Generalized Synchronized Active Learning for Multi-Agent-Based Data Selection on Mobile Robotic Systems

Sebastian Schmidt, Lukas Stappen, Leo Schwinn, and Stephan Günnemann

***Abstract*—In mobile robotics, perception in uncontrolled environments like autonomous driving is a central hurdle. Existing active learning frameworks can help enhance perception by efficiently selecting data samples for labeling, but they are often constrained by the necessity of full data availability in data centers, hindering real-time, on-field adaptations. To address this, our work unveils a novel active learning formulation optimized for multi-robot settings. It harnesses the collaborative power of several robotic agents, considerably enhancing the data acquisition and synchronization processes. Experimental evidence indicates that our approach markedly surpasses traditional active learning frameworks by up to 2.5 percent points and 90% less data uploads, delivering new possibilities for advancements in the realms of mobile robotics and autonomous systems.**

***Index Terms*—Computer vision for transportation, deep learning for visual perception, deep learning methods.**

I. Introduction

In the rapidly advancing field of mobile robotics, perceiving the environment accurately is crucial to ensure safe robot operations and the effective execution of tasks. Deep neural networks often perform essential perception tasks like object detection or semantic segmentation, drawing upon data from different sensors, such as cameras and LIDAR sensors [1]. To ensure generalization across the input domain, robust algorithms [2] and large data quantities are needed [3], [4].

Active learning emerges as a promising technique to address this challenge, offering an intelligent approach to data sample selection for labeling, with the model autonomously determining the data it requires. This technique can be dissected further based on the scenario dictating data selection and the specific method employed in choosing the data. According to Settles [5], the existing scenarios are generally classified into stream-based and pool-based active learning. The former presents data as a continuous stream, necessitating immediate decisions on data selection or disposal, whereas the latter assumes the availability of all data at a centralized data center, accessible at any time. Query strategies typically fall into three categories: uncertainty-based, diversity-based, and learning-based, each with unique approaches to data selection and utilization [6]. Whereas uncertainty-based methods estimate the value for each sample individually, diversity-based and learning-based methods utilize both labeled and unlabeled data to either enhance dataset coverage or ascertain the estimated value of a sample using a trained model.

Although active learning holds considerable potential in the field of robotics, it also poses certain challenges in implementation. The prevailing pool-based scenario demands the centralization of the entire dataset to select samples for labeling, a requirement that often proves infeasible due to logistical and financial constraints. Although stream-based active learning is a viable alternative, current research in this domain inadequately addresses the multifaceted needs of robotic operations [7], especially in the context of perception tasks and multi-robot deployments. Furthermore, a significant gap exists in accommodating multiple data streams simultaneously, which is crucial when coordinating multiple robots.

Given the growing deployment of mobile robots in diverse fields such as autonomous driving, supply delivery, and autonomous hazard zone inspections, enhancing robotic perception remains a critical research frontier. Addressing this, our work aims to devise strategies for efficient label selection to facilitate improved perception, focusing particularly on fostering collaboration among multiple robots in exploring and annotating environments within an active learning framework. We introduce a novel hybrid scenario that integrates the strengths of stream-based and pool-based active learning, promoting effective multi-robot label collection and synchronization while enabling multi-stream processing.

Our contributions can be summarized as follows:

- We propose a comprehensive, generalized formulation for active learning scenarios, uniquely adapted to facilitate multi-agent data collection, setting the stage for more collaborative robotics systems.

Manuscript received 13 March 2024; accepted 16 July 2024. Date of publication 15 August 2024; date of current version 3 September 2024. This article was recommended for publication by Associate Editor Byeongkeun Kang and Editor Cesar Cadena Lerma upon evaluation of the reviewers' comments. *(Corresponding author: Sebastian Schmidt.)*

Sebastian Schmidt is with the Technical University of Munich, 80333 Munich, Germany, TUM School of Computation, Information and Technology, and also with the BMW Group, 80809 Munich, Germany (e-mail: sebastian95.schmidt@tum.de).

Lukas Stappen is with the BMW Group, 80809 Munich, Germany (e-mail: stappen@iee.org).

Leo Schwinn and Stephan Günnemann are with the Technical University of Munich, 80333 Munich, Germany, TUM School of Computation, Information and Technology (e-mail: l.schwinn@tum.de; s.guennemann@tum.de).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3444670, provided by the authors.

Project Page: https://www.cs.cit.tum.de/daml/smartl/

Digital Object Identifier 10.1109/LRA.2024.3444670

- Leveraging the aforementioned formulation, we introduce a pioneering framework for **S**ynchronized **M**ulti-**A**gent-**R**obotic Ac**T**ive **L**earning (**SMARTL**). This framework efficiently adapts and fuses data selections from multiple agents, optimizing both resource allocation and operational effectiveness.
- Through rigorous experimentation, we substantiate the versatility and efficacy of our newly devised framework, showcasing its superior performance in advancing multi-robotic active learning compared to existing stream- and pool-based methods.

In the subsequent sections, we provide a detailed exposition of these contributions, exploring the potential of the SMARTL framework in a multi-robotic environment.

II. Related Work

Research in the field of active learning has primarily focused on pool-based techniques. A limited number of studies have ventured beyond this to explore different facets of active learning, including alternative training strategies [8], and varied scenarios [7].

For sample selection, early adaption of uncertainty methods utilized sampling techniques such as Monte Carlo dropout [9] and ensembles [10], assessing the uncertainty across multiple forward passes or models using entropy or Bald [11] for each sample individually. To increase the effectiveness of batch selection, Kirsch et al. [12] introduced the joint uncertainty of the batches as a selection metric. Latter research directions employ Gaussian mixture models to asses uncertainty [13], [14]. Uncertainty methods have been applied and tailored to the tasks of classification [7], [15], object detection [16], [17], 3D object detection [18], [19], semantic segmentation [20] and graphs [21].

