be impractical for a large number of robots. On the other hand,

the DISTRIBUTED policy can be implemented in parallel, making it more scalable. Our method balances these two policies using a distributed policy with an iterative component.

Model Retraining and Updating Weights: After each data collection round r, the cloud dataset $\mathcal{D}_c^r$ is expanded with selected samples, and the perception model is retrained on the cloud using the updated dataset $\mathcal{D}_c^{r+1} = \mathcal{D}_c^r \cup \bigcup_{i=1}^{N_{\text{robot}}} a_i^r$ to obtain new model weights θ_c^{r+1}. These weights are then shared with the robots. The robots can simply use the updated cloud model θ_c^{r+1} or optionally fine-tune it on their local datasets to yield models θ_i^{r+1}.

4 An Interactive Approach to Fleet Active Learning

This section introduces our proposed policy, called INTERACTIVE , for the data collection problem. The INTERACTIVE policy begins with an empty solution (line 1) and sequentially determines the actions of the robots (lines 4 - 7). Each robot i takes into account the previous actions of the robots 1 to $i-1$, denoted as A_{i-1}^I, and runs the greedy algorithm given in Alg.3 to determine its action a_i^I and includes it in the previous actions A_i^I (line 6). This process is repeated until all robots have determined their actions. Finally, the actions of all the robots are combined (line 8), resulting in the combined action a^I, and the cloud dataset is updated (line 9).

1 **Input:** $\mathcal{D}_c^r$, X_i^r, f , A_{i-1}^I ;
2 **Output:** a_i^I ;
3 $a_i^I = \emptyset$;
4 **for** $j = 1$ to N_i^{cache} **do**
5 $\quad x_j = \underset{x: a_i^I \cup \{x\} \in A_i^r}{\text{argmax}} f(\mathcal{D}_c^r \cup A_{i-1}^I \cup a_i^I \cup \{x\})$;
6 $\quad a_i^I = a_i^I \cup \{x_j\}$;
7 **end**
8 **return** a_i^I ;

Algorithm 3: Action per Robot

1 **Input:** $\mathcal{D}_c^r$, X_i^r, f ;
2 **Output:** $\mathcal{D}_c^{r+1}$;
3 $A_0^I = \emptyset$;
4 **for** $i = 1$ to N_{robot} **do**
5 $\quad$ Get action a_i^I using algorithm 3 ;
6 $\quad A_i^I = A_{i-1}^I \cup a_i^I$;
7 **end**
8 $a^I = \bigcup_{i=1}^{N_{\text{robot}}} a_i^I$;
9 $\mathcal{D}_c^{r+1} = \mathcal{D}_c^r \cup a^I$;
10 **return** $\mathcal{D}_c^{r+1}$;

Algorithm 4: INTERACTIVE Policy

Our proposed INTERACTIVE policy represents an improvement over both the CENTRALIZED and DISTRIBUTED policies. It achieves a solution with the same optimality bound as the CENTRALIZED policy while allowing for distributed execution. Moreover, the selection of points occurs only once between robots, leading to a reduced number of messages passings between robots. To minimize data exchange, we can share the embeddings of the points instead of the actual points themselves [15]. This significantly reduces the amount of data transmitted. Furthermore, for specific functions like the facility location function, additional optimizations can be employed. For example, rather than sharing the points or embeddings, robots can update and share the maximum value of the similarity metric $M_{x,a}$. We also theoretically show that the given algorithm will achieve a solution that is at least $1/2$ good as the optimal solution for submodular monotone objective functions.

