Department of Informatics
Technical University of Munich

Master's Thesis in Robotics, Cognition, Intelligence

# Multi-Task Active Learning for Autonomous Driving

## Aktives Lernen für Multi-Task-Modelle im Kontext des Autonomen Fahrens

**Supervisor** Prof. Dr.-Ing. habil. Alois C. Knoll

**Advisor** Walter Zimmer, M.Sc. & Aral Hekimoglu, M.Sc.

**Author** Philipp Friedrich, B.Sc.

**Date** May 15, 2021 in Garching

# Disclaimer

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Garching, May 15, 2021

(Philipp Friedrich, B.Sc.)

## Abstract

In many of the tasks that are required for autonomous driving, machine learning is more and more the method of choice. To ensure that the results of machine learning are reliable and safe, a huge amount of data is needed. While the collection of such data is somehow easy, its labelling is highly tedious and costly. Especially in a task like semantic segmentation, where each pixel must be labelled, a single image can take up to several minutes to annotate. To avoid the annotation of images that are not useful for the training of a network, active learning can be used. Iteratively, a subset of the unlabeled data pool is selected based on the models' uncertainty. This subset is annotated and used to continue the training. This procedure is repeated until the labelling budget is exhausted or no data is left. Using active learning, the overall amount of needed data, time and cost can be reduced. In this thesis, the research gap of active learning for multi-task models in the context of autonomous driving is filled. Several active learning methods for 2D object detection and semantic segmentation are evaluated and compared. In addition, a novel method combining the knowledge about both task domains is presented. This method alone can leverage the accuracy of both tasks while keeping the annotation costs lower than random selection. Furthermore, a novel approach to combining multiple active learning methods is introduced. With this, the accuracy could be improved even more.

## Zusammenfassung

In vielen Bereichen des autonomen Fahrens ist das maschinelle Lernen die Methode der Wahl. Um sicherzustellen, dass die damit erzielten Vorhersagen zuverlässig und sicher sind, wird eine große Menge an Daten benötigt. Während die Sammlung solcher Daten einigermaßen einfach ist, ist ihre Annotation sehr zeit- und kostenintensiv, insbesondere bei einer Aufgabe wie der semantischen Segmentierung. Aktives Lernen wird eingesetzt um die Annotation von für das Training eines Deep Neural Network (DNN)s ungünstigen Bildern zu vermeiden. Dabei wird iterativ die Unsicherheit des Modell genutzt, um eine Teilmenge des nicht annotierten Datensatzes auszuwählen. Diese Teilmenge wird von einem Experten annotiert und für das weitere Training verwendet. Dies wird so lange wiederholt, bis das Kosten- oder Datenbudget für die Annotation erschöpft ist. Mit Hilfe des aktiven Lernens kann die Gesamtdatenmenge, wie auch der Zeit- und Kostenaufwand reduziert werden. In dieser Arbeit wird das aktive Lernen für Multi-Task-Modelle im Kontext des autonomen Fahrens untersucht. Es werden verschiedene aktive Lernmethoden für 2D-Objekterkennung und semantische Segmentierung evaluiert und verglichen. Darüber hinaus wird eine neue Methode vorgestellt, die das Wissen beider Aufgabendomänen nutzt. Außerdem wird ein neuartiger Ansatz zur Kombination mehrerer aktiver Lernmethoden vorgestellt. Damit konnte die Genauigkeit verbessert werden, während die Annotationskosten niedrig gehalten wurden.

# Contents

# Chapter 1

# Introduction

## 1.1 Background & Motivation

The localization and recognition of objects such as pedestrians and other road users are of enormous importance for autonomous operating vehicles. Without the knowledge about such dynamic objects, but also about static objects like traffic signs and lights, the planning and execution of a safe and comfortable drive is not feasible. This challenge is tackled more and more often by using machine learning approaches. Recent state-of-the-art methods have already achieved great results in both 2D and 3D object detection. Contextual information about not only objects but also areas like the derivable surface, sidewalks and vegetation can be gained using pixel-wise semantic segmentation. The high accuracy of these two tasks often comes at high computational costs, which often causes the approach to not meet the real-time requirement of autonomous driving. This first problem can be solved by using a multi-task architecture, which shares some of the model weights and thus reduces the overall computational effort. Recent publications showed, that a model that predicts both 2D object detections and pixel-wise semantic segmentation can reach a higher accuracy on both tasks compared to the single-task trained models [Dvo+17]. Another problem of machine learning in the context of autonomous driving is the tremendous need for data. To be safe and reliable in all possible scenarios that could happen while participating in the traffic, a large spectrum of various traffic scenes must be captured, labelled and learned. While the capturing and learning of data is somehow easy to accomplish, labelling requires a human expert and the task itself is tedious. The process is even for experienced annotators time-intensive and therefore very costly, especially for the task of semantic segmentation. Recent studies showed that various active learning methods, such as uncertainty estimation, can help to improve the accuracy while keeping the annotation costs to a minimum. In active learning, a small amount of the collected data is labelled and used for training. After the first training iteration, the networks' predictions on the remaining unlabeled data are used to determine, which images are most useful for the further training process and should be annotated. This procedure is repeated until the labelling budget is reached or no data is left. Using active learning, the overall amount of needed data, training and annotation time as well as annotation costs can be reduced. At the time of writing this thesis a lot of ongoing research on active learning for object detection and semantic segmentation is done. But to the best of my knowledge, no research has been done on a multi-task model focusing on these two tasks. This research gap will be filled in the following.

## 1.2 Foundations

This thesis has three major components that need to be understood first. These are the two tasks that are investigated, namely 2D object detection and pixel-wise semantic segmentation. In this work, various methodologies are investigated to improve the performance on these two tasks while keeping the required data annotation costs low. To achieve the optimal trade-off between high accuracy and low costs, the third major component is applied. Using uncertainty estimation methods, the pool of unlabelled samples can be ranked by their potential to improve the overall results. While object detection and semantic segmentation were heavily researched in recent years and thus its basic mechanisms should be known, the third mentioned component, namely uncertainty estimation is less known and therefore described in more detail in the following. The predictions of machine learning-based models aren't and probably never will be completely reliable. They always contain a certain amount of uncertainty, the causes of which can be divided into two categories. One is called aleatoric uncertainty, which is caused by the noise contained within the sensor data or the general inaccuracy of the sensor themselves. One prime example of aleatoric uncertainty is the data produced by an RGB camera during night [Fen+21]. Methods that tackle to remedy this uncertainty are direct modelling and error propagation. In direct modelling, a probability distribution over the model outputs is assumed and the output layers are used to predict the parameters for this distribution. This method is efficient, as it requires only one forward pass. But on the other hand, it requires a modification of the networks' output layers, as well as the loss function [Fen+21]. In error propagation, the uncertainty is approximated in each activation layer and as the name suggests, propagated through the whole model. This approach is computational efficient at inference and requires only limited modifications of the architecture [Fen+21]. The second category of predictive uncertainty is called epistemic uncertainty. It indicates the certainty of a model about its description of a dataset given its learned parameters. The detection of an unknown object which is not in the training data, for example, results in a high epistemic uncertainty [Fen+21]. This type of predictive uncertainty is investigated in this thesis. Existing methods are e.g. Monte-Carlo Dropout or Deep Ensembles. In both approaches, multiple predictions of the same sample are generated and compared with each other. In the Monte-Carlo Dropout method, dropout layers are used. An enabled dropout layer results in deviating predictions on the same input when the inference is performed multiple times. In deep ensembles, multiple instances of the same architecture of a model are trained with randomly shuffled training data using a different parameter initialization. This way, each instance of the ensemble models generates a slight variation of predictions, which then can be used to approximate the predictive probability. Both Monte-Carlo Dropout and Deep Ensembles are computational inefficient as it either requires multiple inference passes or a linearly scaling number of models [Fen+21]. More methods, that have a better efficiency are described in Section 2.1.

## 1.3 Contribution & Outline

This thesis investigates existing methods of active learning for 2D object detection and pixel-wise semantic segmentation in the context of autonomous driving. Furthermore, an existing active learning methodology for 2D object detection is adapted to the semantic segmentation task, forming the novel $InconSeg$ approach. In addition to that, a novel approach called $BoxMask$ is proposed which combines the knowledge from the object detection domain with the knowledge from the segmentation domain. Besides the extensive study of the new and existing sample selection strategies, the alternation of multiple strategies during the training is

investigated. Using a combination of the novel sample selection approach and the alternating training strategy, the existing methods could be outperformed with respect to the accuracy of 2D object detection and semantic segmentation, while requiring fewer annotations compared to traditional training approaches.

The thesis is structured as follows. In Section 2 recent publications on active- and multi-task learning are described in detail. Section 3 will then describe the technical details of the baseline methods, as well as the newly introduced sample selection strategies. The experiment setup, the used datasets and the evaluation metrics are presented in Section 4. Both the quantitative and the qualitative evaluation results are given in Section 5. The discussion of the results and the conclusions that can be drawn from them can be found in Section 6. Finally, the future work is given in Section 7.

# Chapter 2

# Related work

In this chapter, some of the recent works which have relevance to this thesis are described. The works are split into two main sections, Section 2.1 focuses on the work on active learning in recent years, and Section 2.2 is mentioning the latest works on multi-task learning architectures.

## 2.1 Active Learning

In general active learning can be categorized into three approaches to handle the unlabeled sample query. In the first category are query-synthesizing methods, where the active learning agent creates its data samples based on known data. This could be done by augmentation, such as brightness reduction to simulate images taken at night or zooming into the picture to change the perspective of an object. In [MT18] a Generative Adversarial Network (GAN) is used to generate high entropy samples. Their proposed technique searches similar samples from the unlabeled pool. The authors of [Mah+18] use a GAN to generate realistic chest x-ray images with different disease characteristics. This approach is in particular useful if the dataset is small. Another sampling approach is stream-based selective sampling. Here the active learning agent decides, during the training, for each data sample one by one if the agent is confident enough to use the model predictions, or if the oracle should be asked for the actual ground truth. The oracle is often used as an alias for the human expert that annotated the data. The disadvantage of this approach is that if you have a fixed labelling budget of $N$. In the worst case, the budget is exhausted and you selected the first $N$ samples instead of the best $N$ samples of the whole unlabeled dataset. Query-acquiring or pool-based methods solve this disadvantage at the cost of the more required time. All samples of the whole unlabeled dataset are ranked according to a specific acquisition function. The highest-ranked samples are used in the next training cycle. There exist various categories of strategies to determine how to rank the samples. The most common ones are uncertainty-based sampling, diversity-based sampling, and their combination. The underlying assumption of the uncertainty-based sampling methods is that samples that the network is uncertain about, are hard to learn. Therefore, it makes sense to present those objects during the training cycles. However, this often has the drawback that the selected samples do not represent the same distribution as the real-world data, which can cause a poor model performance if applied to real-world tasks. In some cases, the developed model might be confident but wrong due to the non-representative data selection. This is where diversity-based sampling methods come into play. Instead of targeting to identify known unknowns of a model, the goal of diversity-based sampling methods is to spot the unknown unknowns of the model. This is most commonly achieved by clustering the training data and identifying model-based outliers. For many use cases, a

combination of uncertainty and diversity sampling is feasible and effective. Depending on the task at hand, different sampling strategies are more reasonable than others. A broad survey of the different active learning fundamentals is given in [Set10]. A more recent guide to active learning techniques is given in [Mon21].