In contrast, diversity-based methods aim to represent the dataset distribution more accurately using a limited number of samples. A prominent representative of this category is CoreSet [22]. Subsequent studies have endeavored to integrate diversity and uncertainty metrics into a single cost function [23], [24]. Schmidt et al. [25] leveraged distance gradients for diversity-based selection. Following the combination of uncertainty and diversity, this field has given rise to task-agnostic active learning frameworks specifically designed for 3D object detection [26], [27]. Liang et al. [28] combined diversity metrics computed in latent space with diversity metrics in 3D space and time.

Learning-based methods utilize an auxiliary model or extension to determine the utility of a sample, distinguishing them significantly from the approaches mentioned above. Several approaches have been explored, including the introduction of loss prediction modules for ranking samples [29], the development of variational adversarial active learning [30], [31], [32], as well as teacher-student approaches [33], [34]. Caramalau et al. [35] structured the latent space as a graph and trained a graph convolutional network (GCN) to differentiate labeled and unlabeled samples. Their CoreGCN applies CoreSet on the GCN features, while UncertaintyGCN uses the uncertainty of the GCN prediction.

Despite advancements in the aforementioned categories, stream-based active learning remains relatively unexplored, especially in the realms of perception and robotics. The primary focus in this sector has been on distribution shifts [36], [37], [38], with significant theoretical contributions in submodular optimization [39], [40]. Perception has been approached with Mondrian forests [41]. In deep learning-based perception, uncertainty-based methods have been implemented on a single robot [42] or combined with submodular optimization [43]. In subsequent work, Schmidt and Günnemann [7] exploited the temporal characteristics of streams to enhance data selection. Saran et al. [44] introduced a approximate volume sampling in gradient space.

Beyond these areas of research, the emerging field of federated active learning aims to select data from multiple distributed data clients. On these clients, samples have been selected using probabilities [45] or individual local acquisitions [46]. Ahn et al. [47] compared the scores of the local models for a global data selection. Later, Kim et al. [48] expanded upon this by balancing the data subsets of the different clients. In robotics, active learning can be additionally used for control and reinforcement learning [49], whose methodologies differ substantially from their perception counterparts.

However, a significant research gap remains in tailoring these methods to multi-robot scenarios, where centralized training and data storage are the norm, but decentralized computation and storage are restricted. The current literature scarcely addresses active learning scenarios involving multiple agents outside the context of federated active learning.

III. Two Stage Active Learning With Distributed Agents

In this section, we first introduce the setting in which our SMARTL framework operates. From highlighting the limitations of contemporary approaches, we derive a novel generalized active learning formulation. Subsequently, we refine this concept to develop the SMARTL framework.

Preliminaries: Active learning fundamentally involves the selection of a subset $\mathcal{D}_L$ of a dataset $\mathcal{D}$ that should be labeled. The dataset $\mathcal{D}$ is assumed to be a sample from some (unknown) distribution of perception scenarios denoted as $\mathcal{D} \sim \Omega$. Concurrently, an unlabeled subset $\mathcal{D}_U$ exists, satisfying the condition $\mathcal{D}_L \cup \mathcal{D}_U = \mathcal{D}$. The overarching goal of this process is evaluated by the performance of a trained model $F(\omega|\mathcal{D}_L)$ parameterized by its weights ω while maintaining a minimal size for D_L. The active learning process is iterative, encompassing N cycles, wherein each cycle i selects a new batch $\mathcal{D}_l$ of size b. A pool-based query Q_P leverages the current label $\mathcal{D}_L^i$ and unlabeled set $\mathcal{D}_U^i$ as well as the model weights to select the new batch $\mathcal{D}_l$ of size b for annotation:

$$\mathcal{D}_l^{i+1} = Q_P(\mathcal{D}_U^i, \mathcal{D}_L^i, \omega^i, b) \tag{1}$$

$$\mathcal{D}_L^{i+1} = \mathcal{D}_L^i \cup Q_P(\mathcal{D}_U^i, \mathcal{D}_L^i, \omega^i, b) \tag{2}$$

$$\mathcal{D}_U^{i+1} = \mathcal{D}_U^i \setminus \mathcal{D}_l^{i+1} \tag{3}$$

$$\mathcal{D}_L = \bigcup_{i=0}^{n} Q_P(\mathcal{D}_U^i, \mathcal{D}_L^i, \omega^i, b) \tag{4}$$

The pool-based scenario assumes all data to be present in the data center such that a full-scale selection can be conducted, including diversity-based and learning-based methods relying on extensive calculations using all unlabeled and labeled data.

In robotics, unlike other applications where data is primarily stored in a data center, data is initially stored on the robot during the collection and operational phase. If the dataset is not considered to be comprehensive before the robot's operation commences, recorded data from an operation $\mathcal{D}_O^t$ is added to the data pool, such that $\mathcal{D}_U^{i+1} = \mathcal{D}_U^i \cup \mathcal{D}_O^t \setminus \mathcal{D}_l^{i+1}$, where $\mathcal{D}_O^t \sim \Omega$. For the sake of simplification, we presuppose that the operations t are synchronized with the cycles i, thus $i = t$.

However, this scenario has significant downsides for mobile robotics. It assumes the transfer of all data to the data center, which may not always be feasible, depending on the type of data transfer. Additionally, storing all data during the operation puts additional hardware requirements in place.

In contrast to the pool-based scenario, the stream-based scenario dictates that the samples are not retained. Instead, they are either earmarked within an internal storage of size b in the streambatch scenario [7], [43] or instantaneously marked for labeling. Since the samples are not stored, the query function $Q_S(\overrightarrow{\mathcal{D}}_U, \omega, b)$ is restricted to evaluate each sample once, indicated with an arrow $\overrightarrow{\mathcal{D}}_U$. This method has the advantage of facilitating selection online at the mobile operator level, ensuring only necessary data is saved. Nonetheless, this assumes a single agent overseeing data collection and cannot reflect multiple robots.