Theorem 1 (Optimality of INTERACTIVE Policy)**.** *The* INTERACTIVE *policy described in Alg. 4 achieves a solution that is at least* $1/2$ *as good as the optimal solution for submodular monotone objective functions, that is:*

$$f_{\mathcal{D}_c^r}(a^I) \geq \frac{1}{2} f_{\mathcal{D}_c^r}(a^{\text{OPT}}). \tag{3}$$

Here we denote the improvement on the objective function as $f_{\mathcal{D}_c^r}(x) = f(\mathcal{D}_c^r \cup x) - f(\mathcal{D}_c^r)$. a^{OPT} *is the optimal solution to the optimization problem given in Eq. 1 and* a^I *is the solution obtained by the* INTERACTIVE *policy.*

The proof of the theorem is given in Appendix A.2. As demonstrated in the theorem, the optimality bound of the INTERACTIVE policy is independent of the number or ordering of the robots. Furthermore, the number of messages exchanged between robots is $O(N_{\text{robot}})$. This subtle, yet significant, difference between the CENTRALIZED policy and the INTERACTIVE policy enables the INTERACTIVE policy to be scalable to a large number of robots. Our interactive policy is also much more compute-efficient than centralized and we provide complexity results in the Appendix.

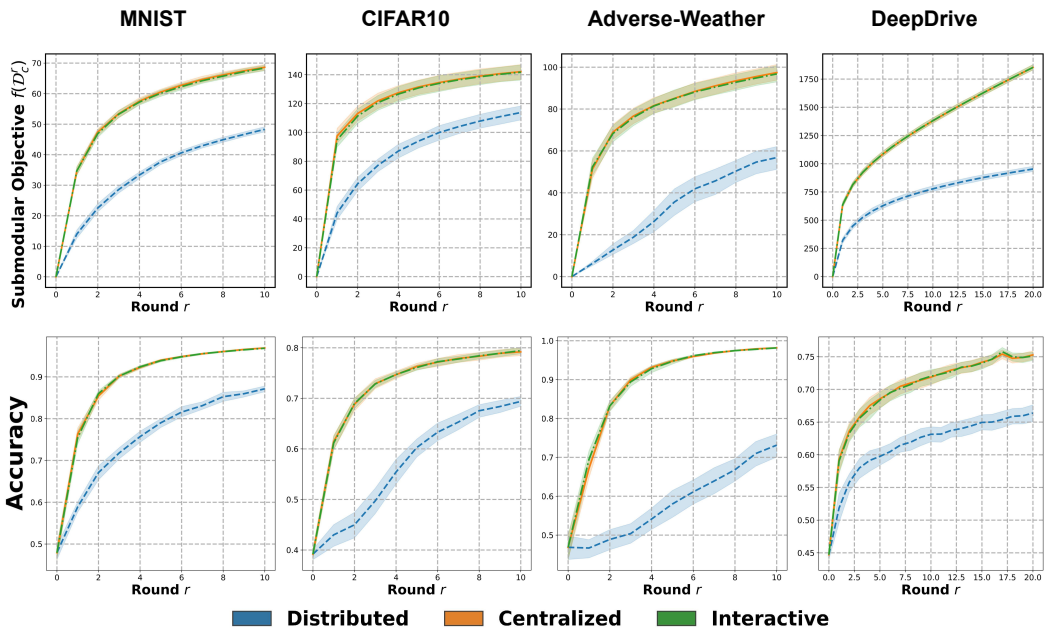

Figure 3: **Our INTERACTIVE policy achieves similar performance with the CENTRALIZED policy and outperforms the DISTRIBUTED policy.** Each column represents a different dataset. *Row 1* shows the submodular objective of the cloud dataset across the rounds. Both the CENTRALIZED and INTERACTIVE policies achieve similar objectives, while the DISTRIBUTED fails to reach the same level of performance. *Row 2* presents the accuracy of the retrained neural networks using the dataset $\mathcal{D}_c^r$ in each round r. The accuracy plots exhibits a similar trend as the submodular objective, with the INTERACTIVE and CENTRALIZED policies consistently outperforming the DISTRIBUTED policy.

5 Experiments

We evaluate our method by comparing it against several benchmarks using four diverse datasets. The first two datasets, `MNIST` [60] and `CIFAR-10` [61], serve as foundational benchmarks, while the `Adverse-Weather` dataset [62] and the `DeepDrive` dataset [3] provide realistic scenarios with data captured by autonomous vehicles. In our simulations, we simulate the data observation processes under heterogeneous environments and diverse data distributions. For classification tasks, we employ state-of-the-art vision models such as the Vision Transformer [63] and ResNet [64], while for object detection tasks, we utilize YOLOv8 [65]. Detailed information regarding our experimental setup and the models is given in the Appendix.