In the following subsections, the most relevant publications related to active learning in the context of autonomous driving will be described in more detail, grouped by their intended application task.

### 2.1.1 Image Classification

Early active learning methods primarily focus on the task of classifying an image. [Bel+18] compares Ensemble-based methods against Monte-Carlo Dropout (MC Dropout) methods. The idea of Ensembles goes back to [HS90]. Multiple instances of the same model architecture are trained on the same set of data, but all have a different initialization, a randomly shuffled data arrangement during training as well as different hyperparameters, and thus predict different probabilities for each unlabeled image. These various predictions can be used to measure the uncertainty of the model for that specific sample image. Besides using Ensembles, another commonly used method for obtaining uncertainty estimations is MC Dropout [GG15]. By running the inference multiple times with an enabled dropout layer, one can get multiple probabilities for the same sample. This approach is, compared to Ensembles, less time and resource consuming as only one model must be trained.

Beluch et al. [Bel+18] apply and evaluate various acquisition functions to rank the samples for the labelling step. One of them is Entropy, which has been introduced in [Sha48]. It can be intuitively understood as the likeliness of one class given a set of classes. If the entropy is low, one single class is very likely, while the others are more unlikely. If the entropy value is high, all classes are equally likely to be correct. Another method used is called Mutual Information between data points and weights, which is also known as BALD and was introduced by Houlsby et al. [Hou+11]. The underlying assumption of this strategy is, that data samples that have high mutual information between their prediction and the network weights will most likely have a high impact on the model training, given that the correct label is available. Another frequently used acquisition strategy utilizes the Variation Ratio [Fre65] which measures the statistical dispersion in nominal distributions. The authors of [Bel+18] not only evaluated uncertainty-based sampling strategies, but also a diversity-based one and a combination of uncertainty and diversity sampling. The former one is a core-set approach, which has been proposed in [SS18]. It targets to minimize the maximum distance between a data point in the overall distribution and its closest neighbour data point in the selected subset. The latter one has been proposed in [Yan+17]. It is a two-step procedure that takes both uncertainty and diversity into account. In the first step, the samples with the highest uncertainty are extracted from the data pool. In a second step, a representative score is assigned to each sample, by calculating the similarity of a sample and the most similar sample in the selected subset. The authors conclude that ensemble-based uncertainties outperform other methods of uncertainty estimation, such as MC Dropout and justify that finding with the decreased model capacity and lower diversity of MC Dropout.

[Ben+21] combines self-supervised learning with active learning. They claim that self-supervised is more effective at reducing labelling efforts if the labelling budget is low. If the labelling budget is high, a combination of both methods outperforms both single methods as well as random sample selection.

### 2.1.2 Object Detection

One of the first papers that focus on active learning, specifically for object detection, is [Kao+19]. They propose two informativeness metrics, namely localization tightness and localization stability. The tightness can be calculated by measuring the change between the region proposal box and the actual predicted box. For stability, noise is added to the image and the change in the detected regions is measured. If there is no change the model is assumed to be already well trained. Their results show that a combination of localization tightness and class uncertainty has the highest performance in most of the cases. They furthermore show that localization stability combined with class uncertainty adds the most improvement for difficult categories.

In [RUN18] a novel active learning method is developed called query-by-committee. They claim that black-box models are outperformed by white-box models. Black-box models are models that only use metrics like confidence and do not use any knowledge about the underlying network architecture, white-box-models on the other hand, use the underlying network architecture as an additional information source. The proposed committee of classifiers is formed by the last layer of the base network along with the extra convolution layers. To query the images, the disagreement between those layers is aggregated for each candidate bounding box in an image. For each bounding box $b$, the neighbouring bounding boxes generated by the other convolution layers are used to compute the margin of the box $b$. The difference between the confidence scores of $b$ and the most confident auxiliary bounding box builds this margin. The higher the margin, the higher the disagreement between the convolutional layers.

A more straightforward method is proposed in [BKD19]. The authors claim that 1-vs-2-margin sampling produces better results than least confidence sampling. In 1-vs-2 sampling the two highest-scoring classes are taken into account for the uncertainty estimation, while for least confidence only the highest scoring class is used. Brust et al. evaluate various methods to use the 1-vs-2 sampling strategy. They take either the sum, the average or the maximum of the 1-vs-2 scores of all detections to rank the images that should be given to the oracle. In addition to the aggregation methods, they also introduce a selection imbalance method. With that, instances, where the predicted class is underrepresented in the training set, are preferred. Their analysis states the sum as the best performing method to aggregate the uncertainty scores. But it is worth mentioning that this is most likely due to the fact, that the sum tends to select samples containing many single objects, which increases the annotation effort.

In [Agh+19] each pixel of each image gets a detection probability assigned. Considering the spatial neighbourhood a per-pixel score is generated, indicating how informative each pixel may be for improving the network. The pixel-level scores are aggregated to an image-level score. The image is divided into non-overlapping regions and the maximum score of each region is calculated. The overall image score is the average of the max-pooled region scores. The authors claim that max-pooling is the best performing aggregation method and that their method outperforms random selection, given that the object detector has enough capacity for the complexity of the targeted domain.

[Sch+20] compares Ensembles with MC Dropout and claims that Ensembles achieve better results. They furthermore propose several new uncertainty scores, not only for 2D but also for 3D object detection. The authors incorporate the Region of Interests (RoI) and propose novel scores, namely the consensus score, consensus score with variation ratio, RoI matching and sequence RoI matching. Their experiments make clear, that the consensus score alone does not outperform random selection, but combined with the variation ratio it performs slightly better. The proposed methods using RoI matching perform best. They also show that

continuous training is not only better performing than training the network from scratch in each training cycle, but it is also more time and data-efficient.

The authors of [Hau+20] evaluate how well various scoring functions work on unlabeled data at scale. They evaluate Entropy, Mutual Information, uncertainty estimation based on the gradient of the output layer and lastly using the confidence of each bounding box. The latter is performing best but favours samples with many bounding boxes, which increases the annotation costs. The best trade-off provides Mutual Information having a slightly smaller Weighted Mean Average Precision (wMAP) and fewer bounding boxes per image on average.

All approaches so far somehow compute a score on an image-level, [DB20] takes a different approach. Instead of querying the images, they rank the bounding boxes based on their informativeness. They then give a certain amount of bounding boxes to the oracle for the correction of the predicted labels. The images containing the top bounding boxes are used in the next learning cycle. The bounding boxes of the images in the training set that haven't been labelled by the oracle yet, are pseudo labelled with the network's prediction and act as noisy labels. The authors use random selection and the proposed metrics from [RUN18; SS18] as the image-level querying method and use random selection, class uncertainty, and core-set strategy as the box-level querying method. Their experiments show that box-level querying methods consistently outperform the image-level ones.

Another simple, yet efficient approach is proposed in [Ele+21b]. For each unlabeled image the, bounding boxes of both the original image as well as the horizontally flipped image are predicted. The predictions are then matched by their Intersection over Union (IoU) and for each matched pair an inconsistency score is calculated using the Kullback–Leibler-divergence (KL-divergence). Samples with the highest inconsistency score are selected for further labelling. The resulting acquisition function favours the most informative and therefore often hard samples. Ignoring confident samples can cause a distribution drift. To remedy these issues, the authors use very confident predictions as pseudo-labels which increases the performance while keeping the annotation effort low. The proposed approach is compared against the methods of [SS18; YK19; Aga+20; Cho+21; Yua+21] as well as against the multi-model approaches of [GIG17; Bel+18]. The conducted experiments show that entropy-based active learning tends to perform better for the best-performing classes, while the proposed inconsistency-based method is more robust in general and thus better suited for low-performing classes. The inconsistency-based acquisition function outperforms all other methods evaluated in the publication in most of the classes. In a detailed ablation study, the authors make clear that pseudo-labels help to make the network more robust and prevent a dataset drift.

In [Li+21] six sampling functions are compared against each other. These are random sampling, least confidence [RUN18], entropy [Agh+19], 1-vs-2 margin [BKD19] and two new proposed methods. Li et al. claim that the least confidence strategy has limitations for object detection that are tackled with the first newly proposed method that is based on the expected error reduction, called Bet Gradient (Bet Gradient (BetG)). It is a promotion of least confidence for object detection. The second one is called weighted Bet Gradient (Weighted Bet Gradient (WBetG)) which is a weighted improvement of BetG. Their experiments show that WBetG in combination with diversity sampling (Weighted Bet Gradient with diversity Sampling (WBetGS)) has the best overall performance.

Most of the methods that have been published in recent years focus primarily on the classification uncertainty of a bounding box and ignore its localization uncertainty. The authors of [Cho+21] propose a method that is based on mixture density networks that learn a Gaussian Mixture Model for both localization and classification. This enables the network to directly predict a probability distribution and therefore the computation of both aleatoric and epistemic uncertainties. The reported results are on par with Ensembles or MC Dropout,

but this method requires way fewer computations and time.

In [Ele+21a] the authors point out the importance of a balanced data selection. If only hard samples are selected for the labelling, a distribution shift could be the case and the model might not learn a good representation of the dataset. During the acquisition step, each sample gets a combined score based on three metrics. If the confidence reaches a specific threshold, the entropy is multiplied by the inconsistency score. This way, only moderately-difficult objects are selected. They use augmentations, e.g. horizontal flipping, and compare the two predictions while ignoring their correctness to compute the consistency. To prevent the waste of labelling resources for samples that are easy to sample, a pseudo labelling module is proposed. The objects where the network is very confident are labelled automatically with the network's prediction and used for the training. As some of the samples might contain objects that weren't pseudo labelled, the training loss is adapted accordingly. The proposed method is compared against random and entropy sampling, core-set by [SS18] as well as against the loss module by [YK19]. Furthermore, they add multi-model methods like an Ensemble-based [Bel+18] and a MC Dropout-based [GIG17] method to the comparison. It is shown that multi-model approaches outperform single models at the cost of increased training time and resources. The proposed method, however, outperforms the other methods and even the multi-model ones. The authors emphasize that Non-Maximum Suppression (NMS) is very important in all active learning methods, which is confirmed in an experiment where outliers are excluded. They show, that active learning methods do not perform better than random sampling if NMS is not applied. Pseudo labelling adds the most improvement in the early training cycles and the added improvement continues over all cycles but saturates towards the end.