Multi-Agent Formulation: To address the aforementioned challenges, we introduce a hybrid two-stage scenario that scales to multiple agents by effectively preventing information overlap among various agents and simultaneously minimizing the requirements for data storage and uploads. It is crucial to consider that mobile robots frequently operate in non-isolated environments and may receive overlapping information. In this setup, we consider j agents engaged in data collection. Given the impracticability of storing and uploading all the data collected by a large number of agents, a stream-based selection becomes indispensable. Each agent j encounters in each cycle i a sensor stream, denoted as $\overrightarrow{\mathcal{D}}_O^{i,j}$, embodying potential candidates for labeling.

To mitigate the data upload problem, a subset $\mathcal{D}_c^{i,j} \subset \overrightarrow{\mathcal{D}}_O^{i,j}$ of size b_s^j for each agent j is selected on the agent, which can be expressed for a single cycle i as follows:

$$\mathcal{D}_c^{i,j} = Q_S(\overrightarrow{\mathcal{D}}_O^{i,j}, \omega^i, b_s^j) \tag{5}$$

$$\mathcal{D}_C^i = \bigcup_{j=1}^{m} Q_S(\overrightarrow{\mathcal{D}}_O^{i,j}, \omega^i, b_s^j); \quad b_s = \sum_{j=1}^{m} b_s^j \tag{6}$$

Upon aggregating the uploaded data $\mathcal{D}_C^i$, we encounter a predicament where the information content from two distinct agents overlaps, implying $\mathcal{D}_c^{i,j} \cap \mathcal{D}_c^{i,j'} \neq \emptyset, \forall i \in N$. To resolve this, we delineate a secondary stage executed at the data center to select b_p samples by defining $\mathcal{D}_C^i$ as an addition for the unlabeled pool comprehensively defined as:

$$\mathcal{D}_l^{i+1} = Q_\text{p}(\mathcal{D}_C^i, \mathcal{D}_L^i, \omega^i, b_p) \tag{7}$$

$$\mathcal{D}_U^{i+1} = \mathcal{D}_U^i \cup \mathcal{D}_C^i \setminus \mathcal{D}_l^{i+1} \tag{8}$$

$$\mathcal{D}_L = \bigcup_{i=0}^{n} Q_\text{p}(\mathcal{D}_C^i \cup \mathcal{D}_U^i, \mathcal{D}_L^i, \omega^i, b_p) \tag{9}$$

TABLE I
Parameter Sets for Our Generalized Active Learning Formulation to Describe Different Active Learning Scenarios

Scenario	Parameters
Pool [5]	$Q_S = Q_I, \mathcal{D}_C^i = \emptyset$
Stream [5]	$Q_P = Q_I, \mathcal{D}_U = \overrightarrow{\mathcal{D}}_O^i, b_s = 1$
StreamBatch [7]	$Q_P = Q_I, \mathcal{D}_U = \overrightarrow{\mathcal{D}}_O^i$
PoolStream [7]	$Q_S = Q_I$

Combing the onboard and data center selections given in (6), (9), the full framework composes to:

$$\mathcal{D}_L = \bigcup_{i=0}^{n} Q_P\left(\bigcup_{j=1}^{m} Q_S(\overrightarrow{\mathcal{D}}_O^{i,j}, \omega^i, b_s) \cup \mathcal{D}_U^i, \mathcal{D}_L^i, \omega^i, b_p\right) \tag{10}$$

Through this formulation, we have created two distinct stages represented by Q_P and Q_S. The query function Q_S operates in real-time, independently on each robotic agent. Conversely, Q_P functions centrally at the data center, orchestrating the subsequent layer of data processing and selection. A visual representation of our framework is shown in Fig. 1. The illustrated two-phase query process begins with the stream-based query Q_S on the agents (1) and the upload of $\mathcal{D}_C$ from the first selection phase (2). Afterward, follows the pool-based query Q_P on the data center (3). To close the cycle, the data $\mathcal{D}_l$ is sent to a human annotator (4) and added to the labeled pool (5). Subsequently, the network is trained with the updated labeled pool (6), and the agents are synchronized with the newly refined model (7).

Generalized Active Learning Formulation: In our pursuit to establish (10) as a generalized expression of active learning, we introduce a fundamental component, the identity query Q_I. This query function is designed to select all data, thereby satisfying $S = Q_I(S)$ for any dataset S. Consequently, this identity element permits the representation of both stream-based or pool-based scenarios [5], [7] as special instances encapsulated within (10), underlining the entitlement as generalized scenario formulation. To substantiate this claim, we provide the corresponding parameters to generate the plurality of active learning scenarios in Table I.

Framework Definition: Enabled by the generalized active learning formulation, we introduce a comprehensive and adaptable hybrid scenario active learning framework:

Stage 1 - Mobile Operator Data Stream Processing: This stage primarily concerns onboard data streams from sources like cameras and LiDAR sensors on the robots. Each robot or agent acts individually in this stage, only involving the deployed perception neural network. Given the stream-based setup inherent to online robotics selection, the query function at this