Comparison Metrics: For classification tasks, we compare the accuracy of the trained model on newly acquired datasets $\mathcal{D}_c^{r+1}$ using the held-out test dataset. We compare the mean Average Precision (mAP) for object detection tasks among different data sampling policies. We also compare the submodular objective value of selected data points in each round. All experiments are repeated with 25 different seeds and the results are averaged.

Policy	mAP
INITIAL	33.2
DISTRIBUTED	37.2
CENTRALIZED	46.5
INTERACTIVE	46.4

Table 1: Results of `DeepDrive` object detection.

Submodular Objective Function: We use the facility location function, given in Eq. 2, as the objective function. To measure the similarity between points x and a, we employ the similarity metric $M_{x,a} = \frac{1}{1+\beta d(x,a)}$. Here, the distance $d(x,a)$ is computed as the L_2 distance between the embeddings of the data points, given by $\|emb_x - emb_a\|_2$. The embeddings are obtained by the models f_{DNN} and we set the hyperparameter $\beta = 0.01$ to control the significance of the distance metric. In our simulations, we employ the lazy greedy algorithm [51], a variant of the greedy algorithm that achieves the same solution while being more efficient in practice.

Results: The experimental results (Fig. 3) show that our INTERACTIVE policy performs as well as the CENTRALIZED policy, as we have shown in the Theorem 1. Moreover, our INTERACTIVE policy consistently outperforms the DISTRIBUTED policy in all simulations.

Does cooperation increase the submodular objective?

Our proposed INTERACTIVE policy outperforms the DISTRIBUTED policy and achieves similar performance to the CENTRALIZED policy on all four datasets. This performance gain of $29.8\%, 19.7\%, 41.2\%, 48.5\%$ is obtained because the INTERACTIVE policy makes selections considering all the other robots before it and is able to reduce redundancy, as shown in Fig. 4.

What is the accuracy of the vision model after each round?

In the second row of Fig.3, we show the perception model's accuracy after each retraining round on the dataset $\mathcal{D}_c^r$. Clearly, our INTERACTIVE policy consistently achieves higher accuracies compared to the DISTRIBUTED policy, outperforming it by $9.7\%, 10.1\%, 25.0\%, 8.8\%$ and achieves similar accuracy values to the CENTRALIZED case.

What is the object detection performance?

In Table 1, we present the mean average precision (mAP) of the initial DNN, denoted as INITIAL , and at the end of all data collection rounds for each policy. Our INTERACTIVE policy and CENTRALIZED policy show significantly better performance compared to the DISTRIBUTED policy, surpassing it by 9.2%.

Limitations: Our formulation, while effective, has several assumptions that limit its generality. Firstly, our formulation assumes that the data collection rounds are synchronous among all robots and that robots have communication capabilities with each other. Finally, we assume that the robots are trying to maximize the objective function cooperatively, whereas, in a real-world setting, the robots may need to be incentivized based on their contributions.

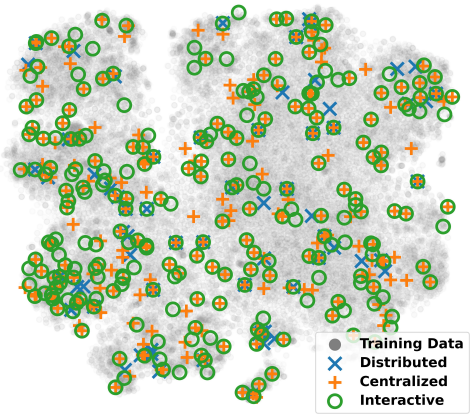

Figure 4: **Visualization of Policy Coverage in Feature Space:** We employed t-SNE [62] to project the embeddings of data points into a 2D space, facilitating the comparison of coverage across different policies. Data points are color-coded according to the policy that selected them, with INTERACTIVE (green), CENTRALIZED (orange), and DISTRIBUTED (blue) policies represented. The visualization demonstrates that our proposed INTERACTIVE method performs on par with the CENTRALIZED baseline and achieves superior coverage of the training data (gray) in comparison to the DISTRIBUTED policy. Although the DISTRIBUTED policy might seem to have fewer points, this can be attributed to the overlapping and repetition of similar data points that are closely mapped together in the visualization.

