Bayesian Active Learning for Sim-to-Real Robotic Perception

Feng, Jianxiang; Lee, Jongseok; Durner, Maximilian; Triebel, Rudolph

Bayesian Active Learning for Sim-to-Real Robotic Perception

Jianxiang Feng[1,2], Jongseok Lee[1], Maximilian Durner[1,2] and Rudolph Triebel[1,2]

Abstract— While learning from synthetic training data has recently gained an increased attention, in real-world robotic applications, there are still performance deficiencies due to the so-called Sim-to-Real gap. In practice, this gap is hard to resolve with only synthetic data. Therefore, we focus on an efficient acquisition of real data within a Sim-to-Real learning pipeline. Concretely, we employ deep Bayesian active learning to minimize manual annotation efforts and devise an autonomous learning paradigm to select the data that is considered useful for the human expert to annotate. To achieve this, a Bayesian Neural Network (BNN) object detector providing reliable uncertainty estimates is adapted to infer the informativeness of the unlabeled data. Furthermore, to cope with misalignments of the label distribution in uncertainty-based sampling, we develop an effective randomized sampling strategy that performs favorably compared to other complex alternatives. In our experiments on object classification and detection, we show benefits of our approach and provide evidence that labeling efforts can be reduced significantly. Finally, we demonstrate the practical effectiveness of this idea in a grasping task on an assistive robot.

I. INTRODUCTION

Over the last years, the performance of computer vision increased sharply, leading to the urge of employing such approaches on robotic vision tasks such as object classification, detection [1], [2] and pose estimation [3]. In this context, the necessity of large amounts of annotated, task-related training data is a main issue, particularly for tasks relying on semantic features such as object classification or detection. Therefore, a compelling solution is to learn from synthetic data. Like this, large amount of annotated data can be obtained from simulation with relatively less time and manual efforts [4]–[6]. With the emergence of open-source image synthesizing pipelines [7], [8], this solution becomes even more accessible in practice. However, although these pipelines continue improving in fidelity and become more photo-realistic, there are subtle but important differences between simulation and real domain. This leads to the so-called *Sim-to-Real gap* which is the main barrier to transfer this technique to real world robotic perception. Several works address this gap by applying techniques such as Domain Randomization (DR) [3], [6] and Domain Adaptation (DA) [4], [9] with certain improvements. Yet, the unpredictable variability of real-world scenes prevents a complete elimination of the reality gap [10].

We encounter similar issues in our real lab environment, when deploying an object detector [1] trained on photo-realistic images on our robotic platform EDAN [11]. From our practical experience, variables such as sensor characteristics, illumination, or textures cannot be modeled to precisely match the real environments. Even after a careful tuning of DR, we find that the object detector fails to generalize well
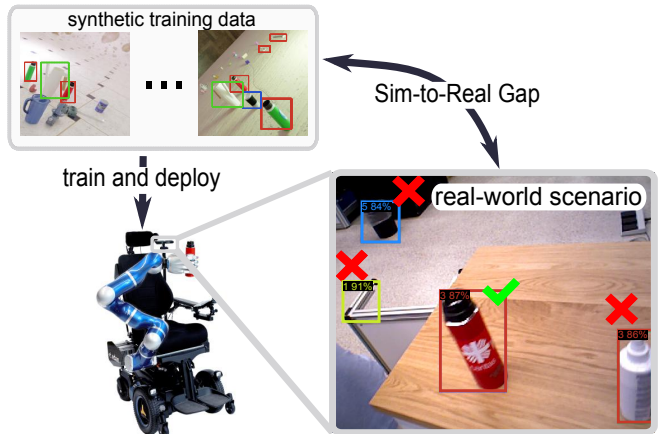

Fig. 1. **Illustration of a practical problem.** Deploying a detector trained with only synthetic images on real-world scenarios leads to under-performances. These inaccuracies (denoted by red crosses) such as false positives are due to the Sim-to-Real gap and for this, a few informative real images can improve the performance. Therefore, this work investigates the question on how to collect such informative real images via active learning.

in the real-world scenario (e.g., clutter in lab environment see Fig. 1). To overcome this, we applied domain-oriented fine-tuning, by using real data of the underlying robotic application, like [12]. Hence, based on our use-case, we advocate that real data is indispensable for a robot to robustly adapt from simulation to real world.

This however, comes with the requirement of tedious, time consuming manual labeling. In this work, we investigate on the question: *How to bridge the Sim-to-Real gap with minimum annotation efforts?* Having a model trained on synthetic images, we propose an Active Learning (AL) pipeline that can efficiently bridge the still present Sim-to-Real gap. In contrast to our previous work [13], we here aim for autonomous acquisition of as few annotated real images as possible. Based on a deep Bayesian Active Learning (AL) framework [14], [15], we analyze different strategies to select the most informative data samples. Further we devise a simple yet effective strategy to mitigate the lack of diversity in the selected data, caused by the label distribution shift between simulation and real domain [16], [17]. Note, that the latter is important for performance gain in AL under domain shift (simulation vs. reality domain in our case) [18], [19]. Moreover, for the more challenging 2D object detection task, we suggest to incorporate regression uncertainty into the selection process due to its multi-task characteristic (including both classification (*cls*) and regression (*reg*)).

Concretely, we train a BNN with synthetic images, which can be obtained from another task-relevant data set or generated by photo-realistic image synthesizers. In a second step, a

pool of task-specific real images are forwarded to the model. According to the scores from the acquisition function and a sampling strategy, a small subset of samples is selected and solicited for human annotations. The labeled data is then used to adapt the model. The aforementioned procedures can be repeated iteratively until the desired performance is achieved.

Besides the empirical validation of the proposed idea on a classification task, we then conduct evaluation on two more challenging 2D object detection data sets, one with large Sim-to-Real domain shift and another with less to show that the proposed idea can help bridge the gap in a cost-effective way, significantly better than the random baseline and competitive against the state-of-the-art approaches. In addition, we provide a failure case on a third object detection data set to help identify the working scenarios of the proposed idea. To demonstrate the practicality and effectiveness, we further deploy the pipeline on our real robot and show a significantly positive impact of the visual perception within grasping as downstream task.