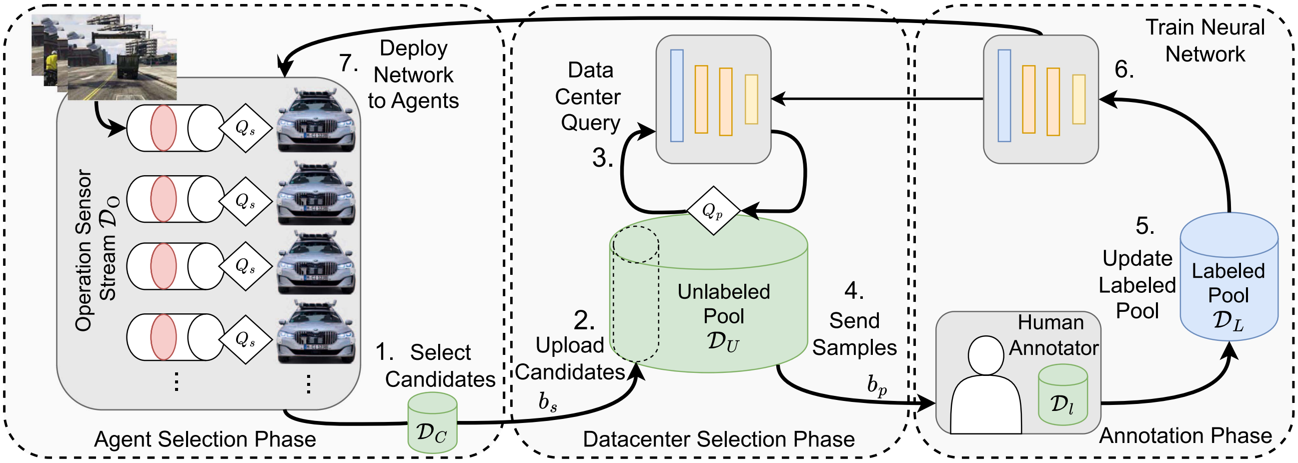


Fig. 1. Multi-Robot Active Learning Framework. **Left**: Selection on the individual robots. **Middle**: query to synchronize and filter the uploaded samples. **Right**: send labels to human annotator for labeling, update training set, and train neural network.

stage is restricted to evaluating only the current sample, which limits them to loss learning and uncertainty-based methods like Monte Carlo dropout. However, SMARTL is agnostic among this group of methods, enabling flexibility in the first stage. We highlight different selection methods in Section IV. As this can cause overlaps in the selection, we set $b_s > b_p$ to collect more uncertain samples for succeeding diversification.

Stage 2 - Data Center Pool Processing: Subsequent to the mobile operator stage, a post-operation initiates a pool-based selection and synchronization phase. After the robots upload b_s samples, a diversity-based method is employed to accurately reflect the relationship between the newly acquired data and the data previously uploaded to the system, optimizing coverage over the entire dataset. The diversity-based state queries b_p samples, so possible information overlaps of samples are mitigated. This can be achieved by minimizing a set of latent space features f using a pairwise distance function d. In our approach, we adopt the greedy L2 distance metric of the CoreSet approach [22] and enrich it with metadata, which has been successfully presented by Liang et al. [28]. The metadata contains the information about the selecting robot and whether it is in the labeled or unlabeled pool. We concatenate the normalized metadata to feature vectors f to reflect the relationship of samples collected by one robot in contrast to those collected by different agents.

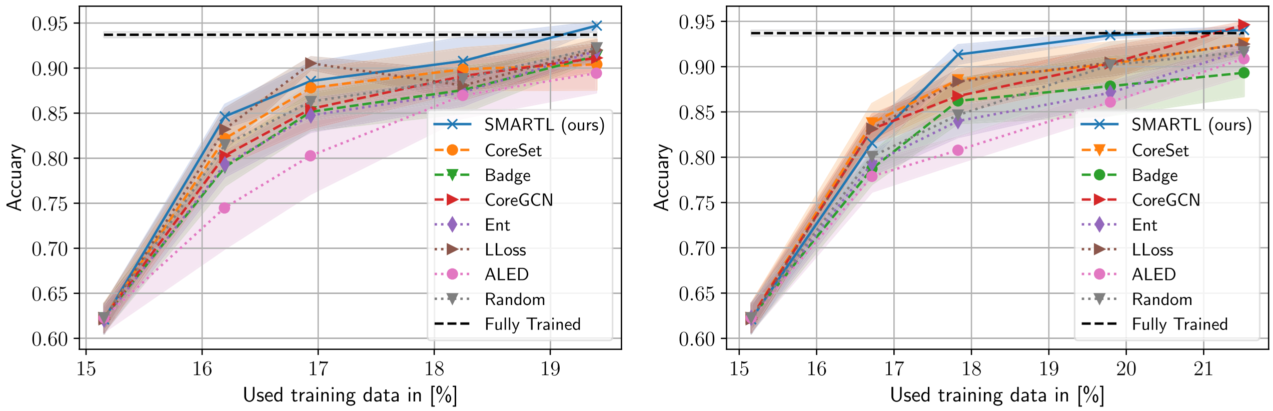
Application Example: Consider a large-scale active learning scenario with hundreds of autonomous vehicles, representing an exemplary mobile robot type, acting as agents. In this scenario, each vehicle is equipped with a *computer vision system* comprising a *camera sensor* and a *computing unit* running a deep neural network.

Implementation of SMARTL: Importantly, SMARTL is agnostic with respect to the applied uncertainty estimation technique. It allows employing methods based on minor neural network modifications [7], [29] or multiple [9], or temporal network forward passes [17] that are most feasible for the existing computing unit and task. We will highlight this flexibility in the experiments and refer to original works for details on implementation. Additionally, data storage for b_s samples is required. As SMARTL has no real-time requirements, the samples can be uploaded via any protocol or transferred via physical devices at hubs. This flexibility ensures that SMARTL operates without any latency dependencies. In our example, we assume that collected data is transferred daily. Furthermore, the number of agents can be adjusted dynamically, as data collection is decentralized with inter-robot dependencies, enhancing scalability. Since SMARTL modifies only the data collection phase, it does not affect the data labeling processes used in traditional frameworks.

Relation to existing frameworks: Traditional active learning frameworks primarily assume the presence of all data on a data center, requiring each agent to store all data collected and subsequently transfer it to a centralized data center for selection. Given the use of high-definition cameras, this approach could demand several GB per hour per agent, resulting in substantial data storage and transfer challenges. *SMARTL* considers a selection and collection outside the data center, enabling data selection at the individual agent level. This approach removes the need to gather all data at a centralized data center before the selection process and eliminates most storage and transfer costs. The second selection phase on the data synchronizes the selection of the individual agents and removes overlaps, leveraging scalability.