### 2.1.3 Semantic Segmentation

As the labelling effort for semantic segmentation is tremendously high, [Mac+19] developed an active learning method that automatically selects regions for labelling instead of the whole image. The authors introduce a cost model behind the segmentation model, that estimates the clicks that a human annotator would have needed to annotate the image. Using the sliding-window approach, the most informative regions from the information map are selected and at each location, the values are accumulated. The same procedure is applied to the cost map. The information and cost map are fused, and non-maximum suppression is applied to retrieve region candidates. The top-scoring region proposals are selected for labelling. In a detailed ablation study it is shown, that the combination of information content and cost estimates is very powerful. On the Cityscapes dataset [Cor+16], the proposed method achieved 95% of the full training set's performance while requiring only 17% of the labelling effort.

Other region-based methods are proposed in [Kas+19]. The authors calculate the entropy on an image-level and pixel-level. Furthermore, they use a canny edge detector to identify edge pixels and calculate the entropy for the edge pixels. As most pixels have some kind of relation to their neighbouring pixels, it is useful to consider the spatial correlation. The authors use super-pixels to compute the entropy on a super-pixel level. Super-pixel algorithms divide an image into non-overlapping regions by grouping similar pixels together. Since most pixels within a super-pixel are from the same semantic category, natural object boundaries are preserved. This way the need for polygon and intersection clicks is reduced, making the annotation of the pixels a lot easier. Using this approach the authors achieved 93.8% accuracy while using only 10% of the annotations compared to the baseline on the Cityscapes dataset.

The method proposed in [SVN19] uses multiple viewpoints to measure the inconsistency

between the single predictions. They furthermore also use super-pixel labelling to reduce the labelling effort and show that the results are still good performing. They accomplish an accuracy of 95% compared to the maximum model performance with just 13% labelling effort.

A method that can be used on single viewpoint datasets, such as Cityscapes, is proposed in [Xie+20]. The authors add another branch to the segmentation network that learns to predict a semantic difficulty score for each pixel. This branch is supervised by an error mask that is generated by the actual segmentation branch. For the error mask each pixel value is set to one if the prediction is not equal to the ground truth, and to zero otherwise. Two aggregation functions are proposed, the difficulty-aware uncertainty score and the difficulty-aware semantic entropy. Incorporating the semantic difficulty to select the most informative samples helps to leverage the performance, especially for hard areas. The method reaches the upper performance bound of the full training data with just about 60% of the data.

For semantic segmentation, the labelling costs are particularly high, which is why [Col+21] focuses on reducing the actual labelling cost by approximating the segmentation contours to estimate the number of required clicks during the labelling process. They train a meta-regression model to estimate the segment-wise IoU of each predicted segment of unlabeled images which results in priority maps. By combining the two methods, they target informative regions with low annotation costs. Their method is requiring, depending on the architecture, only 10-30% annotation costs for achieving 95% of the full set Mean Intersection over Union (mIU).

In [Bel+21] pseudo-labelling is used to further reduce the need for annotations. Their algorithm selects a subset $U'$ for the oracle to label, furthermore, another subset $U''$ is generated based on the K-Nearest Neighbours (K-NN) samples of $U'$. The subset $U'$ will contain correct annotation generated by the oracle. The set $U''$ will contain pseudo-labels generated by the model. In the next training cycle, the model is trained on the combined set of pseudo and correct labels. The authors claim that their proposed method outperforms other active learning methods, namely Entropy and MC Dropout.

Most of the mentioned publications count the number of images used during the training to measure the saved annotation effort but neglect the important fact, that not every data sample has the same annotation cost. Especially for semantic segmentation, the labelling effort difference between two images can be very high, depending on the amount, size and diversity of objects contained in the image. Following this fact, it is important to incorporate the labelling costs into the data selection process. Cai et al. revisit in [Cai+21] the super-pixel based approach used in e.g. [SVN19; Kas+19] and take a realistic click-based annotation cost estimation into account. They point out that simply taking the percentage of labelled pixels cannot evaluate the effectiveness of super-pixel based approaches. In addition, they introduce a class-balanced sampling scheme to overcome the degraded performance for under-represented classes resulting from datasets with imbalanced class distributions. In their experiments, the authors show that a super-pixel approach outperforms the traditional rectangle-polygon approach. Furthermore, the introduced class-balanced acquisition function achieves better results than random or uncertainty-based selection.

### 2.1.4 Task Agnostic

Not all active learning methods are bound to a specific task. The aforementioned methods such as Ensembles [HS90; Bel+18] or MC Dropout [GG15] can be used in models with various tasks. In [YK19] a so-called loss parametric module is proposed. With that, the network learns to predict the target losses of unlabeled inputs, which makes it task-agnostic. The predicted loss for unlabeled data inputs is used to select which data samples should

be trained next. The efficiency of this method is shown for various tasks, namely image classification, object detection and human pose estimation. In all three tasks, the learning loss approach outperforms random, entropy and core-set sampling, making it an effective and versatile method.

Agarwal et al. [Aga+20] observe a gap in the active learning literature, where prior methods do not capture the diversity in the spatial and semantic context of an image. They introduce a new distance metric called Contextual Diversity, which aims to capture the diversity in the spatial and semantic context. They replace the Euclidean distance in the core-set [SS18] approach with contextual diversity, which results in superior performance compared to the other evaluated methods, inter alia, random sampling, core-set [SS18] and MC Dropout [GG15]. Their findings are validated for image classification, object detection as well as semantic segmentation.

[Liu+21] proposes an influence selection task- and model-agnostic active learning algorithm. It calculates the influence of a sample by estimating its expected gradient and queries the samples with the greatest expected influence for the labelling. This approach is evaluated on both image classification and object detection. It is compared with random sampling, core-set sampling [SS18], learning loss sampling [YK19] and localization stability sampling [Kao+19]. To achieve the same accuracy, the proposed method required 13% fewer annotations than core-set and even 26% less than localization stability sampling.

### 2.1.5 Multi-Task

At the time of writing this thesis, no work on multi-task active learning in the context of autonomous driving could be found. However, two papers deal with Multi-Task Active Learning (MTAL) for Natural Language Processing (NLP). Reichart et al. [Rei+08] introduce two MTAL protocols that could be applied in any Multi-Task setting. The first protocol is called Alternating Multi-Task Active Learning and as the name suggests the data selection strategy is alternated in every cycle. In each of these cycles, a single one-sided learning method is used to determine the next cycle's dataset. A one-sided learning method focuses on solely one task and only uses the knowledge in this task domain for the data selection. The second protocol is more advanced as it combines knowledge about all task domains. For each task, a usefulness score is calculated and translated into a rank. The rank numbers are summed for each sample to form a unified rank. The samples with the lowest rank numbers are selected for the annotation and thus for the next cycle. Both protocols are compared against random sampling as well as the one-sided learning methods. The evaluation shows, that one-sided learning performs in the designated task, but not necessarily on the other tasks. The two proposed MTAL protocols outperform random sampling and extrinsic learning. Extrinsic learning is if a task is trained using data that has been selected by a selection that focuses on another task. Therefore, the authors conclude that their MTAL protocols provide a good trade-off between good performance on all tasks while keeping the annotation costs low. Furthermore, they point out, that even though the more sophisticated rank combination protocol performs better than the alternating protocol, the margin between those two is relatively small.

Ikhwantri et al. [Ikh+18] use MTAL for Semantic Role Labeling and Entity Recognition and use both protocols proposed by [Rei+08]. However, they slightly change the alternating protocol. Instead of alternating the one-sided learning method to select which samples should be labelled, they randomly select the task that is used for the selection process. The authors claim that active learning requires 12% fewer data compared to passive learning and, in some scenarios, even outperforms the one-sided learning methods in the respective task.

## 2.2 Multi-Task Learning

In autonomous driving two of the many required tasks are object detection and semantic segmentation. While the localization and recognition of objects is urgently needed to obtain information about dynamic objects and pass it on to the path planning modules, semantic segmentation is particularly useful to obtain information about free space, static objects and contextual information in general. A lot of research has been done for both tasks and the trend of state-of-the-art methods is towards deep neural networks. The quite good results have one disadvantage. They are computationally exhaustive. But since both tasks are applied to the same domain, namely a camera image of the environment around the car, it makes sense to share some of the computation for both tasks. Multi-Task architectures aim to solve both tasks simultaneously, which in practice often is a compromise between a good detection quality and the real-time requirement. One of the first architectures that was able to improve the accuracy of both object detection and semantic segmentation, while sharing some of the computational efforts, is called BlitzNet and was published by Dvornik et al. [Dvo+17]. The proposed architecture follows a fully convolutional approach where all model weights are shared until the last layer. To perform the actual prediction, a single layer is added for each task. This way the authors reached a mIU of up to 72.8% and an 80.0% Mean Average Precision (mAP) on the Pascal VOC 2012 validation set [Eve+15] while still meeting real-time inference speed of up to 24 frames per second.

A more recent model was proposed by Peng et al. [Pen+20]. Just like BlitzNet, this model uses a shared ResNet50 backbone [He+15], but it is extended with three additional layers. Furthermore, the authors designed a novel initialization mechanism to propose more object candidates, called PriorBox. It can generate dense object candidates with special aspect ratios, which is beneficial for object detection in complex and dynamic traffic scenes. For the segmentation, they use Multi-Scale Atrous Convolution (MSAC) to boost the performance in small areas. The combination of PriorBox and MSAC results in the highest accuracy on the Cityscapes dataset compared to BlitzNet [Dvo+17] and other approaches that were published in the meantime [CPL18; Che+18]. The authors achieved a 40.0% mAP on the classes that are relevant for autonomous driving and a 55.5% mIU for the semantic segmentation task on the Cityscapes validation set.