In summary, the main contributions of this work are:

- we propose to actively and efficiently close the Sim-to-Real gap by applying a BNN in an Active Learning (AL) framework.
- we introduce a simple yet effective sampling strategy to mitigate the label distribution shift in Bayesian AL under domain shift.
- we conduct experiments to empirically show the positive impact of the proposed pipeline on both object classification and 2D object detection tasks, clearly outperforming the random baseline and closely competing against the state-of-the-art approaches
- we demonstrate the applicability in reducing labeling efforts on a real robotic system.

Importantly, the accompanying video provides qualitative results including the demonstration on an assistive robot. The code of the implementation will be publicly available[1].

II. RELATED WORK

a) Sim-to-Real Transfer: Sim-to-Real transfer is mainly tackled with DR and DA. The former treats the real test scenario as one instance of many synthetic ones generated by randomizing the parameters of the synthesizer such as materials, lightening, backgrounds, and plausible geometric configurations [20], [21]. In contrast, DA focuses on learning domain-invariant representations across the different domains (e.g. synthetic and real domain in this context) by sometimes including data of the target domain [4]. Though DA has achieved impressive performance, as mentioned by different researchers, when only relying on unlabeled data, the domain gap is hard to diminish both in theory [9] and in practice [22], [23]. Considering this issue, the paradigm of active learning is appealing to address the reality gap by utilizing annotated real data in an efficient way. In pool-set based active learning [24], the aim is to reach certain level of performance with as less data as possible. In case of supervised

[1]https://github.com/DLR-RM

learning, the data is selected based on their informativeness, which can be measured by different quantities such as the output uncertainty, the disagreement of a committee, or the expected model change [25], [26]. We also stress that active learning is complementary to the aforementioned techniques. While recent works such as [18], [19] argue for the fusion of DA and active learning to obtain better performance, we additionally use DR in this work. Nevertheless, none of them considers employing BNNs for this purpose and most of them focus on classification tasks, which are less relevant for the robots in the real world. Wen *et al.* [27] apply BNNs for DA, but they only focus on conventional passive learning paradigm and classification tasks. We aim to study the active learning paradigm for Sim-to-Real transfer on a more challenging real-world object detection task, which is arguably more relevant for various use-cases of the robots.

b) Active Learning for Object Detection: In the context of active learning for object detection, specific metrics related to characteristics of the underlying network can be applied [28]. While in [29] the margin of the bounding box scores in different layers is used, Kao *et al.* [30] consider the localization tightness and stability. Meanwhile, uncertainty based approaches [19], [25], [31] are also able to achieve competitive performances in the field of object detection. Most of uncertainty based approaches are built on BNNs [14], [32] which can produce more reliable uncertainty estimates. Along with its theoretic soundness, the task-agnostic characteristic of these approaches can facilitate wider applicability for different fields. While some only exploit the classification branch for the uncertainty estimation [33], [34], others [15] consider both classification and regression branches. Yet, they rely on larger amount of annotated real world data to initialize the training of the model and update the model in each iteration, while we assume relatively small amount of real data.

III. PROBLEM FORMULATION AND OVERVIEW

We consider two domains: the simulation domain S and the real domain R. In S, we assume the availability of annotated data set, i.e., given the synthetic data $\mathbf{x}^S$ and annotated labels $\mathbf{y}^S$, we denote the synthetic data set as $\mathscr{D}_S = \{(\mathbf{x}_i^S, \mathbf{y}_i^S)\}_{i=1}^{N_S}$ where N_S is the number of data points. In contrary, R contains an unlabeled data set $\mathscr{D}_T = \{(\mathbf{x}_i^R)\}_{i=1}^{N_R}$ which constitutes of N_R number of real images $\mathbf{x}^R$. We further extend the notations to define an object detection task including classification (*cls*) and regression (*reg*) tasks. Given the space of inputs $\mathscr{X}$ (both synthetic and real images) and outputs $\mathscr{Y}$ (sets of object classes $\mathbf{c}$ and their 2D location as bounding boxes $\mathbf{b}$), we define the object detector as a function $\mathscr{M}_\theta : \mathscr{X} \to \mathscr{Y}$ with parameters θ. Naturally, our objective is to obtain an object detector in the real domain R, for which synthetic data $\mathscr{D}_S$ can be exploited.

To achieve this goal, the proposed pipeline (depicted in Fig. 2) relies on deep Bayesian AL. What motivates our approach is that in practice, this so-called Sim-to-Real transfer can be achieved by combining (a) the large amounts of annotated synthetic data, and (b) a few but the most

(a) synthetic images with annotations

1. train initial detector

object detector

2. scoring

pool of real images

4. adapt detector

3. sampling

annotations from expert
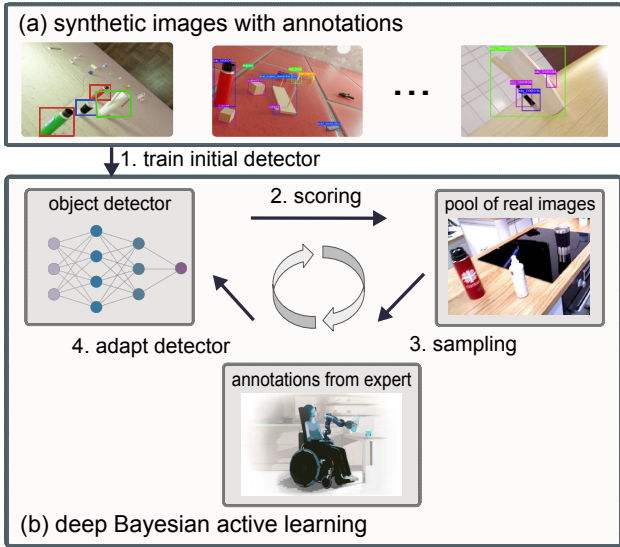
(b) deep Bayesian active learning