IV. Evaluation of Two Stage Active Learning

In the subsequent section, we evaluate the performance of our hybrid active learning framework. To showcase the versatility of our approach, we conduct experiments involving various tasks and datasets. We evaluate classification with the GTA V streets (GTAVs) dataset [7] and CIFAR-100 [50] to validate our approach's efficacy for environments with a larger number of classes. Furthermore, we validate the semantic segmentation task on A2D2 [51] and CityScapes [1]. Given that distributed multi-agent collection stands as a central pillar of our framework, we concurrently process multiple streams during each active learning cycle as depicted in Fig. 1.



(a) GTAVs ResNet18 no overlap

(b) GTAVs ResNet18 overlap

Fig. 2. Comparison of various active learning configurations on GTAVs, highlighting standard errors. Dotted lines represent stream-based selections, while dashed lines indicate pool-based methods that upload all data.

TABLE II

Comparison of Upload Frequencies Across Different Methods for Classification Experiments Reveals That Our SMARTL , Functioning as a Hybrid Model, Requires Significantly Fewer Uploads Than Pool-Based Methods

	Method	Cycle 1	Cycle 2	Cycle 3	Cycle 4
GTAV$_{S0}$	SMARTL	562	400	708	622
	Pool-based	5620	4000	7080	6220
	Stream-based	281	200	354	311
CIFAR	SMARTL	5000	5000	5000	5000
	Pool-based	10000	10000	10000	10000
	Stream-based	2500	2500	2500	2500

We postulate the existence of a set of mobile robots engaged in data acquisition from multiple streams, with subsequent synchronization occurring at the data center. This scenario is compared to a purely pool-based data center selection, as well as isolated steam-based selection processes undertaken by individual robots. Importantly, we monitor the data saved during the data upload process, as minimizing the volume of uploads constitutes a relevant objective of our methodology.

In our experiments, we compare our method against several baselines. As other methods cannot leverage a two-stage selection, we added the identity query Q_I in the data center phase for stream-based selections and in the agent selection phase for pool-based methods. Note that this grants a considerable advantage of a full data upload for pool-based methods. We consider the **Badge** [23], **CoreSet** [22], **CoreGCN** [35] and **UncertaintyGCN** [35] as pool-based baselines. As stream-based baselines, we consider active learning on edge devices (**ALED**) [43], Monte Carlo dropout Entropy (**Ent**) [9] and **Bald**, loss learning (**LLoss**) [29] and a **Random** selection.

Classification: For the classification task, we utilize a ResNet18 [52] model and the GTAVs dataset [7], specifically designed for operation domain detection. This dataset encompasses seven distinct routes. In our setup, ten selection agents are deployed, each encountering an alternating segmented subset of the current route. We use the setting and routes presented in [7], but start the early stopping after 50 epochs. Given its unstructured nature, which is atypical in robotic applications, the CIFAR-100 dataset is partitioned randomly. This deliberate selection compensates for the limited class variety encountered in operation domain detection datasets. For CIFAR-100, we followed the parameters in [53] that are provided in the benchmark repository.[1] We additionally used the augmentations LLoss guidelines established for CIFAR-10 in [29]. During each cycle, 10000 samples are presented in partitions of 1000 to each agent in a stream-based manner, with a selection size of $b_p = 25\%$. In contrast, the GTAVs dataset saw a more conservative selection of $b_p = 5\%$ per cycle, adhering to the experimental setup described in [7]. We choose our mobile operator selection size to $b_s = 2 \cdot b_p$ and all b_s^j to be equal. The robot executes a LLoss with modifications of [7] for GTAVs and an Ent query for CIFAR-100. We conduct two distinct experiments to simulate the behavior of autonomous agents traversing overlapping routes: One scenario with no overlap (0%) and another with significant overlap, constituting 30% of the routes. For an overlap, we appended samples of the previous and subsequent subsets of one robot to that of another robot, which increased the stream size and, consequently, the number of selected samples.

The experiments conducted on the GTAVs dataset, as illustrated in Fig. 2, demonstrate that our method achieves stronger performance with fewer queries and matches the performance of the fully trained model earlier. While the LLoss with the modifications of [7] exhibits good performance in the second cycle, the lack of robot synchronization caused the performance to drop again. For the overlap experiment in Fig. 2(b), a noticeable increase in performance is observed, as our method attains a performance level comparable to the CoreGCN approach. Furthermore, a review of the data uploads presented in Table II reveals that our SMARTL significantly reduces the number of uploads by 90% for the GTAVs dataset. It should be noted that since all uploaded samples are available for pool-based methods, these techniques can capitalize on interesting scenarios that might be overlooked by methods that select directly on the

[1] [Online]. Available: https://github.com/weiaicunzai/pytorch-cifar100

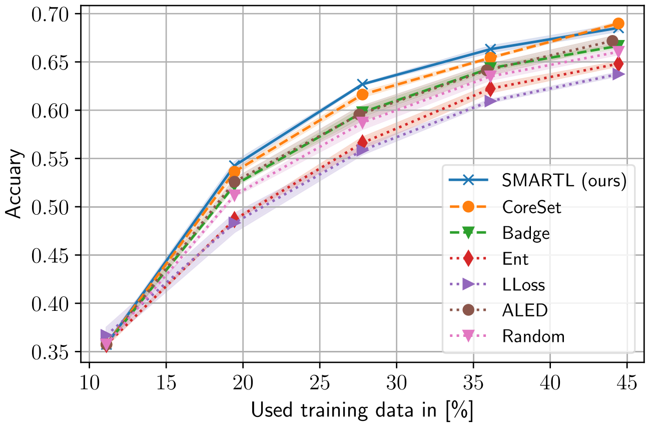


Fig. 3. CIFAR-100 with overlap on ResNet18, with highlighting standard errors. Stream-based selections in dotted, pool-based methods that upload all data in dashed.