In [Sal19] Niels Ole Salscheider proposed a network architecture that concurrently performs 2D object detection and semantic segmentation. The two tasks share a backbone that is based on ResNet-38 [WSH16] and has been adapted to reduce computational costs. The backbones output is then the input for the two branches that perform the object detection and semantic segmentation, respectively. The later branch has a convolutional encoder-decoder structure, the final layer reduces the number of channels to the number of classes and applies a softmax function. It is trained using cross-entropy loss. The branch for object detection shares three ResNet modules and is then split into four separate sub-networks of identical structure. The first sub-network predicts whether an anchor box contains a relevant object in a binary classification manner. The second sub-network predicts the object class for each anchor box that contains an object. The third sub-network gives the bounding box parameters, namely the 2D coordinates as well as the height and width. The last sub-network is optional and can be used to learn a feature embedding for each detected object. The losses for each sub-network are focal loss, cross-entropy loss, L1 loss and contrastive loss, respectively. It achieves an mAP of 56.47% on the object detection and a mIU of 73.1% for the semantic segmentation on the Cityscapes dataset. On the KITTI dataset [GLU12] it achieves an object detection average precision of 69.3% for cars and 67.7% for pedestrians. Even if the accuracy is surpassed by other single-task architectures, this architecture convinces with its speed. It runs at 10 Hz on 1MP images on current hardware built into autonomous vehicles.

# Chapter 3

# Approach

In the following Section 3.1 the technical details of the investigated methods used to select the most promising samples from the unlabeled data pool are presented. First, the existing baseline methods Least Confidence and 1-vs-2 margin sampling will be described in the Sections 3.1.1 and 3.1.2. In Section 3.1.3 another baseline method is presented, which is then in Section 3.1.4 adapted to a novel segmentation-focused method. In Section 3.1.5 it is shown how the existing loss prediction module can be applied to object detection and semantic segmentation. Finally, the newly introduced $BoxMask$ approach is presented in Section 3.1.6. In the last Section 4.7, the possibilities of combining two or more methods are discussed.

## 3.1 Sample Selection Strategies

As comparing against the results achieved using all data from the first training step is not fair, all methods are compared against Random selection. Each image in the dataset gets a random value between zero and one assigned. In each cycle, a fixed amount of data is selected using the images with the lowest scores. While having the same amount of data in each cycle, the goal of any method mentioned in the following section is to select a subset of the remaining data that results in higher performance concerning 2D object detection, semantic segmentation or both of these two. In this thesis, the focus lies on single-model methods. This means, that the data selection is solely based on the predictions of one single model. This stands in contrast to Ensemble methods, which train multiple models on the same data and use the variance in the models' predictions to select the images for the next training cycle. Ensemble-based methods have the disadvantage of increased need for resources as well as increased time consumption, as several models need to be trained in each cycle. The effectiveness is controversial, some say that Ensembles provide better results compared to e.g. MC Dropout [Sch+20], while others claim the opposite [Bel+18]. In addition, most of the methods covered here could also be used in an Ensemble setting. Therefore, applying these methods in an Ensemble setting remains for future work and will not be covered in this thesis.

### 3.1.1 Least Confidence

In most state-of-the-art object detection methods, each bounding box has an objectness score which represents the networks' certainty about an object being present in that area of the image. This score can be used to measure the general uncertainty of the network given an unlabelled image. The inference is run for each image in the unlabelled data pool. The

amount of bounding boxes is reduced using non-maximum suppression. To get a unified score for each image, the confidence scores of each predicted bounding box are aggregated. Three different aggregation methods are used in this thesis. The first is taking the maximal confidence $c$ of the bounding box $d$ for all detected boxes $D$ in an image $i$.

$$S_{max}(i) = max_{d \in D_i} c_d \tag{3.1}$$

The second aggregation method calculates the sum of the confidences of all detected bounding boxes for each image.

$$S_{sum}(i) = \sum_{d \in D_i} c_d \tag{3.2}$$

The last method is taking the average of the confidences of all detected bounding boxes as the total score.

$$S_{average}(i) = \frac{1}{|D_i|} \sum_{d \in D_i} c_d \tag{3.3}$$

The samples with the lowest scores are selected for labelling and are added to the training dataset afterwards.

### 3.1.2 One Versus Second Margin

The three previously described aggregation methods are used by Brust et al. [BKD19] as well. But instead of using just the objectness score of the bounding box, they introduce the 1-vs-2-margin score. It can be understood as the difference between the most likely class $cl$ and the second most likely class of the predicted bounding box and is defined as:

$$S_{1vs2}(d) = (1 - (max_{cl_1 \in C} \hat{p}(cl_1|d) - max_{cl_2 \in C} \hat{p}(cl_2|d)))^2 \tag{3.4}$$

The resulting scores for each bounding box can be aggregated to a single score using either the maximum, the sum or the average of all scores $S_{1vs2}$. But instead of selected the images with the lowest scores, those with the highest scores are passed to the annotation.

### 3.1.3 Inconsistency for Object Detection

A more sophisticated method has been proposed by Elezi et al. [Ele+21b]. In their approach, the inference is run not only for the original image but also for the horizontally flipped image. They then measure the inconsistency between the two sets of predicted bounding boxes and select the image with the highest inconsistency for annotation. The inconsistency score is calculated in three steps. First, the predictions $D$ of the original image and the predictions $\hat{D}$ of the augmented image are matched by their intersection over union. For each matched pair of bounding boxes, the inconsistency is calculated using the KL-divergence of the class probabilities $cl'$ and $\hat{cl}'$:

$$L_{con}(cl'_d, \hat{cl}_d{}') = \frac{1}{2}\big(KL(cl'_d, \hat{cl}_d{}') + KL(\hat{cl}_d{}', cl'_d)\big) \tag{3.5}$$

The overall inconsistency score of an image is the maximum of the bounding box inconsistency scores:

$$I(D_i, \hat{D}_i) = max_{d \in D_i}(L_{con}(cl'_d, \hat{cl}_d{}')) \tag{3.6}$$

In addition to the inconsistency, the entropy of the original image predictions $D$ is calculated:

$$H(D) = max_{d \in D}(H(cl_d)) \qquad where \quad H(x) = -\sum_{i=1}^{N} P(x_i) \cdot logP(x_i) \tag{3.7}$$

Finally, the inconsistency and entropy are unified to a single image score:

$$S_{inconsistency}(i) = H(D_i) \cdot I(D_i, \hat{D}_i) \tag{3.8}$$

Just like in 1-vs-2-margin sampling, the images with the highest scores are given to the annotator.

In an earlier publication, Elezi et al. [Ele+21a] claim the labelling of only images that contain difficult objects could lead to a distribution drift as those objects might not be good representatives of the dataset. To prevent this issue, they take the objectness score of a bounding box into account. This results in a bounding box score defined as:

$$S_{inconsistency}(d) = \begin{cases} H(d) \cdot I(d, \hat{d}), & \text{if } c_d \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

and an overall image score:

$$S_{inconsistency}(i) = max_{d \in D_i}(S_{inconsistency}(d)) \tag{3.10}$$

### 3.1.4 Inconsistency for Semantic Segmentation

Inspired by the prior method, two novel methods were developed. They are similar to the methods proposed by Elezi et al. [Ele+21b; Ele+21a] but focus on the Semantic Segmentation task. Again, the predictions are generated for the original image and its horizontal flipped version. In a naïve approach, the inconsistency is measured by comparing the two predicted classes of the original image pixel $p_j$ and the augmented image pixel $\hat{p}_j$ for each pixel of the image. The image score is defined as:

$$S_{inconsistency_{seg}}(i) = \sum_{j=0}^{height \cdot width} \begin{cases} 1, & \text{if } max_{cl \in C} p_j(cl) \neq max_{cl \in C} \hat{p}_j(cl) \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

An advanced strategy for semantic segmentation works like the method by Elezi et. [Ele+21b], but instead of using the class predictions of the bounding boxes, the pixel class predictions $cl'_j$ and $\hat{cl}'_j$ are used.

$$L_{con}(cl'_j, \hat{cl}'_j) = \frac{1}{2}\big(KL(cl'_j, \hat{cl}'_j) + KL(\hat{cl}'_j, cl'_j)\big) \tag{3.12}$$

$$S_{inconsistency_{seg+KL}}(i) = \frac{1}{height \cdot width} \cdot \sum_{j=0}^{height \cdot width} H(p_j) \cdot L_{con}(p_j, \hat{p}_j) \tag{3.13}$$

For both the naïve and the more advanced method, the images with the highest score are selected for annotation.

### 3.1.5 Loss Prediction Module

The loss prediction module presented in [YK19] is task-agnostic and can be used for both of the applications examined here. The network is extended by two loss prediction modules that learn to predict the loss of one of the two tasks, respectively. Three intermediate layer outputs from both, the backbone and the respective task head, are used as input for the loss prediction module. Global Average Pooling, followed by a Fully-Connected and Rectified Linear Unit (ReLU) activation layer, is applied to each input layer. This way, they are all reduced to the same size and can be concatenated. Finally, a Fully-Connected layer is applied to get a single loss prediction value. The gradient of the loss prediction module is stopped at its input layers, such that the optimization of the loss prediction module has no impact on the learning of the actual network. The authors of the method claim that simply using the Mean Squared Error (MSE) between the target loss and the predicted loss does not perform well and is just learning the loss scale instead of its exact value. They, therefore, propose to use a mini-batch approach, where prediction pairs are used to learn the loss prediction. The loss prediction loss on a sample pair $\{x^p = (x_i, x_j)\}$ consists of the task loss $l$ and the loss prediction $\hat{l}$.

$$\begin{aligned} L_{loss}(l^p, \hat{l}^p) &= max\big(0, -\mathbb{1}(l_i, l_j) \cdot (\hat{l}_i - \hat{l}_j) + \xi\big) \\ s.t. \quad \mathbb{1}(l_i, l_j) &= \begin{cases} +1, & \text{if } l_i > l_j \\ -1, & \text{otherwise} \end{cases} \end{aligned} \tag{3.14}$$

The overall loss given a batch $B$ consisting of the predictions $\hat{y}$ and the corresponding groundtruths $y$ is defined as:

$$\begin{aligned} L_{pred}(B) = &\frac{1}{|B|} \sum_{(y,\hat{y}) \in B} \delta_{obj} \cdot L_{obj}(\hat{y}, y) + \delta_{seg} \cdot L_{seg}(\hat{y}, y) \\ &+ \frac{2}{|B|} \sum_{(y^p,\hat{y}^p) \in B} \delta_{lossObj} \cdot L_{lossObj}(\hat{y}^p, y^p) + \delta_{lossSeg} \cdot L_{lossSeg}(\hat{y}^p, y^p) \end{aligned} \tag{3.15}$$

where $L_{obj}$ is the Loss of the object detection task, $L_{seg}$ is the Loss of the semantic segmentation task. $L_{lossObj}$ and $L_{lossSeg}$ are the losses of the loss prediction modules for the respective task. $\delta$ is the weight for the specific loss. A visualization of the architecture and its loss composition is provided in Figure 3.1.