Fig. 2. **The proposed Sim-to-Real pipeline.** Using labeled synthetic images, we first train an initial BNN object detector. Then, we rely on deep Bayesian AL to select the most informative images from a pool of unlabeled real images. The scoring of all the images in the pool is obtained via an acquisition function, while sampling is applied to deal with the foreground class imbalance problem. Based on the selected images, the human expert performs the annotation and the detector is adapted via fine-tuning. The process is repeated to close for Sim-to-Real transfer.

informative real images with annotations from human expert. Importantly, we conjecture that the real images can bridge the reality gap in a simple and effective manner, and thus, this work focuses on reducing the amounts of needed real images. For this, as shown in Fig. 2, (i) we use $\mathscr{D}_S$ to train an initial model with domain randomization. (ii) Then, treating the unlabeled real data $\mathscr{D}_T$ as a pool set $\mathscr{D}_{pool}$, we rank the informativeness of each images with an acquisition function $\mathscr{A}(\cdot)$ and then (iii) apply a sampling strategy to create the subset. (iv) The labels of this subset is queried to a human expert for manual annotation. This process can be repeated for multiple times until the reality gap is diminished. Next, we describe and motivate these steps in detail.

IV. THE PROPOSED PIPELINE

This section describes our pipeline of Sim-to-Real transfer for 2D object detection. The main components are a BNN object detector for uncertainty quantification (Sec. IV-A), and deep Bayesian AL framework (Sec. IV-B).

A. Bayesian Neural Networks for Object Detection

We choose to model the object detector $\mathscr{M}_\theta$ as a BNN, in order to obtain its uncertainty estimates. BNNs achieve this by reasoning about the model uncertainty, which indicates *what the model does not know*. Reasoning about the model uncertainty, the AL framework can later leverage this information to label the most uncertainty data to the model itself. To do so, given the training data $\mathscr{D}_{train}$ and a test data sample $\mathbf{x}^*$, BNNs produce the output distribution $p(\mathbf{y}^* \mid \mathbf{x}^*, \mathscr{D}_{train})$ by marginalizing over the models' distribution:

$$p(\mathbf{y}^* \mid \mathbf{x}^*, \mathscr{D}_{train}) = \int p(\mathbf{y}^* \mid \mathbf{x}^*, \theta) p(\theta \mid \mathscr{D}_{train}) d\theta. \quad (1)$$

In (1), $p(\mathbf{y}^* \mid \mathbf{x}^*, \theta)$ is the observation likelihood, and $p(\theta \mid \mathscr{D}_{train}$ is the distribution over the weights θ. As a closed form solution to the integral in (1) does not exist, the Monte-Carlo integration is often used for a numerically solution [35]. As a note, our AL pipeline uses both the synthetic and the annotated real images as the training set $\mathscr{D}_{train}$, and the new images $\mathbf{x}^*$ are samples from the pool set $\mathscr{D}_{pool}$.

However, applying BNNs to the existing anchor-based detectors such as Retinanet [1] requires several adaptations [15], [33]. This is due to their post-processing steps, i.e., (i) *miss-correspondence between the anchor predictions and final outputs*, and (ii) *hard cut-off behavior in non-maximum suppression (NMS) step*. For these, the BayesOD framework [15] can be used, which performs Monte-Carlo sampling for each anchor prediction before NMS steps, and relies on Bayesian inference to infer the output distributions. Intuitively, BayesOD clusters outputs in anchor level using spatial affinity. To explain, assume that such cluster contains M anchors and consider the highest classification score as center of this cluster (indexed by 1). The other outputs are considered as measurements to provide information for the center, denoted by $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{b}}_i$. By further denoting the final predictive distributions for *cls* and *reg* of this cluster as $p_{[\hat{\mathbf{c}}_1,...,\hat{\mathbf{c}}_M]}(\mathbf{c}|\mathbf{x}^*, \mathscr{D}_{train})$ and as $p_{[\hat{\mathbf{b}}_1,...,\hat{\mathbf{b}}_M]}(\mathbf{b}|\mathbf{x}^*, \mathscr{D}_{train})$ respectively, the final output distributions are computed as:

$$p_{[\hat{\mathbf{c}}_1,...,\hat{\mathbf{c}}_M]}(\mathbf{c}|\mathbf{x}^*, \mathscr{D}_{train}) \propto p_{\hat{\mathbf{c}}_1}(\mathbf{c}|\mathbf{x}^*, \mathscr{D}_{train}) \prod_{i=2}^{m} p(\hat{\mathbf{c}}_i|\mathbf{c}, \mathbf{x}^*, \mathscr{D}_{train}),$$

$$p_{[\hat{\mathbf{b}}_1,...,\hat{\mathbf{b}}_M]}(\mathbf{b}|\mathbf{x}^*, \mathscr{D}_{train}) \propto p_{\hat{\mathbf{b}}_1}(\mathbf{b}|\mathbf{x}^*, \mathscr{D}_{train}) \prod_{i=2}^{m} p(\hat{\mathbf{b}}_i|\mathbf{b}, \mathbf{x}^*, \mathscr{D}_{train}).$$

Here, $p_{\hat{\mathbf{c}}_1}(\mathbf{c}|\mathbf{x}^*, \mathscr{D}_{train})$ represents the per-anchor predictive distribution of the cluster center, while $\prod_{i=2}^{m} p(\hat{\mathbf{b}}_i|\mathbf{b}, \mathbf{x}^*, \mathscr{D}_{train})$ is the likelihood of each cluster member given the output. When we choose the Gaussian and Categorical distributions for *cls* and *reg* tasks respectively, the sufficient statistics of them such as mean and covariance matrix can be computed analytically. We refer more details in [15] and next, we discuss the AL framework that relies on the BayesOD framework.

B. Bayesian Active Learning for Sim-to-Real

With the uncertainty estimates of an object detector, the AL pipeline needs to choose the images for annotation. This selection of images is done via an acquisition function. Moreover, due to the domain shift between *S* and *R*, a sampling strategy also needs to be devised to mitigate the bias in the selected data set. We describe below these components and our design choices.