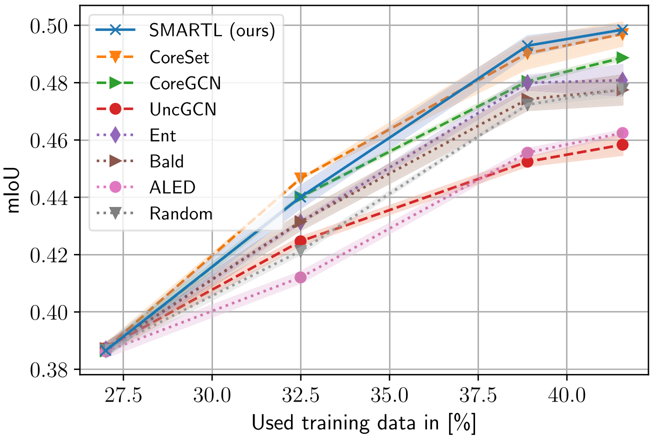


Fig. 4. Active learning comparison on A2D2. Stream-based selections as dotted lines, pool-based methods dashed.

robot. As the $\mathcal{D}_l$ is limited by b_p, stream-based methods can reduce uploads compared to SMARTL.

As depicted in Fig. 3, our SMARTL consistently exhibits robust performance throughout all cycles on CIFAR-100, with only CoreSet marginally outperforming it in the final cycle. Notably, CoreGCN encountered memory issues while handling this dataset, a drawback attributable to its reliance on distance-based calculations, which become intractable for a larger amount of uploaded samples. This scenario highlights the versatility of our framework, which is adept at accommodating multi-class datasets - a theoretical exploration that holds promising relevance for diverse and practical applications in robotics.

Our ambition is not to overshadow existing state-of-the-art performances but to align with them, showcasing efficiency through a significant reduction in the number of necessary uploads. In addition to assessing the commonly evaluated metric of performance gain in an active learning setup, we are also keen on scrutinizing the volume of data transfer in order to furnish a comprehensive analysis of our system's efficacy and efficiency.

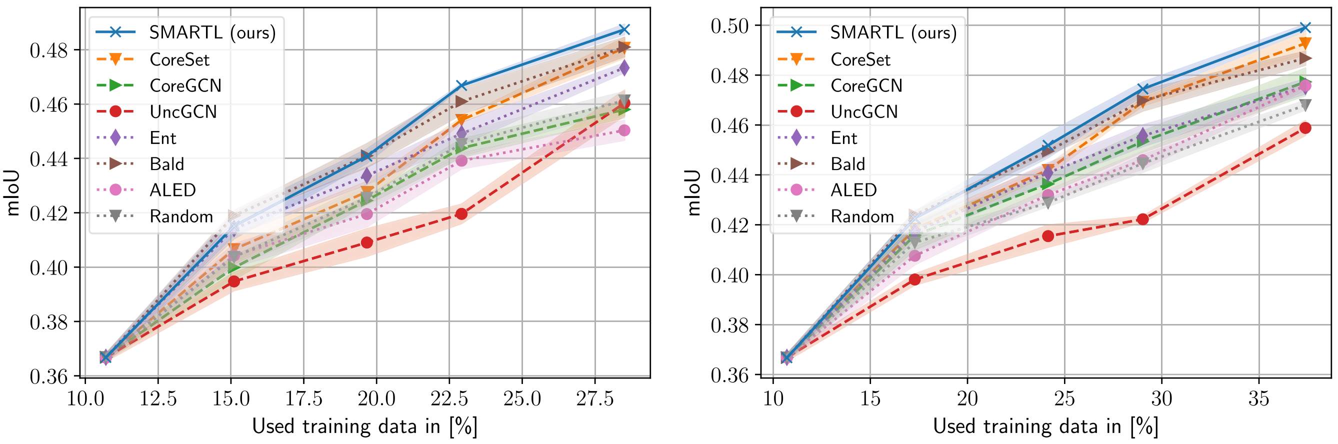
Semantic Segmentation: To validate our framework on a more challenging task, we use the CityScapes [1] dataset and A2D2 [51]. Both datasets contain different drives in different cities. We assume four and three robots operate concurrently in separate cities or drives for CityScapes and A2D2, respectively. For our experiments, we used a DeepLabV3+ [54] model outfitted with a ResNet34 backbone, utilizing pre-trained weights provided by PyTorch. Regarding the Cityscapes dataset, we adhered to the parameter settings outlined in [55], with the exception of the resize crop parameters, where we adjusted the factor to a range of 0.5 to 1 and set the resized dimensions to 256 × 512. For A2D2, we followed the guidelines established in [7] but decreased the learning rate to 0.01 for the backbone. We use an auxiliary loss with an FCN head at the third ResNet34 block to improve the diversity-based selection within our data center query, delivering the diversity calculation features. Our framework utilizes Monte Carlo Dropout Ent for Q_S in the A2D2 dataset and BALD for the Cityscapes dataset with max accumulation. We further select $b_p = 20\%$ of each stream with $b_s = 2 \cdot b_p$. As for classification, we conduct two different experiments where we model the behavior of overlapping routes of the autonomous agent: One with 0% overlap and one with 30% overlap between the agents.

TABLE III
COMPARISON OF UPLOAD FREQUENCIES ACROSS DIFFERENT METHODS FOR CITYSCAPES (CS) AND A2D2 HIGHLIGHTING THE UPLOAD EFFICIENCY OF SMARTL AGAINST POOL-BASED METHODS

	Method	Cycle 1	Cycle 2	Cycle 3	Cycle 4
CS_0	SMARTL	222	230	164	282
	Pool-based	555	575	410	705
	Stream-based	111	115	82	141
A2D2	SMARTL	680	792	332	-
	Pool-based	1700	1980	830	-
	Stream-based	340	396	166	-

In Fig. 5, we present the evaluation of our experiments conducted on the CityScapes dataset. As depicted in Fig. 5(a), our method demonstrates superior performance, improving faster and surpassing all other methods. This is particularly evident when compared to stream-based selections, where our approach excels in effectively synchronizing data. However, in the overlap experiments depicted in Fig. 5(b), the overlap weakens the individual robots' selection process, narrowing the performance gap between our method and the CoreSet approach, which uploads all data. In our A2D2 experiments, we observed similar trends, substantiating our initial findings. Notably, A2D2 offers a larger pool of selectable samples, 4510, compared to CityScapes' 2245. In Fig. 4, it is evident that while our SMARTL approach initiates with a slightly subdued performance compared to pool-based methods, it gains momentum over subsequent cycles, ultimately eclipsing the performance of all other contenders. Conversely, the stream-based approaches lag, showcasing a weaker performance trajectory. Since A2D2 recordings around Munich are not overlap-free, SMARTL confirms its robustness in the overlap scenario.