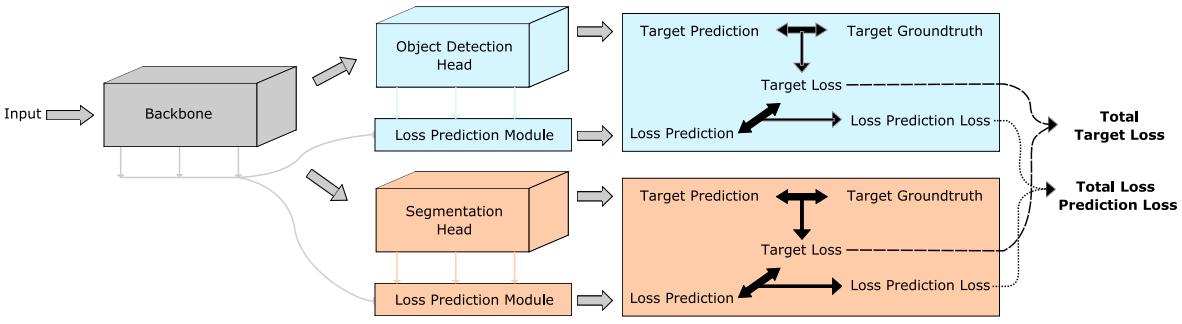



**Figure 3.1:** Visualization of the network architecture consisting of a backbone, two task heads, as well as a Loss Prediction Module for each task. The figure does also show how the overall Loss is composed.

### 3.1.6 Box Mask

The two tasks, object detection and semantic segmentation are closely related and it has been shown, that the two can benefit from each other when combined [Dvo+17]. In a Multi-Task setting like it's used in this thesis, the decision on which samples to select must not only be based solely on one task. It makes sense to incorporate the predictions of both tasks and calculate the scores of the remaining samples in a combined and unified scoring method. A novel approach that does this is developed in this thesis. The idea is to match the object detection predictions with the predictions of the semantic segmentation and thus, measure the inconsistency between the two tasks. The samples with the highest inconsistency are selected for the next cycle. To calculate the inconsistency a mask is drawn using the predicted bounding boxes from the object detection. Each pixel that is within a bounding box gets the class label as its value. Therefore, this generated mask has the same format as the predictions from the semantic segmentation. For each pixel is checked, whether the box mask label is the same as the segmentation label. If not, the sample score is increased by one. The method will be referenced as $BoxMask$. Its pseudo-code of the algorithm is given in Algorithm 1 and the resulting pixel-mask is shown in Figure 3.2 b). To avoid an imbalance

**Algorithm 1** The pseudo code of the BoxMask scoring method.

**function** BOXMASK($boxes, labels$) ▷ bounding boxes & segmentation labels
  $box\_mask = zeros\_like(labels)$

  **for** $box \in$ boxes **do**
    $min_y, max_y, min_x, max_x = get\_box\_edges(box)$
    $box\_mask[min_y : max_y, min_x : max_x] = box.class\_label$
  **end for**
  $S_{BoxMask} = 0$
  **for** $i \in$ height **do**
    **for** $j \in$ width **do**
      **if** $box\_mask[i, j] \neq labels[i, j]$ **then**
        $S_{BoxMask} = S_{BoxMask} + 1$
      **end if**
    **end for**
  **end for**
  $S_{BoxMask_{Norm}} = S_{BoxMask}/|boxes|$
  $S_{BoxMask_{PixelNorm}} = S_{BoxMask}/(height_{img} \cdot width_{img})$
  $S_{BoxMask_{BN}} = S_{BoxMask}/count\_non\_zero(box\_mask)$
**end function**

between images that contain a lot of objects and therefore potentially a lot of bounding boxes, and images that contain only a few or no objects at all, there are also improved versions introduced that normalize the score based on either the number of detected bounding boxes, the total image pixels or the size of the specific bounding box. The normalization methods are indicated with the subscript $Norm$, $PixelNorm$ and $BN$ (BoxNorm), respectively. During the analysis of the $BoxMask$ approach, it quickly became clear that a lot of background noise flows into the score calculation. This is clearly visible in Figure 3.2, where the ratio between background pixels and object pixels is unbalanced. Therefore, the approach was further developed to minimise the influence of background noise. A first approach was to reduce the bounding box mask to an ellipse instead of a rectangle. In the datasets used, the number of classes that fit this rounder shape, e.g. cars, clearly outweighs the number of classes that are better suited for a rectangular shape, e.g. trucks. As in the $BoxMask$

approach, for each detected bounding box an ellipse with 50% of the height and width of the box around the centre of the box was drawn into the mask. The resulting output of the $BoxMaskEllipse$ ($BoxMaskEl$) approach is visualized in Figure 3.2 c). In the second approach to reduce the background noise, the mask generation was turned around. Instead of first extracting the bounding boxes and then counting the segmentation pixels in the area of the boxes, in the $BoxMaskSegmentation$ approach the segmentation results are extracted first. In the second step, the bounding box mask is generated considering only the areas where an object has been detected by the semantic segmentation. The resulting mask output is shown in Figure 3.2 d). In the approaches discussed so far, the pixels are counted where the prediction of the object recognition and the prediction of the semantic segmentation do not match. Predictions that are completely different, such as *Truck* and *Pedestrian*, are treated the same as predictions that are very similar, such as *Truck* and *Bus*. In order to penalize the cases where the predictions differ significantly, the $BoxMask$ approach has been further developed and in $BoxMask_{KL}$ the class probabilities are taken into account for the sample score calculation. For this purpose, the class probabilities of an object predicted by the object detection are transformed into the same probability domain as the class probabilities predicted by the semantic segmentation. Each pixel that is within the area of the detected bounding box is assigned the class distribution that has been predicted for the object. The probability of the classes that are not trained in object recognition is set to $10^{-9}$. The score of a sample is then calculated from the transformed object class probability distribution $P_{od}(j)$ and the class probability distribution of the segmentation $P_{seg}(j)$ at a pixel $j$ for each pixel that lies within an object as follows:

$$S_{BoxMask_{KL}}(i) = \frac{1}{2} \sum_{j \in box\_mask} KL\big(P_{seg}(j), P_{od}(j)\big) + KL\big(P_{od}(j), P_{seg}(j)\big) \cdot \frac{1}{|box\_mask|} \quad (3.16)$$

## 3.2 Alternating Methods

The methods just presented all differ in how they work and each has its advantages and disadvantages. However, they all focus primarily on one task. In a multi-task network, several tasks are performed simultaneously and it is of course of interest that both tasks achieve the best possible results. Therefore, it makes sense to combine different methods of active learning. In the publications by Reichart et al. [Rei+08] and Ikhawantri et al. [Ikh+18] it was investigated to what extent the alternation of two active learning methods influences the performance of a multi-task network for natural language processing. Both works use just two single methods and alternate these two either based on a random selection or based on the task domain knowledge. Since there are several methods available, the alternation should not be limited to just two methods. For each training cycle, one could always use the method that is best performing on object detection or the method that has the best results on semantic segmentation, or the method that offers the best trade-off of the two tasks at the specific cycle. One could also just use the overall best method for object detection and alternate it with the overall best method for semantic segmentation. Another option would be to use the method that had the highest increase in performance compared to the previous cycle. And since the annotation effort is also an important aspect to consider, one could also alternate between the methods that result in the lowest annotation costs. Not only the method itself could be alternated but also the metric on which the best intermediate checkpoint selection is based on. This means one could select the checkpoint that performs best on object detection in the first training cycle and use the same method in the next cycle, but this time one uses the

**(a)** An input image taken from the nuImages dataset

**(b)** BoxMask output

**(c)** BoxMaskEllipse output

**(d)** BoxMaskSegmentation output

**Figure 3.2:** A visualization of the BoxMask approaches. White pixels represent that the prediction of segmentation and object detection are equal. All gray colored pixels are the areas where the two task predicted different classes.

checkpoint that has the highest performance on semantic segmentation. It becomes clear that the possible combinations are endless. However, due to the time and resource limitations of this thesis, a selection of the combinations must be done. The most promising method combinations are described in Section 4.7. The general procedure of the alternating methods is always the same. In the first cycle, the data is randomly selected. For all subsequent cycles, a new decision is made on which method to use for each cycle, depending on the approach.

# Chapter 4

# Evaluation

The evaluation chapter is structured as follows. First, the general experiment setup is described in Section 4.1. This includes the model architecture, the used datasets, the active learning framework and the evaluation metrics. In the Sections 4.2-4.6 the experiment setup for the single-methods are presented, followed by the combined methods in Section 4.7. Lastly, most of the methods are compared against each other on different datasets. The comparison settings are described in Section 4.8.