1) Acquisition Function: We define the acquisition function based on the uncertainty estimates from the BNN detector. In this step, the acquisition function is used to obtain the informativeness scores for each detected instance on one image, and then *aggregated* into one final score to represent the informativeness of the entire image. Once the scores are obtained for all the images in the pool set $\mathscr{D}_{pool}$, we sample a subset of them for annotation (IV-B.2) in order to

adapt the model. Specifically, we consider uncertainty from both *category classification* and *bounding box regression*, which are referred to as semantic and spatial uncertainty respectively [36]. For the semantic uncertainty of the *j*-th detection instance on an image, given the Shannon Entropy measure $\mathscr{H}(\cdot)$, the *cls* acquisition function $\mathscr{U}_{j,cls}$ is modeled with a Bernoulli distribution as:

$$
\begin{aligned}
\mathscr{U}_{j,cls} &= \sum_{i=1}^{|\mathscr{C}|} \mathscr{H}(p(c_i|\mathbf{x}^*, \mathscr{D}_{train})), \\
&= \sum_{i=1}^{|\mathscr{C}|} [-p(c_i|\mathbf{x}^*, \mathscr{D}_{train}) \log p(c_i|\mathbf{x}^*, \mathscr{D}_{train}) \\
&\quad - (1 - p(c_i|\mathbf{x}^*, \mathscr{D}_{train})) \log (1 - p(c_i|\mathbf{x}^*, \mathscr{D}_{train}))].
\end{aligned}
\tag{2}
$$

In (2), the steps follows from the definition of the entropy, and optimizing the given measure is equivalent to maximizing the information gain [37] or information content.

The uncertainty from regression is defined as differential entropy of $p(\mathbf{b}|\mathbf{x}^*, \mathscr{D}_{train})$ which is approximated by a multivariate Gaussian with covariance matrix $\mathbf{C}_b$ calculated from the samples of predicted bounding boxes:

$$
\begin{aligned}
\mathscr{U}_{j,reg} &= \mathscr{H}(p(\mathbf{b}|\mathbf{x}^*, \mathscr{D}_{train})) \\
&= \frac{k}{2} + \frac{k}{2} \ln(2\pi) + \frac{1}{2} \ln(|\mathbf{C}_b|),
\end{aligned}
\tag{3}
$$

where k is the dimensionality of random variable $\mathbf{b}$. Again, this regression acquisition function $\mathscr{U}_{j,reg}$ follows from the definition of entropy for Gaussian distributions, and represents the information content of an image.

We choose to exploit these two quantities by a combination function $comb(\cdot)$, in order to produce the uncertainty score for each of N_k detected instance on k-th image. Then, the acquisition function for k-th image $\mathscr{A}$ is defined by aggregating scores with a function $agg(\cdot)$ denoted by:

$$
\mathscr{A}(\mathbf{x}_k) = agg_{j \in N_k}(comb(\mathscr{U}_{j,cls}, \mathscr{U}_{j,reg})),
\tag{4}
$$

The combination function $comb(\cdot)$ can be a weighted sum (*sum*) or maximum (*max*) operation [31]. The aggregation function $agg(\cdot)$ can be a maximum (*max*), summation (*sum*) or average (*avg*) operation [29]. What motivates this is the problem itself, i.e. object detection involves both *cls* and *reg* tasks and multiple instances in one image.
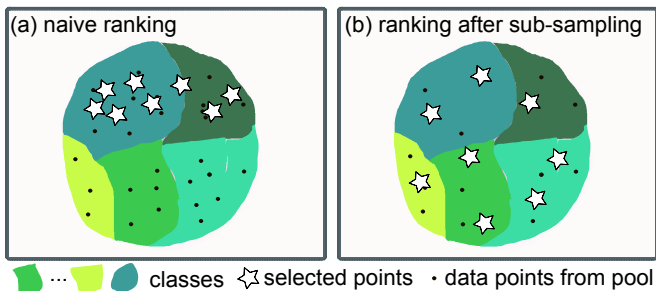

Fig. 3. **Sub-sampling Strategy.** We illustrate the ranking after sub-sampling strategy. A naive ranking selects the most informative points from a few classes of pool data, while the ranking after sub-sampling enables to evenly select the most informative points across the variety of classes. This mitigates the class imbalance problem of AL for object detection, and introduced diversity can improve the performance.

2) Sampling Strategy: One problem in the naive TopN sampling motivates us to combine the TopN sampling with the popular sub-sampling technique [38]. The problem is the violation of the assumption that the simulation domain *S* and the real one *R* are the same, which does not hold in fact. This will further lead to a performance degradation for both AL and object detection training [39], [40]. More specific, to select the *B* most informative images scored based on the model trained on *S* will result in an imbalance problem in the selected data set. Since the algorithm queries only images from real domain *R*, we attribute the under-performance during AL to the label distribution shift [16]. To explain, we denote the distribution followed by sub-sampling as $P_{ss}(\mathbf{c}, \mathbf{b})$ and the distribution followed by uncertainty sampling as $P_{unc}(\mathscr{A}(\mathbf{c}, \mathbf{b}))$, which can be a product of delta distribution with probability mass placed at the top *B* scored predictions. Therefore, the selected data during AL follow the a label distribution $P_{ss}P_{unc}$. Additionally, we use $P_r(\mathbf{c}, \mathbf{b})$ for the real label distribution, which is assumed to be uniform. The goal is to adapt the model with data points drawn from P_r, which is unavailable for unlabeled data. Instead we adapt the model with data points drawn from $P_{ss}P_{unc}$, which ideally should be aligned with P_r. Unlike classification case, in which the label distribution lies in a discrete finite space and importance weighting correction [17] can be easily adapted, the label space for object detection is more complex when there is an additional regression task involved. The trade-off between alleviation of label distribution shift and utilization of information contained in the uncertainty estimates is thus determined by the distribution form of P_{ss} and the amount of data to be sub-sampled. Intuitively, by assuming there is certain degree of redundancy in the data set, we select the uniform distribution for P_{ss}, which works empirically well, shown in the experiments. In practice, the pool set data is filtered by P_{ss} first, and then with P_{unc}, the learner thus can choose by considering the informativeness in the sub-sampled data. An illustrative explanation on the class imbalance problem, one instance of label distribution shift, is shown in Fig. 3.