6 Conclusion and Future Work

We propose a framework to scale active learning algorithms to the multi-robot setting utilizing submodular maximization. Our framework enables robots to determine their actions sequentially, taking into account the actions of the previous robots. We show that when the objective function is submodular and monotone, the proposed framework achieves $1/2$ optimality bound to the optimal solution, which is NP-hard to compute. Through experiments on real-world datasets, we confirm that such an iterative algorithm will result in increased accuracy and objective function when compared to the distributed setting. Our work is especially useful when the robots are operating in a resource-constrained environment, where data sharing is costly, and collective actions need to be optimized to reduce redundancy. We believe that our work is a step towards the development of a general framework for active learning in multi-robot systems.

Future research directions include incorporating more realistic constraints. This involves incorporating asynchronous data collection rounds, where the robots can collect data at different times, and incorporating privacy-preserving mechanisms to ensure confidentiality. Furthermore, we aim to extend our formulation where robots collect data for multiple tasks, each with varying value and importance.

References

[1] Data is the new oil in the future of automated driving. https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/#gs.LoDUaZ4b, 2016. [Online; accessed 10-June-2021].

[2] 10gbps 5g data speeds - qualcomm and keysight achieve industry-first milestone. https://datacenternews.asia/story/10gbps-5g-data-speeds-qualcomm-and-keysight-achieve-industry-first-milestone, 2021. [Online; accessed 14-June-2022].

[3] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2636–2645, 2020.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning, 2021.

[5] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH, 4:129–145, 1996.

[6] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1183–1192. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/gal17a.html.

[7] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[8] Simon Tong. Active learning: theory and applications, volume 1. Stanford University USA, 2001.

[9] Kai Yang, Jie Ren, Yanqiao Zhu, and Weiyi Zhang. Active learning for wireless iot intrusion detection. IEEE Wireless Communications, 25(6):19–25, 2018. doi:10.1109/MWC.2017.1800079.

[10] Cloud robotics and automation. http://goldberg.berkeley.edu/cloud-robotics/, 2019. [Online; accessed 02-Jan.-2019].

[11] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. IEEE Trans. Automation Science and Engineering, 12(2):398–409, 2015.

[12] J Kuffner. Cloud-enabled robots in: Ieee-ras international conference on humanoid robots. Piscataway, NJ: IEEE, 2010.

[13] Sandeep Chinchali, E. Pergament, M. Nakanoya, E. Cidon, E. Zhang, D. Bharadia, M. Pavone, and S. Katti. Sampling training data for continual learning between robots and the cloud. In 2020 International Symposium on Experimental Robotics (ISER), Valetta, Malta, 2020.

[14] Ken Goldberg and Ben Kehoe. Cloud robotics and automation: A survey of related work. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5, 2013.

[15] Sandeep Chinchali, Apoorva Sharma, James Harrison, Amine Elhafsi, Daniel Kang, Evgenya Pergament, Eyal Cidon, Sachin Katti, and Marco Pavone. Network offloading policies for cloud robotics: a learning-based approach. Autonomous Robots, 45(7):997–1012, 2021. doi:10.1007/s10514-021-09987-4. URL https://doi.org/10.1007/s10514-021-09987-4.

[16] Ajay Kumar Tanwani, Nitesh Mor, John D. Kubiatowicz, Joseph E. Gonzalez, and Ken Goldberg. A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering. 2019 International Conference on Robotics and Automation (ICRA), pages 4559–4566, 2019.

[17] Yuchong Geng, Dongyue Zhang, Po-han Li, Oguzhan Akcin, Ao Tang, and Sandeep P Chinchali. Decentralized sharing and valuation of fleet robotic data. In 5th Annual Conference on Robot Learning, Blue Sky Submission Track, 2021.