## 4.1 Experiment Setup

### 4.1.1 Model Architecture

Since this thesis aims to investigate active learning in the context of autonomous driving, the speed of the applied architecture plays a major role. The choice, therefore, fell on the architecture presented by Salscheider [Sal19], which is described in more detail in Section 2.2. The corresponding public code [Sal20] was used as a basis for this thesis and adapted by the following changes and additions. Except for the full training baseline, the learning rate is kept steady, instead of decreasing the learning rate continuously while the training progresses. A decreasing learning rate showed worse results in the experiments, which could be due to the fact that new data is added to the training set in each cycle. All the available losses, except the embedding loss, are used and their weights are kept. The architecture is extended by a loss prediction module for both object detection and semantic segmentation based on the method proposed in [YK19]. The output of three intermediate backbone layers and three intermediate task head layers is used as input for the loss prediction module. In addition, depending on the task, the output of another three layers of the respective task branch is added to the input of the loss prediction module. Using global average pooling, all inputs are downscaled to the same size. Then a fully-connected layer and a ReLU activation are applied to each input layer. Finally, all the layers are concatenated and passed to a single linear layer to produce the loss prediction. The gradient for the input layers of the loss prediction module is stopped during the propagation, as this showed better results in both object detection and semantic segmentation. Another change to the original code from Salscheider was the addition of a softmax layer to the segmentation labels and object detection class predictions to get, not only the most likely class but also the full distribution. This makes selection methods, like the 1-vs-2 margin sampling possible. In all experiments, if not stated otherwise, the overall loss $L$ is composed as the weighted sum of the segmentation loss $L_{seg}$, the object detection loss $L_{obj}$ and the loss prediction loss $L_{pred}$:

$$L = \delta_{seg} \cdot L_{seg} + \delta_{obj} \cdot L_{obj} + \delta_{pred} \cdot L_{pred} \tag{4.1}$$

with $\delta_{seg} = 50.0$, $\delta_{obj} = 1.0$ and $L_{pred} = 1.0$ if the loss prediction module is activated, $L_{pred} = 0.0$ otherwise. $L_{seg}$ is defined as the cross-entropy loss, $L_{pred}$ is defined in Equation 3.15. The object detection loss again consists of three losses from the sub networks, namely the object loss $L_{rel}$ that comes from the sub network which solves a binary classification problem and decides whether an anchor box contains a relevant object. A Focal loss with $\alpha = 1.0$ and $\gamma = 2.0$ is used here. The second loss $L_{cls}$ is defined as a cross-entropy loss and comes from the sub network that predicts the objects class. The last sub network predicts the bounding box parameters and is trained using a smooth L1 loss $L_{box}$. Thus, the overall loss for the object detection is defined as:

$$L_{obj} = \delta_{rel} \cdot L_{rel} + \delta_{cls} \cdot L_{cls} + \delta_{box} \cdot L_{box} \tag{4.2}$$

with $\delta_{rel} = 0.1$, $\delta_{cls} = 50.0$ and $\delta_{box} = 100.0$. In all experiments a batch size of 4 and a learning rate of 0,001 was used.

### 4.1.2 Datasets

#### NuImages

As this thesis required semantic segmentation labels, the NuImages subset of the nuScenes dataset has been used. The 2019 published nuScenes dataset [Cae+19] provides 3d object annotations from 1000 scenes in the cities of Boston and Singapore. This dataset has been extended with NuImages, which adds another 93,000 images to the dataset. NuImages offers 2d annotations for around 800,000 foreground objects and 100,000 semantic masks. A total of 93,000 images are taken by six different cameras pointing to the front, back and sides. In this thesis, only a subset of the NuImages dataset was used. This is because the same object could potentially occur in multiple images that were taken by different cameras at the same time, which could cause a distortion of the results of the active learning methods. To match the domain with the one from other datasets like Cityscapes [Cor+16], only the images taken from the front camera were used. This resulted in 13,187 images for the training set and 3,249 images for the validation set. The selected subset has a total of 138,569 objects. The class distribution is shown in Figure 4.1. This dataset is particularly interesting for this work because active learning has already been used for scene selection. This allows the effectiveness of the methods studied to be demonstrated even better. The augmentations motioned in Section 4.1.3 were all applied except for the horizontal flipping, which was deactivated on this dataset. The images of both the training and the evaluation set were downscaled to 1536 x 768 pixels.

#### A9

The A9 dataset [Cre+22] offers camera and Light Detection And Ranging (LiDAR) frames from two overhead gantry bridges on the A9 autobahn near Munich, Germany, with the corresponding objects labelled with 3D bounding boxes. As the cameras are static, the background is static as well. Therefore, this dataset provides an interesting insight into how the actual objects, compared to the background of the scene, influence the sample choice of active learning methods. To match the task domain with the other experiments, the 3d labels were transformed into 2d labels. And since the dataset does not provide labels for semantic segmentation, these were generated using another high performing model. The model

| Object class | # Train | # Validation | # Total |
|---|---|---|---|
| *Car* | 55,138 | 12,758 | 67,896 |
| *Pedestrian* | 36,764 | 8,808 | 45,572 |
| *Truck* | 11,874 | 2,866 | 14,740 |
| *Motorcycle* | 3,342 | 806 | 4,148 |
| *Bicycle* | 2,513 | 608 | 3,121 |
| *Bus* | 2,355 | 737 | 3,092 |
| **Total** | **111,986** | **26,583** | **138,569** |

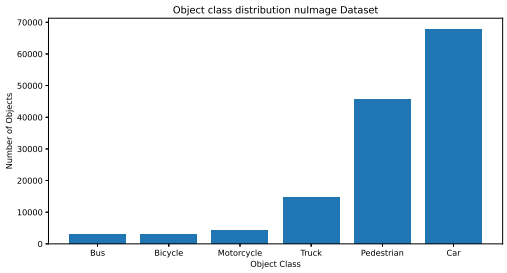



**Figure 4.1:** Object class distribution of the NuImages dataset.

| Object class | # Train | # Validation | # Test | # Total |
|---|---|---|---|---|
| *Car* | 17,299 | 8,133 | 2,843 | 28,275 |
| *Truck* | 7,646 | 3,702 | 1,244 | 12,592 |
| *Bus* | 32 | 24 | 3 | 59 |
| *Motorcycle* | 18 | 14 | 1 | 33 |
| *Pedestrian* | 19 | 1 | 0 | 20 |
| *Other* | 233 | 106 | 38 | 377 |
| **Total** | **25,247** | **11,980** | **4,129** | **41,356** |

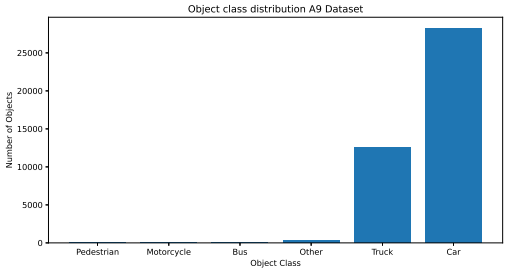



**Figure 4.2:** Object class distribution of the A9 dataset. For the experiments only the classes *Car, Truck, Bus* and *Motorcycle* were considered. All other classes are merged into *Other* and are ignored during the evaluation.

of choice is the Mask R-CNN X152 model from the detectron2 [Wu+19] model zoo, as the inference speed was not an issue and its accuracy is high. It was trained on the ImageNet [Den+09] dataset and resulted in good annotations for the A9 dataset. Nevertheless, it must be noted in the following results that the annotation for semantic segmentation does not match the accuracy of human annotation. Only the labels containing an object such as a car, bus, truck or motorcycle were labelled. All the other pixels were marked with the ignore label and were not considered during training and evaluation. The dataset used for this training consists of 1,440 images split into a training, validation and test set (60%, 30% and 10%). An overview of the class distribution is given in Figure 4.2. Again, all augmentations mentioned in Section 4.1.3, except for the horizontal flipping, have been applied to the input images. A downsizing to 1008 x 632 pixels during training has been done. During the evaluation, the image size remains at 1344 x 840 pixels.

### Cityscapes

The Cityscapes dataset [Cor+16] is a large-scale collected set of urban street scenes captured by a stereo camera placed at the front windshield of a car. The data was collected from 50 cities over a longer time period and thus at different lightning and weather conditions. It provides around 5,000 images with fine annotations and around 20,000 additional images that were coarsely annotated. The dataset provides semantic, instance-wise, and dense pixel annotations for 30 classes. The data is split into a training, validation and test subset, containing around 282,500 fine annotated objects and around 694,000 coarsely annotated objects. The ground-truth of the test set is not available for public. A detailed listing of the class distribution of the fine annotated subsets can be found in Figure 4.3, the distribution of the coarsely annotated subsets is visualized in Figure 4.4. In contrast to the other two datasets, randomly flipping the image along the horizontal axis has been applied as augmentation in addition to the other augmentations described in Section 4.1.3. To reduce the time needed to run the experiments the training input image size was reduced to 1152 x 576 pixels and

| Object class | # Train | # Validation | # Test | # Total |
|---|---|---|---|---|
| *Car* | 27,155 | 4,667 | - | 31,822 |
| *Truck* | 489 | 93 | - | 582 |
| *Bus* | 385 | 98 | - | 483 |
| *Motorcycle* | 739 | 149 | - | 888 |
| *Pedestrian* | 17,994 | 3,419 | - | 21,413 |
| *Bicycle* | 3,729 | 1,175 | - | 4,904 |
| **Total** | **50,491** | **9,601** | **-** | **60,092** |
| *Other* | 185,199 | 37,208 | - | 222,407 |
| **Total** | **235,690** | **46,809** | **-** | **282,499** |

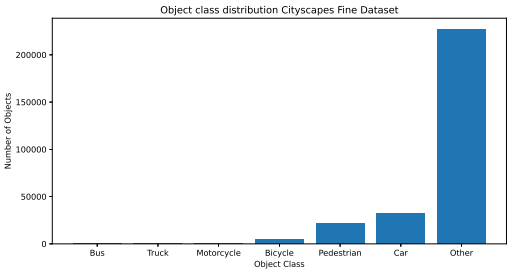



**Figure 4.3:** Object fine annotated class distribution of the Cityscapes dataset. For the experiments only the classes *Car, Truck, Bus* and *Motorcycle* were considered. All other classes are merged into *Other* and are ignored during the evaluation.

| Object class | # Train | # Train Extra | # Validation | # Total |
|---|---|---|---|---|
| *Car* | 3,290 | 24,107 | 545 | 27,942 |
| *Truck* | 155 | 644 | 39 | 838 |
| *Bus* | 127 | 559 | 46 | 732 |
| *Motorcycle* | 344 | 1,229 | 65 | 1,638 |
| *Pedestrian* | 3,921 | 13,026 | 756 | 17,703 |
| *Bicycle* | 1,797 | 5,177 | 497 | 7,471 |
| **Total** | **9,634** | **44,742** | **1,948** | **56,324** |
| *Other* | 82,392 | 539,772 | 15,580 | 637,744 |
| **Total** | **92,026** | **584,514** | **17,528** | **694,068** |

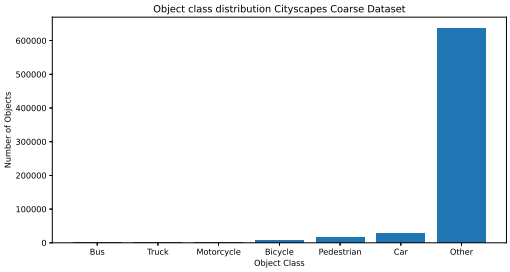



**Figure 4.4:** Object coarsely annotated class distribution of the Cityscapes dataset. For the experiments only the classes *Car, Truck, Bus* and *Motorcycle* were considered. All other classes are merged into *Other* and are ignored during the evaluation.

to 1536 x 768 during evaluation.