V. EXPERIMENT

In this section, we first validate the proposed sampling strategy on a classification task, in which the model is transferred from MNIST [41] to MSNIST-M [42]. Then we move on to two *more challenging but task-relevant* self-collected data-sets on *2D object detection*. To note that, we employ two data-sets with different magnitudes of Sim-to-Real gap (one is large and the other small) to demonstrate that the proposed pipeline can efficiently bridge the gap for both cases. In all experiments, *we instantiate the Sim-to-Real gap by subtracting the performance of the corresponding models trained on purely the real and simulated data-set*. Nevertheless, we address the limitation of the proposed idea by including one *failure case* on the public YCBV data set [43] to further identify the operational scenario. In the end, we show the practical effectiveness of our idea by deploying the model on an assistive robot within a grasping
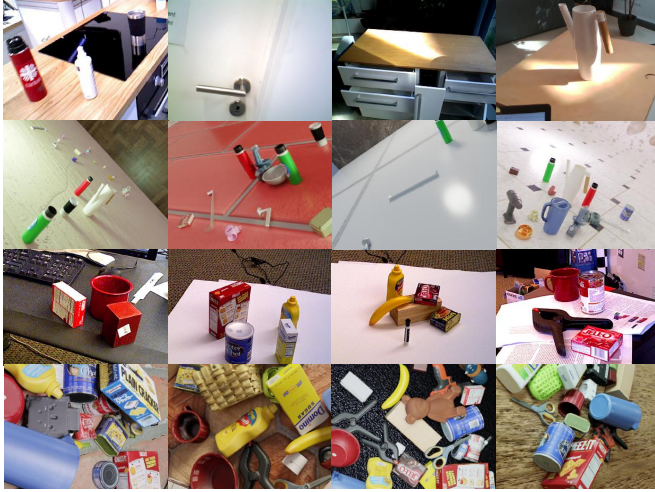
Fig. 4. **The real and synthetic data.** Exemplary images from real (1st,3rd row) and synthetic domain (2nd, 4th row) of EDAN (1-2 rows) and YCBV (3-4 rows) data sets.

task. The implementation details and parameter settings of the proposed pipeline are then provided, which is followed by results and discussions.

a) Data sets: (1) **Digits** include MNIST and MNIST-M digit data sets with *10* classes. MNIST-M contains digits from MNIST but blended with random color patches. We can treat MNIST-M as MNIST digits in real-world in this case and perform Sim-to-Real transfer for them. (2) **EDAN** includes *5* classes: ikea bottle, watering can, door handle, drawer handle and grey mug. With simple textures and geometry of the objects and the *indoors lab environments* (see Fig. 4), the domain gap on this data set is small. (3) **SAM** [44] includes *3* classes: cage, pipe and hook. With more complex textures and geometry of the objects and different weather conditions in *outdoor environments*, the domain gap on this data set is much larger than EDAN. (4) **YCBV** contains images of *21* classes from common objects such as pitcher, sugar box and so on. Basic information of the aforementioned data sets is summarized in Table I and the synthetic data sets except for the one of 1 are generated by BlenderProc [7] with domain randomization applied.

TABLE I

BASIC INFORMATION AND TRAINING HYPER-PARAMETERS ON FOUR DATA SETS

Data set (size of sim, real-pool, real-val, real-test set, number of class)	Query Size (image)	Maximum Training Period during AL (epoch)	Learning Rate	Network Archi-tecture
Digits data set (60k, 55k, 5k, 10k, 10)	20	50	linearly from $1e^{-5}$ to $1e^{-3}$	the same CNN in [26]
EDAN (10k, 0.5k, 0.1k, 1k, 5)	20	10	$1e^{-4}$	RetinaNet [1]
SAM (2.5k, 2k, 0.1k, 0.5k, 3)	80	10	$1e^{-4}$	RetinaNet [1]
YCBV (50k, 1.4k, 0.1k, 0.5k, 21)	50	10	$1e^{-3}$	RetinaNet [1]

TABLE II

COMPARISON OF DIFFERENT ACQUISITION FUNCTIONS: MEAN MAP OVER 10 ITERATIONS FOR DIFFERENT AGGREGATION AND COMBINATION FUNCTIONS WITH AND WITHOUT SUB-SAMPLING STRATEGY AND WITH 10 AND 30 SAMPLES ON EDAN DATA SET.

Agg.	Comb.	10 samples		30 samples	
		w.o. sub.	w. sub.	w.o. sub.	w. sub.
Avg	Max	74.73%	76.51%	74.47%	76.76%
	Sum	74.77%	**77.09%**	75.17%	77.19%
Sum	Max	75.80%	74.54%	76.08%	76.76%
	Sum	71.83%	75.35%	74.31%	76.67%
Max	Max	73.67%	72.67%	73.02%	74.74%
	Sum	75.36%	76.89%	74.98%	**77.49%**

b) Baselines: In order to validate the proposed idea, we compare with the following baselines. (1) **Random**: an approach to randomly select data points for query in each iteration. (2) **Batch-bald** [26]: an approach to query a batch of data with *jointly maximum mutual information* instead of individually. (3) **Clue** [19]: an approach for active domain adaptation that considers both *diversity* and *uncertainty* in the acquisition function. (4) **Coreset** [45]: a *diversity*-oriented approach for AL, whose greedy version is a k-center algorithm. For clue[2] and batch-batch[3], we use the open-sourced implementation and only apply to 1 with *max* aggregation function due to their iterative calculation characteristic. For efficiency within coreset and clue, we use the logits layer as latent features.

c) Implementation details: Training hyper-parameters are summarized in Table I. Within the *sum* combination function, we set the weight of 1 to 1 on all data sets. For regression, we select 0.01 for EDAN and SAM, 0.001 for YCBV. The percentage of sub-sampling is set to 1% for digits data sets and 50% for the others based on the performance on validation set. We set dropout rate to 0.1 in BayesOD and apply Bayesian inference *only for bounding box regression* instead of both heads to avoid under-performance observed in preliminary experiments. We use 100 Monte-Carlo samples to approximate the joint distribution in batch-bald. Regarding the evaluation metric, following the convention in [19], we employ the mean accuracies for classification and mean MAP for object detection over AL iterations.