[18] V. Casamayor Pujol and S. Dustdar. Fog robotics—understanding the research challenges. IEEE Internet Computing, 25(05):10–17, sep 2021. ISSN 1941-0131. doi:10.1109/MIC.2021.3060963.

[19] Oguzhan Akcin, Po-han Li, Shubhankar Agarwal, and Sandeep P. Chinchali. Decentralized data collection for robotic fleet learning: A game-theoretic approach. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, Proceedings of The 6th Conference on Robot Learning, volume 205 of Proceedings of Machine Learning Research, pages 978–988. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/akcin23a.html.

[20] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, K Dharmarajan, Brijen Thananjeyan, P. Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In Conference on Robot Learning, 2022.

[21] Federated learning: Collaborative machine learning without centralized training data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, 2017. [Online; accessed 15-June-2021].

[22] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. CoRR, abs/1610.05492, 2016. URL http://arxiv.org/abs/1610.05492.

[23] Yu Xianjia, Jorge Peña Queralta, Jukka Heikkonen, and Tomi Westerlund. Federated learning in robotic and autonomous systems. Procedia Computer Science, 191:135–142, 2021. ISSN 1877-0509. doi:https://doi.org/10.1016/j.procs.2021.07.041. URL https://www.sciencedirect.com/science/article/pii/S187705092101437X. The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 16th International Conference on Future Networks and Communications (FNC), The 11th International Conference on Sustainable Energy Information Technology.

[24] Elsa Rizk, Stefan Vlaski, and Ali H. Sayed. Optimal importance sampling for federated learning. In ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3095–3099, 2021. doi:10.1109/ICASSP39728.2021.9413655.

[25] Aleksandar Armacki, Dragana Bajovic, Dusan Jakovetic, and Soummya Kar. Personalized federated learning via convex clustering. arXiv preprint arXiv:2202.00718, 2022.

[26] Hong Xing, Osvaldo Simeone, and Suzhi Bi. Decentralized federated learning via sgd over wireless d2d networks. In 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pages 1–5, 2020. doi:10.1109/SPAWC48557.2020.9154332.

[27] Bin Gu, An Xu, Zhouyuan Huo, Cheng Deng, and Heng Huang. Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. IEEE Transactions on Neural Networks and Learning Systems, pages 1–13, 2021. doi:10.1109/TNNLS.2021.3072238.

[28] Solmaz Niknam, Harpreet S. Dhillon, and Jeffrey H. Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. IEEE Communications Magazine, 58(6):46–51, 2020. doi:10.1109/MCOM.001.1900461.

[29] Sebastian Caldas, Jakub Konečny, H. Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements, 2018.

[30] Xueying Zhan, Huan Liu, Qing Li, and Antoni Chan. A comparative survey: Benchmarking for pool-based active learning. pages 4679–4686, 08 2021. doi:10.24963/ijcai.2021/634.

[31] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In International Conference on Machine Learning, 2017.

[32] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

[33] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=ryghZJBKPS.

[34] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/95323660ed2124450caaac2c46b5ed90-Paper.pdf.

[35] Ajay Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. pages 2372–2379, 2009.

[36] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. ICML, Williamstown, pages 441–448, 2001.

[37] Jiaxi Wu, Jiaxin Chen, and Di Huang. Entropy-based active learning for object detection with progressive diversity constraint. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9387–9396, 2022.

[38] Jingyun Yang, Junwu Zhang, Connor Settle, Akshara Rai, Rika Antonova, and Jeannette Bohg. Learning periodic tasks from human demonstrations. In 2022 International Conference on Robotics and Automation (ICRA), pages 8658–8665, 2022. doi:10.1109/ICRA46639.2022.9812402.

[39] Maya Cakmak, Crystal Chao, and Andrea L. Thomaz. Designing interactions for robot active learners. IEEE Transactions on Autonomous Mental Development, 2(2):108–118, 2010. doi:10.1109/TAMD.2010.2051030.

[40] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. Transparent active learning for robots. In 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 317–324, 2010. doi:10.1109/HRI.2010.5453178.