Upload Volume Analysis: To examine the influence on the different query sizes b_s and b_p, we compare different rations in Table IV. Our results indicate that the sizes balance



(a) CityScapes

(b) CityScapes with Overlap

Fig. 5. Comparison of different active learning methods - mIoU over used data with indicated standard errors. Dotted lines represent stream-based selections, while dashed lines indicate pool-based methods that upload all data.

TABLE IV
PERFORMANCE (MIOU) COMPARISON OF DIFFERENT SELECTION BUDGETS FOR CITYSCAPES (CS) AND A2D2. STANDARD ERROR TO THE LAST DIGIT IN BRACKETS

	Query Size	Cycle 1	Cycle 2	Cycle 3	Cycle 4
CS_0	$b_s = 1.5 \cdot b_p$	0.418 (2)	0.446 (1)	0.470 (3)	0.491 (2)
	$b_s = 2 \cdot b_p$	0.415 (5)	0.441 (3)	0.467 (1)	0.487 (2)
	$b_s = 3 \cdot b_p$	0.412 (3)	0.432 (1)	0.457 (2)	0.492 (3)
A2D2	$b_s = 1.5 \cdot b_p$	0.435 (1)	0.487 (3)	0.497 (1)	-
	$b_s = 2 \cdot b_p$	0.440 (4)	0.492 (3)	0.498 (2)	-
	$b_s = 3 \cdot b_p$	0.444 (2)	0.494 (3)	0.499 (5)	-

between uncertainty- and diversity-based selection in the different phases. The b_s can be reduced for CityScapes with more diverse and overlap-free routes, while for the overlapping A2D2, a higher b_s leads to a performance increase. Following this SMARTL can effectively combine the strengths of both uncertainty- and diversity-based strategies.

Our experiments demonstrate that the distributed structure of our approach seamlessly integrates the uncertainty and diversity components of active learning query functions. SMARTL effectively reduces the number of required data uploads, as shown in Table III, while maintaining the highest performance over all experiments. This preserves even in scenarios with overlapping operations. Due to the absence of inter-robot dependency SMARTL is seamlessly scalable to any number of robots. The openness in the uncertainty stage and the embedding space distance calculation enabled SMARTL to outperform existing methods on various tasks on established benchmark datasets. As these characteristics are shared for all computer vision tasks SMARTL is not limited to presented tasks.

V. Conclusion

In this study, we have introduced *SMARTL*, a pioneering active learning framework that facilitates simultaneous data collection from multiple mobile agents, addressing the existing gaps in modeling multi-agent active learning scenarios. By deriving a general active learning formulation, we pave the way for novel active learning scenarios. A central feature of our SMARTL framework is its ability to strategically branch out the active learning query into two components: an uncertainty-based query managed by the robot and a diversity-based query conducted at the data center. This division allows for a scalable integration with an arbitrary number of agents, with minimal adaption of the perception model. The flexibility in the uncertainty estimation component allows task-specific adaption, amplifying scalability.

Our extensive experiments underscore the framework's efficacy, demonstrating a significant reduction in data upload volumes while surpassing the label selection performance of conventional pool-based active learning methods, necessitating complete data transfer to the data center. In future work, we plan to expand the capabilities of the diversity-based query component by incorporating metadata more extensively to further capitalize on the distributed nature of multi-robot setups. Besides, we intend to investigate robot synchronization to tackle overlapping routes and include an unsupervised learning phase in our framework.

References

[1] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

[2] L. Schwinn et al., "Improving robustness against real-world and worst-case distribution shifts through decision region quantification," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 19434–19449.

[3] T. Altstidl, D. Dobre, B. Eskofier, G. Gidel, and L. Schwinn, "Raising the bar for certified adversarial robustness with diffusion models," 2023, *arXiv:2405.01462*.

[4] B. Nieth, T. Altstidl, L. Schwinn, and B. Eskofier, "Large-scale dataset pruning in adversarial training through data importance extrapolation," 2024, *arXiv:2406.13283*.

[5] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Tech. Rep., 2010. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60660

[6] P. Ren et al., "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, pp. 1–40, 2022.

[7] S. Schmidt and S. Günnemann, "Stream-based active learning by exploiting temporal properties in perception with temporal predicted loss," in *Proc. Britisch Mach. Vis. Conf.*, 2023. [Online]. Available: https://proceedings.bmvc2023.org/664/

[8] F. Dayoub, N. Sunderhauf, and P. I. Corke, "Episode-based active learning with bayesian neural networks," in *Proc. CVF/IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 498–500.

[9] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. 33th Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[10] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, “The power of ensembles for active learning in image classification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9368–9377.

[11] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, “Bayesian active learning for classification and preference learning,” 2011, *arXiv:1112.5745*.

[12] A. Kirsch, J. V. Amersfoort, and Y. Gal, “BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning,” in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 7026–7037.

[13] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, “Active learning for deep object detection via probabilistic modeling,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10264–10273.

[14] J. Mukhoti, A. Kirsch, J. V. Amersfoort, P. H. S. Torr, and Y. Gal, “Deep deterministic uncertainty: A new simple baseline,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 24384–24394.

[15] L. Schwinn et al., “Identifying untrustworthy predictions in neural networks by geometric gradient analysis,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2021, pp. 854–864.