A. Results and Analysis

a) Design choices: We firstly conduct an initial empirical study on the effects of aggregation and combination functions in Eq. (4), number of samples to approximate Eq. (1) on EDAN data set (Table II). We can observe: 1. More weight posterior samples can lead to slightly better results;

2. The sub-sampling strategy can improve performance most of the cases; 3. When using *avg* and *max* to aggregate uncertainties of detections on the image, the *sum* combination function yields better results; Only within *sum* aggregation function, the *max* operation outperforms.

In general, the setting pairs of *max* + *sum* and *avg* + *sum* provide the best results. As this ablates our design choices,
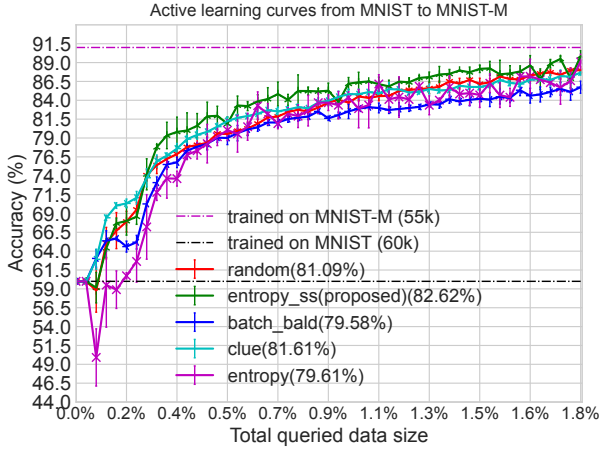
[2]https://github.com/virajprabhu/CLUE
[3]https://github.com/BlackHC/BatchBALD

Fig. 5. **Results on digits data sets (MNIST → MNIST-M).** Active learning learning curves of 3 random runs (with 50 iterations and 20 images queried in each iteration). The black and purple dotted lines represent the performance (on MNIST-M test set) of model trained on MNIST and MNIST-M training set with size in the parentheses, respectively. The compared methods include the proposed one (*entropy_ss*) and other baselines. Values in the parentheses are mean accuracies over 50 iterations.

TABLE III

RESULTS SUMMARY FOR OBJECT DETECTION DATA SETS. VALUES IN THIS TABLE ARE 1. **PERCENTAGE OF ANNOTATED IMAGES REQUIRED TO BRIDGE SIM-TO-REAL GAP** (LOWER THE BETTER) AND 2. **MEAN mAP OVER 10 ITERATIONS WITHIN AL** (HIGHER THE BETTER).

	Random	Proposed	Coreset	Clue	Batch-bald
EDAN	> 40% / 75.7%	**36%** / **77.1%**	> 40% / 75.0%	> 40% / 75.7%	> 40% / 72.9%
SAM	> 40% / 81.4%	32% / 82.2%	**20%** / **85.6%**	32% / 85.0%	> 40% / 82.0%
YCBV	**40%** / **65.2%**	> 40% / 63.5%	>40% / 61.1%	**40%** / **65.2%**	> 40% / 64.8%

we use this insight and mainly focus on these two settings with 10 samples and report only the one with better results.

b) Results on digits data sets: In Fig. 5, we can see that the domain gap can be bridged with ∼ 2% data by the proposed sub-sampling strategy (*entropy_ss*), faster and better than the *random* and *clue* baseline. In contrast, the naive *entropy* and *batch_bald* perform worse than *random* along with large variations. This shows that the proposed sampling strategy for mitigating distribution shift in AL is able to provide greater performance gain than the one considering trade-off between *uncertainty* and *diveristy*.

c) Results on EDAN data set: In Fig. 6 and Table III, we can learn that the gap can be eliminated by the proposed method with *avg* to aggregate detections and *sum* to combine *cls* and *reg* uncertainties(*avg_sum_ss*) with only 36% data, outperforming both the strong baseline *random* and *clue*. In contrary, while *clue* is on a pair with *random* and slightly better than *coreset*, *batch_bald* has the lowest mean mAP. This demonstrates that utilization of information from both *cls* and *reg* with sub-sampling is advantageous in the case of data set with moderate distribution shift like EDAN.

d) Results on SAM data set: The final detector on this data set can achieve a quite decent mAP (> 90%),
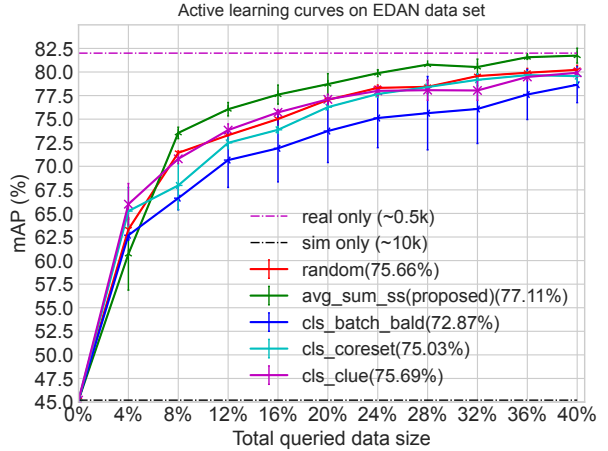