[41] Annalisa T. Taylor, Thomas A. Berrueta, and Todd D. Murphey. Active learning in robotics: A review of control principles. Mechatronics, 77:102576, 2021. ISSN 0957-4158. doi:https://doi.org/10.1016/j.mechatronics.2021.102576. URL https://www.sciencedirect.com/science/article/pii/S0957415821000659.

[42] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. Journal of Machine Learning Research, 17(235):1–44, 2016. URL http://jmlr.org/papers/v17/mirzasoleiman16a.html.

[43] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, page 1954–1963. JMLR.org, 2015.

[44] Bahman Gharesifard and Stephen L. Smith. Distributed submodular maximization with limited information. IEEE Transactions on Control of Network Systems, 5(4):1635–1645, 2018. doi:10.1109/TCNS.2017.2740625.

[45] Vahab Mirrokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, STOC '15, page 153–162, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335362. doi:10.1145/2746539.2746624. URL https://doi.org/10.1145/2746539.2746624.

[46] David Grimsman, Mohd. Shabbir Ali, João P. Hespanha, and Jason R. Marden. The impact of information in distributed submodular maximization. IEEE Transactions on Control of Network Systems, 6(4):1334–1343, 2019. doi:10.1109/TCNS.2018.2889005.

[47] Micah Corah and Nathan Michael. Distributed submodular maximization on partition matroids for planning on large sensor networks. In 2018 IEEE Conference on Decision and Control (CDC), pages 6792–6799, 2018. doi:10.1109/CDC.2018.8619396.

[48] Andreas Krause and Daniel Golovin. Submodular function maximization. In Tractability, 2014.

[49] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In Matteo Fischetti and David P. Williamson, editors, Integer Programming and Combinatorial Optimization, pages 182–196, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-72792-7.

[50] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–i. Math. Program., 14(1):265–294, dec 1978. ISSN 0025-5610. doi:10.1007/BF01588971. URL https://doi.org/10.1007/BF01588971.

[51] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions, volume 7, pages 234–243. 01 2006. ISBN 3-540-08708-7. doi:10.1007/BFb0006528.

[52] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions—II, pages 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978. ISBN 978-3-642-00790-3. doi:10.1007/BFb0121195. URL https://doi.org/10.1007/BFb0121195.

[53] Bin Du, Kun Qian, Christian Claudel, and Dengfeng Sun. Jacobi-style iteration for distributed submodular maximization. IEEE Transactions on Automatic Control, 67(9):4687–4702, 2022. doi:10.1109/TAC.2022.3180696.

[54] Nived Rajaraman and Rahul Vaze. Submodular maximization under A matroid constraint: Asking more from an old friend, the greedy algorithm. CoRR, abs/1810.12861, 2018. URL http://arxiv.org/abs/1810.12861.

[55] Angel Andres Daruna, Weiyu Liu, Zsolt Kira, and Sonia Chernova. Robocse: Robot common sense embedding. CoRR, abs/1903.00412, 2019. URL http://arxiv.org/abs/1903.00412.

[56] Youdi Gong, Guangzhen Liu, Yunzhi Xue, Rui Li, and Lingzhong Meng. A survey on dataset quality in machine learning. Information and Software Technology, page 107268, 2023. ISSN 0950-5849. doi:https://doi.org/10.1016/j.infsof.2023.107268. URL https://www.sciencedirect.com/science/article/pii/S0950584923001222.

[57] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. 12 2017.

[58] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 6950–6960. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

[59] A. M. Frieze. A cost function property for plant location problems. Math. Program., 7(1): 245–248, dec 1974. ISSN 0025-5610. doi:10.1007/BF01585521. URL https://doi.org/10.1007/BF01585521.

[60] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, 2, 2010.

[61] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[62] Adverse weather dataset. http://sar-lab.net/adverse-weather-dataset/.

[63] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. doi:10.1109/CVPR.2016.90.

[65] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8, 2023.

[66] Thibaut Horel. Notes on greedy algorithms for submodular maximization thibaut. 2015.