### 4.1.3 Active Learning Framework

All the used datasets have the full training set already labelled. However, to evaluate the effectiveness of a selection strategy it is assumed that the labels are unknown. In the beginning, all data samples are considered to be in the unlabeled data pool. As the model is not trained yet, no assumptions about the usefulness of the data samples can be made yet. Therefore, 30% of the training data is selected randomly and moved to the labelled data pool. The model is trained on this initial labelled data pool and the resulting model state is used as starting point for all the other methods. From here each cycle follows the same procedure consisting of three steps. First, the best checkpoint of the last training cycle is selected. This can be done using the evaluation results from either the object detection or the semantic segmentation. For object detection, the mean average precision is used to select the best performing intermediate model state. For the semantic segmentation, the mean intersection over union was used as the difference between the methods was higher using this metric. In the second step, the remaining data samples in the unlabeled pool are evaluated using the current model and the specified score is calculated for each sample. The samples are then ranked by their scores and the highest-ranked samples are moved to the labelled data pool. The size of this pool is therefore increased by 10% of the full training dataset size in each cycle. The last step of a cycle is the model update. The weights of the previous checkpoint are loaded and the model training is continued on the increased training data pool. This procedure is repeated until no unlabeled data samples are left. To have a fair comparison between the methods in the following experiments, the learning loss module was activated in the first training cycle using

30% of the data. From there on, it was deactivated for all methods, except the learning loss ones. The experiments using the NuImages and Cityscapes dataset were trained for 30,000 steps per cycle, while the experiments using the A9 dataset were trained for only 3,000 steps per cycle. The current state of the model was saved and evaluated every 1,000th step for the NuImages and Cityscapes dataset and every 100th step for the A9 dataset. Several augmentations have been applied during training as described in [Sal21]. These augmentations are randomly cropping, flipping along the horizontal axis, changing the gamma curve, contrast, brightness, and hue as well as changing the white balance of the image. Furthermore, Gaussian blur and noise are added to the image. Non-maximum suppression is applied to the set of detections and if not stated otherwise, only the detected objects with a confidence score higher than 60% were kept for the evaluation and next cycles sample selection. All experiments were trained and evaluated using Tesla V100 GPUs with 32GB of memory. For the experiments with a batch size greater than 1, multiple GPUs were used with a batch size of 2 for each GPU. The code was implemented in Python 3.8 with TensorFlow 2.2, Cuda 10.1, cudnn 7, Boost 1.60 and OpenCV 3.4.16. The code is available online [Fri22].

### 4.1.4 Evaluation Metrics

To compare the performance of the investigated methods, it is important to look at various metrics. The two obvious metrics are the performance metrics of the respective tasks, object detection and semantic segmentation. But in active learning, pure performance is not the only point of interest. Factors such as the required amount of annotations and thus the resulting labelling costs are of major importance as well. In the following subsection, the metrics that have been used throughout this thesis are described in more detail.

#### Object Detection

The most prominent metric to evaluate the performance on object detection tasks is the mean Average Precision [Fen+21]. It is based on the precision metric which is defined as

$$P_c = \frac{TP_c}{TP_c + FP_c} \tag{4.3}$$

where $TP_c$ is the number of true positives and $FP_c$ is the number false positives of a class $c$. From here the mean average precision can be calculated by averaging the precision of all classes $C$ that are evaluated.

$$mAP = \frac{1}{|C|} \sum_{c \in C} P_c \tag{4.4}$$

In this thesis, the mAP is always evaluated with an Intersection over Union (IoU) threshold of 50%. The IoU measures how well a detected bounding box matches the ground-truth box with respect to location and size [Mon21, Chap. 6.1.1].

$$IoU(A,B) = \frac{A \cap B}{A \cup B} \tag{4.5}$$

#### Semantic Segmentation

For the evaluation of the semantic segmentation the Jaccard Index is used. It is also known as the PASCAL VOC IoU [Eve+15]. In this thesis the mean over all classes $C$ is taken.

$$mIU = \frac{1}{|C|} \sum_{c \in C} \frac{TP_c}{TP_c + FP_c + FN_c} \tag{4.6}$$

#### Data Correlation

A metric that is useful to compare different selection strategies against each other is the data correlation. This measures the overlap of two given sets of data samples. If the two sets are identical the correlation is at its maximum, if there is not a single sample that exists in both data sets, the correlation is at its lowest value which is zero. Two calculate the correlation, a simple approach is used. The samples that are within both given sets are counted and this number is divided by the total amount of samples in the datasets.

#### Annotation Cost Estimation

The main goal of active learning is the reduction of labelling costs. Therefore, it is important to estimate the required labelling cost of an unlabeled sample. Cai et al. [Cai+21] argue that not all samples have an equal annotation cost as the objects differ in their size and shape. They introduce a click-based annotation cost estimation, which reflects a more realistic estimation. However, many labelling companies are paid by the number of objects that were annotated, regardless of their size. Therefore, the amount of objects within an image is used as the annotation cost. To estimate the costs of a subset containing the images selected by a selection strategy, the ground truth bounding boxes of each sample in that subset are counted and summed together. This method is easily implemented, fast to evaluate and gives a good enough orientation to compare the various selection strategies.

#### Cost Efficiency

Some selection strategies might result in a lower mAP or mIU value compared to others but require fewer annotations. In applications where costs are critical, a strategy with slightly poorer performance but a significantly lower cost might be preferable. Therefore, it is important to weigh the annotation cost against the achieved performance on the respective tasks. To measure the cost efficiency of the two tasks object detection and semantic segmentation, the annotation costs of the selected subset at a given cycle are divided by the achieved mAP and mIU for the object detection and semantic segmentation, respectively.

## 4.2 Baseline

To evaluate the effectiveness of the researched active learning methods, two baselines were established. The first one serves as a comparison against the traditional training procedure where it is assumed that the labels for all data samples are available and no active selection of data samples is done. During the training 100% of the data is used from the very first training step on already. This baseline will be referenced as $Full$ and was trained for 300,000 training steps on the NuImages and Cityscapes datasets, respectively. For the experiments on the A9 dataset, the $Full$ method was trained for 30,000 training steps. The second baseline will be referenced as $Random$ and uses the active learning procedure described in Section 4.1.3. In each cycle, a randomly selected subset is added to the labelled data pool. The cycle iterations were set to 10% of the $Full$ training steps. Therefore, the models were trained for 30,000 iterations each cycle on the NuImages and Cityscapes dataset and 3,000 on the A9 dataset, respectively. The weights of the best checkpoint of the first random training cycle are used as the starting point for all the following methods. The best checkpoint has been selected based on the object detection mAP on the validation set. The reason was that all runs had similar performance for the semantic segmentation task, but the results on the object detection differed a lot. It is not to be expected that an active learning method outperforms the full data

training, however, each method aims to outperform random selection concerning the trained tasks, but also with respect to the annotation costs. Even though a method might not have a higher mAP than random selection, a lower annotation cost might still be desirable for some applications. It must also be mentioned that for the $Full$ training a decreasing learning rate schedule was used as proposed in [Sal19]. For the active learning method experiments, on the other hand, the same schedule caused a lower accuracy. This can be explained by the fact, that new data is added to the dataset and then the decreased learning rate might be too low. Therefore, this schedule was deactivated for the active learning methods and the learning rate was kept the same throughout all training steps. However, an adapted learning rate schedule could result in an increased performance towards the end. Research on this remains for future work.

## 4.3 Aggregation Methods - Sum, Maximum and Average

The methods described in Section 3.1.1 and 3.1.2 calculate a score based on the detected objects. But since the decision of the sample selection is done on an image and not on a bounding box level, one must find a way to aggregate the scores of the detected objects into one single image score. In the first experiment of this thesis, two sample selection methods with three different aggregations each are investigated. Possible aggregations are the sum, where all box scores are simply added up, the average, where the sum of all boxes is divided by the number of boxes, and finally the maximum, where only the box with the highest score determines the total score of an image. The first method is using the least confident bounding boxes, as described in Section 3.1.1. The results of this method using the different aggregations will be referenced as $Conf_{sum}, Conf_{avg}, Conf_{max}$. The second sample selection method is described in Section 3.1.2 and uses the difference between the most likely class prediction and the second most likely class prediction. In the results section these methods will be referenced as $1vs2_{sum}, 1vs2_{avg}, 1vs2_{max}$. The results of the evaluation of all previously mentioned methods are presented in Section 5.1.

## 4.4 Inconsistency Methods

Instead of relying on the predictions of a single image to calculate the score, it is also possible to run the inference twice. Once for the original image and then again for the same image, but horizontally flipped. The overall image score is then calculated by measuring the inconsistency between the two predictions. This approach was proposed by Elezi et al. and described in more detail in Section 3.1.3. The method can be divided into two sub approaches, one is using all detected bounding boxes and the other takes only bounding boxes into account that surpass a certain confidence threshold $c_i$, which was set to 50%. In the results, those two methods will be referenced as $Incon_{od}$ and $InconConf_{od}$. The idea of using augmentation to calculate the inconsistency can not only be applied to object detection, but also to semantic segmentation. In Section 3.1.4 two approaches are proposed. One is just counting the pixels where the two predictions do not match, and the other one is more sophisticated and uses the KL-divergence to calculate the inconsistency. The two will be referenced as $Incon_{seg}$ and $Incon_{seg+KL}$, respectively. The results of the evaluation can be found in Section 5.2.

## 4.5 Loss Prediction Methods

A simple yet very efficient approach is the loss prediction module method, which is described in detail in Section 3.1.5. It does not rely on any task related predictions but adds an additional module to the network architecture which then learns to predict the loss that a sample will produce if it would be used during training. The samples with the lowest estimated loss are chosen for the next cycle. Selecting the samples with the highest estimated loss has been tried as well, but had worse accuracy in both object detection and semantic segmentation. The general approach uses three intermediate layers of the backbone and three intermediate layers of the respective task head as input for the loss prediction module. The backbone layers had the channel sizes 128, 256, 512, and the head layers had 128, 64, and 32 channels, respectively. Global average pooling is applied to each input layer, resizing the layers to an equivalent size. A densely connected layer with 128 units follows the pooling and a ReLU activation is used. The resulting outputs of the separate input layers are then concatenated and passed to a densely connected layer with just one unit. Thus, a loss value can be predicted based on the intermediate values of the model. The loss prediction module can be used for three various approaches. First $Loss_{od}$, just the object detection loss. A single loss prediction module is added to the model architecture and is used to learn to predict the object detection loss which is then used to select the samples for the next training cycle. Second $Loss_{seg}$, the semantic segmentation loss. This approach works the same as the previous one, but it is learning to predict just the segmentation loss. The third approach, referenced as $Loss_{combined}$, adds two loss prediction modules and learns to predict both losses simultaneously. In this approach, the loss of both tasks is predicted and summed together. The samples with the lowest combined predicted loss are selected for the next training cycle. The two tasks are equally weighted during this selection process. In the publication of Yoo et al. [YK19], it's not explicitly mentioned whether they froze the weights of the model when training the loss prediction module. Not stopping the gradient for the model layers during the backpropagation resulted in reduced accuracy on both tasks. Therefore, the weights of all layers, including the input layers of the loss prediction module, are frozen while optimizing the layers of the loss prediction module. This way, the training of the loss prediction module does not affect the tasks themselves. Different weights of the loss prediction module loss have been tried. At first, the weight was set to $\delta_{pred} = 10.0$, but $\delta_{pred} = 1.0$ showed to be the better setting. The results of the three approached $Loss_{od}, Loss_{seg}$ and $Loss_{combined}$ are presented in Section 5.3.