Fig. 6. **Results on the EDAN data set.** Active learning curves of 3 random runs (with 10 iterations and 20 images queried in each iteration). The black and purple dotted lines represent the performance of model trained on sim and real data sets with size in the parentheses, respectively. The compared methods include the proposed one (*avg_sum_ss*) and other baselines. Values in the parentheses are mean mAP over 10 iterations.
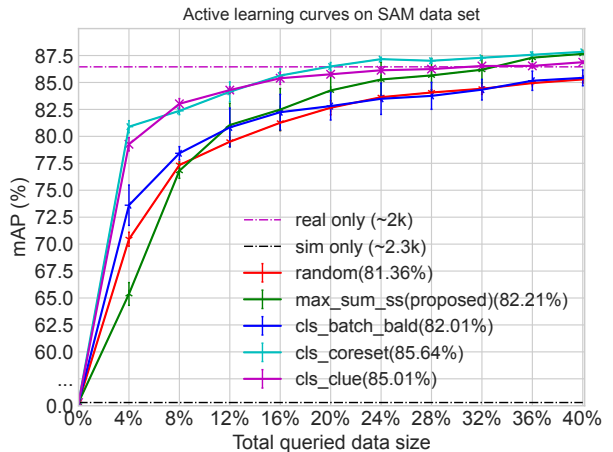

Fig. 7. **Results on the SAM data set.** Active learning curves of 3 random runs (with 10 iterations and 80 images queried in each iteration). The black and purple dotted lines represent the performance of model trained on sim and real data sets with size in the parentheses, respectively. The compared methods include the proposed one (*max_sum_ss*) and other baselines. Values in the parentheses are mean mAP over 10 iterations.

therefore in this experiment we aim to bridge the gap up to a sufficient level, which is *95% of gap*. In Fig. 7 and and Table III, the proposed method with *max* as aggregation and *sum* as combination function followed by sub-sampling strategy (*max_sum_ss*) is still able to beat the strong *random* baseline as well as *batch_bald* and diminish the gap. Nevertheless, *clue* and *coreset* perform better than *max_sum_ss* probably due to the larger domain gap on this data set. It could attribute to the reason that our proposed sampling strategy aims to compensate the shift in label distribution, thereby less effective for a large shift in the input distribution.

e) Limitation Analysis: In this sub-section, we show a failure case on YCBV data set to demonstrate the limitation of the proposed idea. With this, we aim to identify the

operational scenarios of AL for Sim-to-Real transfer and highlight the characteristic of this problem with the hope of providing some enlightening thoughts for the community.

In the last row of Table III, we see that all approaches are on a par with (*clue*) or worse than (*avg_sum_ss*, *coreset*, *batch_bald*) the *random* baseline. To investigate the reason behind, inspired by [39], we compute the average inter class variations over AL iterations in Table IV. The inter class variation is defined as the $\sigma \times C$, where C is the number of class and σ is the standard deviation of number of instances for all classes. The lower this value is, less variations and more balance the object category distribution possesses. We can quantitatively observe that variations of YCBV are significantly larger than the others due to greater number of class, which might pose greater difficulty on decreasing the label distribution shift. Further from row-wise comparison, there is an obvious inversely proportional relation between inter class variations and the performance on YCBV, which is obscure on EDAN and SAM. Therefore, we infer that the impact of label distribution shift is more severe on data sets with greater number of class, thus impeding the effective utilization of uncertainty estimates. Considering this, we suggest that it is more effective to employ the proposed pipeline for bridging the reality gap when the class imbalance problem, one instance of label distribution shift for object detection problem, is at a small scale.

TABLE IV
INTER CLASS VARIATIONS FOR THE SELECTED DATA SET IN EACH
ITERATION DURING ACTIVE LEARNING. LOWER THE BETTER.

	Random	Sub-sampling	Core-set	Clue	Batch-bald
EDAN	90	152	**78**	114	126
SAM	81.3	39.9	71.8	75.3	**12.6**
YCBV	**268**	316	305	**268**	318

B. Deployment on EDAN

On account of the working scenarios (e.g. care-giving) for an assistive robot [11], a variety of objects need to be detected and the manual efforts required for adaptation must be kept as minimum as possible. Therefore, we show the effectiveness of the proposed idea in a shared-control grasping task on EDAN (Fig. 8), where a user such as people with motor disability sitting on the chair intends to control the robot arm for tasks like pouring by using an input device (EMG signal sensors or a spacemouse (used in demo)) with lower degrees of freedom (DoFs) than that of the end effector (3 vs. 6). The mis-correspondence of DoFs between the input device and the manipulator demands that the user needs to tediously switch input mapping between them for task completion in a pure manual control mode. In order to ease task execution, we employ shared-control templates [46], which require robust and precise 2D object detection and pose estimation. For more details on how to incorporate the perception pipeline into the shared-control module, we refer readers to the original work [46].
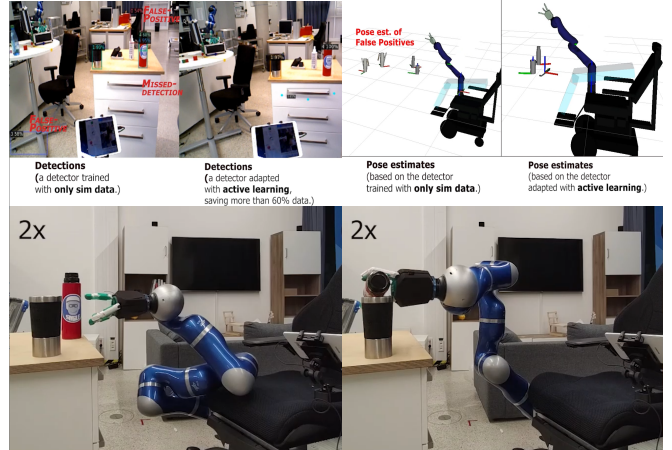

Fig. 8. **Exemplary screenshots of a pouring task via shared control on an assistive robot.** The two screenshots on the top show the performance of the detector and the corresponding pose estimates (visualized in Rviz) before (left in each column) and after (right in each column) adaptation via the proposed pipeline. The two screenshots at the bottom show the sequence of a grasping and pouring task execution with shared-control [11].

In this demo, we integrate the adapted detector trained with a similar setting introduced in the previous section and further use the Agumented autoencoder (AAE) and Iterative Closest Point (ICP) pipeline [3] for accurate pose estimation. This perception pipeline can be deployed on an embedding system such as a NVIDIA Jetson TX2 or a workstation PC (used in demo), the predictions (i.e. pose estimates of the detected objects) are then sent to the shared-control module via Links and Nodes (LN) middle-ware. Based on this, the user is able to control the manipulator to perform a series of common daily tasks such as pouring and drinking with much less cognitive workload. We also provide a video to showcase the deployment.