[16] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, “Deep active learning for efficient training of a LiDAR 3D object detector,” in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 667–674.

[17] S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, “Advanced active learning strategies for object detection,” in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 871–876.

[18] A. Hekimoglu, M. Schmidt, A. Marcos-Ramiro, and G. Rigoll, “Efficient active learning strategies for monocular 3D object detection,” in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 295–302.

[19] Y. Park, W. Choi, S. Kim, D.-J. Han, and J. Moon, “Active learning for object detection with evidential deep learning and hierarchical uncertainty aggregation,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://iclr.cc/virtual/2023/poster/11241

[20] P. Y. Huang, W. T. Hsu, C. Y. Chiu, T. F. Wu, and M. Sun, “Efficient uncertainty estimation for semantic segmentation in videos,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 520–535.

[21] D. Fuchsgruber, T. Wollschläger, B. Charpentier, A. Oroz, and S. Günnemann, “Uncertainty for active learning on graphs,” in *Proc. Int. Conf. Learn. Representations*, 2024.

[22] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1aIuk-RW

[23] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, “Deep batch active learning by diverse, uncertain gradient lower bounds,” in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ryghZJBKPS

[24] J. T. Ash, S. Goel, A. Krishnamurthy, and S. Kakade, “Gone fishing: Neural active learning with fisher embeddings,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8927–8939.

[25] S. Schmidt, L. Schenk, L. Schwinn, and S. Günnemann, “A unified approach towards active learning and out-of-distribution detection,” 2024, *arXiv:2405.11337*.

[26] C. Yang, L. Huang, and E. J. Crowley, “Plug and play active learning for object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 17784–17793.

[27] Y. Luo, Z. Chen, Z. Wang, X. Yu, Z. Huang, and M. Baktashmotlagh, “Exploring active 3D object detection from a generalization perspective,” in *Proc. Int. Conf. Learn. Representations*, 2023.

[28] Z. Liang, X. Xu, S. Deng, L. Cai, T. Jiang, and K. Jia, “Exploring diversity-based active learning for 3D object detection in autonomous driving,” 2022, *arXiv:2205.07708*.

[29] D. Yoo and I. S. Kweon, “Learning loss for active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 93–102.

[30] S. Sinha, S. Ebrahimi, and T. Darrell, “Variational adversarial active learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5972–5981.

[31] K. Kim, D. Park, K. I. Kim, and S. Y. Chun, “Task-aware variational adversarial active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8166–8175.

[32] B. Zhang, L. Li, S. Yang, S. Wang, Z.-J. Zha, and Q. Huang, “State-relabeling adversarial active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8756–8765.

[33] F. Peng, C. Wang, J. Liu, Z. Y. Noah, and A. Lab, “Active learning for lane detection: A knowledge distillation approach,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15152–15161.

[34] A. Hekimoglu, M. Schmidt, and A. Marcos-Ramiro, “Monocular 3D object detection with LiDAR guided semi supervised active learning,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2024, pp. 2346–2355.

[35] R. Caramalau, B. Bhattarai, and T.-K. Kim, “Sequential graph convolutional network for active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9583–9592.

[36] Ł. Korycki, A. Cano, and B. Krawczyk, “Active learning with abstaining classifiers for imbalanced drifting data streams,” in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2334–2343.

[37] T. Pham, D. Kottke, G. Krempl, and B. Sick, “Stream-based active learning for sliding windows under the influence of verification latency,” *Mach. Learn.*, vol. 111, no. 6, pp. 2011–2036, Jun. 2022.

[38] H. Zhang, W. Liu, and Q. Liu, “Reinforcement online active learning ensemble for drifting imbalanced data streams,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3971–3983, Aug. 2022.

[39] K. Fujii and H. Kashima, “Budgeted stream-based active learning via adaptive submodular maximization,” in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 514–522. [Online]. Available: https://dl.acm.org/doi/10.5555/3157096.3157154

[40] E. Kazemi, M. Mitrovic, M. Zadimoghaddam, S. Lattanzi, and A. Karbasi, “Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3311–3320.

[41] A. Narr, R. Triebel, and D. Cremers, “Stream-based active learning for efficient and adaptive classification of 3D objects,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 227–233.

[42] L. Puck et al., “Distributed active learning for semantic segmentation on walking robots,” in *Proc. IEEE 20th Int. Conf. Adv. Robot.*, 2021, pp. 461–467.

[43] Y. Senzaki and C. Hamelain, “Active learning for deep neural networks on edge devices,” 2021, *arXiv:2106.10836*.

[44] A. Saran, S. Yousefi, A. Krishnamurthy, J. Langford, and J. T. Ash, “Streaming active learning with deep neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 30005–30021.

[45] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, “Active federated learning,” 2019, *arXiv:1909.12641*.

[46] J. Qian, S. Sengupta, and L. K. Hansen, “Active learning solution on distributed edge computing,” 2019, *arXiv:1906.10718*.

[47] J.-H. Ahn, Y. Ma, S. Park, and C. You, “Federated active learning (F-AL): An efficient annotation strategy for federated learning,” *IEEE Access*, vol. 12, pp. 39261–39269, 2024, doi: 10.1109/ACCESS.2024.3376746.

[48] S. Kim, S. Bae, H. Song, S.-Y. Yun, and K. Ai, “Rethinking federated active learning based on inter-class diversity,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3944–3953.

[49] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, “Active learning in robotics: A review of control principles,” *Mechatronics*, vol. 77, 2021, Art. no. 102576.

[50] A. Krizhevsky, V. Nair, and G. Hinton, “Learning multiple layers of features from tiny images,” Canadian Institute for Advanced Research, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[51] J. Geyer et al., “A2D2: Audi autonomous driving dataset,” 2020, *arXiv:2004.06320*.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[53] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” 2017, *arXiv:1708.0455v2*.

[54] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[55] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, *arXiv:1706.05587*.