VI. CONCLUSION

This paper presents an active Sim-to-Real pipeline for 2D object detection, in which, a model is initially learned from synthetic data. Having observed the sub-optimal performance of learning only from simulation, we propose to efficiently use real annotated data via exploiting deep Bayesian active learning. Empirically, we demonstrate the encouraging impact of the proposed pipeline on classification and 2D object detection data sets, further address the limitation of the proposed pipeline and show its applicability on a real robotic system. In particular, our experiments indicate that the proposed sampling strategy can alleviate the label distribution shift which can have a vital impact on the success of our pipeline. More importantly, our work provides an empirical evidence that the real annotated images can efficiently reduce the reality gap.

VII. ACKNOWLEDGMENTS

for Data Science (MUDS) and Rudolph Triebel is a member of MUDS.

References

[1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Int. Conf. on Computer Vision*, 2017.

[2] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z.-C. Márton, and R. Triebel, "Unknown Object Segmentation from Stereo Images," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[3] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *Int. Journal of Computer Vision*, vol. 128, no. 3, 2020.

[4] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *ICRA*, 2018.

[5] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *arXiv:1702.07836*, 2017.

[6] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," *IROS*, 2017.

[7] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "Blenderproc," *arXiv:1911.01911*, 2019.

[8] M. G. Müller, M. Durner, A. Gawel, W. Stürzl, R. Triebel, and R. Siegwart, "A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments," in *IROS*, 2021.

[9] A. K. Tanwani, "Domain invariant representation learning for sim-to-real transfer," in *CoRL*, 2020.

[10] M. Ranaweera and Q. H. Mahmoud, "Virtual to real-world transfer learning: A systematic review," *Electronics*, vol. 10, no. 12, 2021.

[11] J. Vogel, A. Hagengruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Höppner, D. Leidner, and A. Albu-Schäffer, "Edan: An emg-controlled daily assistant to help people with physical disabilities," in *IROS*, 2020.

[12] M. Durner, S. Kriegel, S. Riedel, M. Brucker, Z. Márton, F. Bálint-Benczédi, and R. Triebel, "Experience-based optimization of robotic perception," in *Int. Conf. on Advanced Robotics (ICAR)*, 2017.

[13] J. Feng, M. Durner, Z.-C. Marton, F. Balint-Benczedi, and R. Triebel, "Introspective robot perception using smoothed predictions from bayesian neural networks," *Robotics Research. ISRR 2019.*

[14] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. on Machine Learning (ICML)*, 2016.

[15] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[16] A. Prabhu, C. Dognin, and M. Singh, "Sampling bias in deep active classification: An empirical study," *arXiv:1909.09389*, 2019.

[17] E. Zhao, A. Liu, A. Anandkumar, and Y. Yue, "Active learning under label shift," in *Int. Conf. on Artificial Intelligence and Statistics*, 2021.

[18] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, "Active adversarial domain adaptation," in *Winter Conf. on Applications of Computer Vision (WACV)*, 2020.

[19] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *CVPR*, 2021.

[20] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *Europ. Conf. on Computer Vision (ECCV) Workshops*, 2018.

[21] T. Hodan, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter, "Photorealistic Image Synthesis for Object Instance Detection," *arXiv*, 2019.

[22] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, "Adapting object detectors via selective cross-domain alignment," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *CVPR*, 2018.

[24] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, 1996.

[25] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, "Deep active learning for efficient training of a lidar 3d object detector," in *Intelligent Vehicles Symposium (IV)*, 2019.

[26] A. Kirsch, J. Van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, "Bayesian uncertainty matching for unsupervised domain adaptation," *arXiv:1906.09693*, 2019.

[28] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. López, "Active learning for deep detection neural networks," in *Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 3672–3680.

[29] S. Roy, A. Unmesh, and V. P. Namboodiri, "Deep active learning for object detection." in *British Machine Vision Conf. (BMCV)*, vol. 362, 2018.

[30] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu, "Localization-aware active learning for object detection," in *ACCV*, 2018.

[31] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, "Active learning for deep object detection via probabilistic modeling," *ICCV*, 2021.

[32] J. Lee, M. Humt, J. Feng, and R. Triebel, "Estimating model uncertainty of neural networks in sparse information form," in *ICML*, 2020.

[33] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *ICRA*, 2018.

[34] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel, "Trust your robots! predictive uncertainty estimation of neural networks with sparse gaussian processes," in *5th Annual Conference on Robot Learning (CoRL)*, 2021.

[35] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, *et al.*, "A survey of uncertainty in deep neural networks," *arXiv:2107.03342*, 2021.

[36] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, "Probabilistic object detection: Definition and evaluation," in *WACV*, 2020.

[37] D. J. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, 1992.

[38] Y. Yang, G. Ma, *et al.*, "Ensemble-based active learning for class imbalance problem," *Journal of Biomedical Science and Engineering*, vol. 3, no. 10, 2010.

[39] U. Aggarwal, A. Popescu, and C. Hudelot, "Active learning for imbalanced datasets," in *WACV*, 2020.

[40] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *Trans. on Pattern Analysis and Machine Intelligence*, 2020.

[41] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[42] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, 2016.

[43] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *Robotics: Science and Systems (RSS)*, 2018.

[44] J. Lee, R. Balachandran, Y. S. Sarkisov, M. De Stefano, A. Coelho, K. Shinde, M. J. Kim, R. Triebel, and K. Kondak, "Visual-inertial telepresence for aerial manipulation," in *ICRA*, 2018.

[45] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *Int. Conf. on Learning Representations (ICLR)*, 2018.

[46] G. Quere, A. Hagengruber, M. S. Z. Iskandar, S. Bustamante Gomez, D. Leidner, F. Stulp, and J. Vogel, "Shared control templates for assistive robotics," in *ICRA*, 2020.