## 4.6 BoxMask Method

In the first $BoxMask$ experiment, the effect of the bounding box confidence threshold on the overall performance was examined. In the prior experiments, this threshold was always set to 60%. This way, only bounding boxes with relatively high confidence have been used to calculate the samples score. However, if this threshold is set that high, the amount of detected bounding boxes is quite low, especially in the early training cycles when the model is not so well trained yet. To examine whether this intuition is true, two experiments have been evaluated. Their settings were identical, except for the bounding box threshold, which was set to 60% in $BoxMask_{60}$ and to 30% in $BoxMask$. The results show, that there is not a huge difference in the achieved task performance. The smaller box threshold has a minimal better performance on the object detection. For semantic segmentation, there is no noticeable difference. However, the reduced box threshold resulted in lower annotation costs. Therefore, the box threshold was set to 30% in the remaining BoxMask experiments.

## 4.7 Combined Methods

The training of a model using active learning must not be limited to just one selection strategy. There are endless possibilities to combine the various selection strategies. The methods can be combined based on their estimated cost, the correlation between their selected subsets, or based on their performance on the respective tasks. But not only the strategies can vary. The most basic combination method, that even does not require a change in the model architecture is the intermediate checkpoint selection. The effect of different combinations of the so far presented methods as well as the checkpoint selection is presented in the following section.

### 4.7.1 Checkpoint Selection

When training a multi-task model, one has multiple metrics to determine which method generates the best results on the respective tasks. These metrics are then used to evaluated the learned parameters of the model at a given checkpoint. During the full pipeline of active learning many intermediate checkpoints are generated and at the end of each cycle a decision must be made, which checkpoint is the best to continue the training from. In object detection this decision is mostly based on the mean average precision and in semantic segmentation the mean intersection over union is the most frequently used metric for this decision. There is not a one-and-only choice of metric to choose in a multi-task setting. One could use the mAP, which could result in worse performance on the segmentation task, or one could use the mIU, but this would lead to a not optimal performance on the object detection task. This experiment evaluates the effect of the checkpoint selection using the $Incon_{seg}$ method described in Section 3.1.4. In $Incon_{seg-od}$ the mAP is used to determine the best checkpoint to continue from. In $Incon_{seg-seg}$ the mIU is used and in $Incon_{seg-both}$† both mAP and mIU influence the checkpoint selection. To find the checkpoint that performs best on both tasks, the checkpoints are ranked by the respective metric for each task, with the best checkpoint having the highest rank. In the next step, the ranks of each checkpoint are summed and the checkpoint with the overall highest sum of ranks is used as the starting point of the next training cycle. The effect of the combined checkpoint has also been investigated using the $Conf_{max}$ method, as the results showed it to be the best performing baseline method. The training settings were kept as they were described in Section 4.3. This experiment will be referenced as $Conf_{max-both}$†. The results of the two experiments are shown in Section 5.5.1.

### 4.7.2 Half Split Alternation

Another idea was to combine a method that performs well in the earlier phase of the training cycles with a method that performs better in the later phase of the training. As later shown in the results section the loss prediction module method $Loss_{od}$ is a good performing in the early training stages. The BoxMask method $BoxMaskEl_{BN}$ on the other hand is well performing at later stages. Thus, this two have been combined, using the $Loss_{od}$ selection strategy for the first 60% of the data and from there on $BoxMaskEl_{BN}$ was used to select the remaining samples for the labeling. This experiment will be referenced as $Loss + BoxMask$. As the $1vs2_{avg}$ method has a lower cost efficiency compared to the $BoxMaskEl_{BN}$ and has a high accuracy in the later training phases as well, the half split has also been applied to the combination of $Loss_{od}$ with $1vs2_{avg}$. This experiment will be referenced as $Loss + 1vs2$. The results of the two half split experiments are compared against their single trained components and presented in Section 5.5.2.

### 4.7.3 Alternation of Classification and Localization Optimizing Methods

The investigated methods can be classified into two categories, depending on what the method is trying to optimize. Some methods are mainly focusing on the predicted classes and ignore the localization of an object. Other methods do it the vice versa. In this experiment, a method of each category is combined such that each training cycle optimizes either the class prediction or the localization in an alternating fashion. The method choice fell on $1vs2_{max}$ as the class optimizer method and $BoxMaskEl_{BN}$ as the localization optimization. Even though $LeastConf_{max}$ has higher accuracy on both tasks, $1vs2_{max}$ has a better cost efficiency, which is why this method was selected. If the two methods $BoxMaskEl_{BN}$ and $Incon_{seg}$ are compared fairly by using the same checkpoint selection strategy, $BoxMaskEl_{BN}$ has the better overall performance. The training of this experiment uses $BoxMaskEl_{BN}$ as the sample selection method for the first active learning cycle, then continues with $1vs2_{max}$ for the second cycle. The two methods are then alternated until no data remains. This experiment will be referenced as $ClassLocalization$ and its results are presented in Section 5.5.3.

### 4.7.4 Alternation of Low Correlating Methods

In the next approach to alternate two methods, the idea was to combine methods that select sample subsets that have a low correlation to each other. Based on the experiments done with the methods described earlier, the correlation between the several methods could be measured easily as described in Section 4.1.4. The correlation measurements showed that the $BoxMaskEl_{BN}$ method has the lowest correlation with the $Conf_{max}$ method compared to the other combinations for most of the cycles. Therefore, in the experiment $BoxMaskEl_{BN} - Conf\dagger$ the first training cycle used $BoxMaskEl_{BN}$ as the selection strategy and the second cycle then used $Conf_{max}$. The two methods are alternated at each cycle till no data is left. Based on the findings in Section 5.5.1, the combined checkpoint selection was applied in this experiment. This means that a fair comparison with the other experiments is no longer possible, but it does give an indication of the results that a combination of all the findings in this thesis is capable of. The results of this experiment on the NuImages dataset are presented in Section 5.5.4.

### 4.7.5 Other Alternating Approaches

During this thesis other method combinations have been evaluated as well. These may not have unique characteristics that make them a useful combination. Based on the results obtained and the costs involved, it was nevertheless interesting to initiate the following experiments. All three variations of the loss prediction module method showed impressive results on both or either one of the two tasks. As shown in Section 5.3, their combination in a non-alternating fashion $Loss_{comb}$ offers a good trade-off between the two single task focused variations. In $Loss_{od} - Loss_{seg}$ the two single-task focused variations are alternated, instead of combining the two methods. In this experiment the intermediate checkpoints were selected based on the object detection performance. In $Loss_{od} - Loss_{seg}*$ the same methods have been alternated, but in addition the checkpoint selection has been alternated as well. In the cycles were the samples were selected using $Loss_{od}$, the checkpoint was selected based on the object detection accuracy. In the cycles where $Loss_{seg}$ was used to choose the next batch of unlabelled data, the intermediate checkpoint was selected based on the segmentation accuracy. So far only two different methods have been combined with each other. In the experiment

referenced as $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$, the three methods $Loss_{od}$, $BoxMaskEl_{BN}$ and $Loss_{seg}$ were alternated in this order. After the cycles that used $Loss_{od}$ and $BoxMaskEl_{BN}$ as the selection strategy, the checkpoint was selected based on the object detection results. In the cycles of $Loss_{seg}$, the semantic segmentation results were used to determine the best checkpoint. The same experiment setting was applied to $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$. The only difference is the changed order of the methods, now starting with $BoxMaskEl_{BN}$, followed by $Loss_{od}$ and $Loss_{seg}$. Out of all non-alternating methods $Conf_{max}$ has the best performance on both object detection and semantic segmentation. This comes at the drawback, that this method causes one of the highest annotation costs. Therefore, it was worth to investigate the combination of this well performing, but expensive method with another well performing, but cheaper method. The intention was to keep the task accuracy high, but lower the overall annotation costs. The two methods under consideration were $Loss_{comb}$ and $BoxMaskEl_{BN}$. Both methods have comparable performance on the two tasks, but $BoxMaskEl_{BN}$ has slightly lower cost. Therefore, method $BoxMaskEl_{BN}$ was alternated with $Conf_{max}$ in the experiment called $BoxMaskEl_{BN} - Conf_{max}$†, which was described in Section 4.7.4. A combined checkpoint selection was applied to this experiment as well. The results achieved on the NuImages dataset using this alteration of methods are presented and compared in Section 5.6.1.

## 4.8 Method Comparison

Some of the proposed methods have also been evaluated on two other datasets. These are the A9 and Cityscapes datasets. Both of them are described in detail in Section 4.1.2. The settings for each method were the same as on the NuImages dataset. All experiments were run twice to avoid bias due to possible favourable initialisation. The reported mAP and mIU is the average of these two runs. The results of the different methods on all datasets are presented in Section 5.6.1.

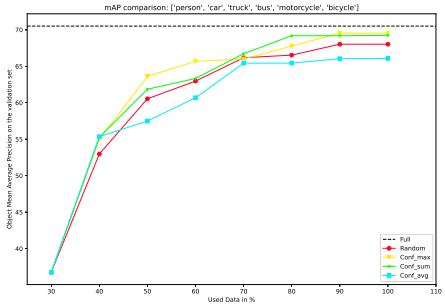
# Chapter 5

# Results

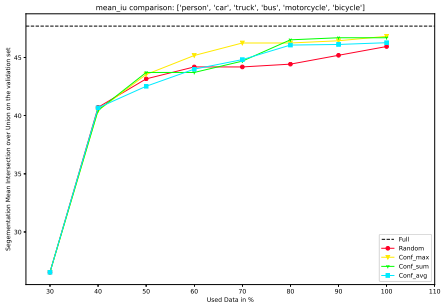
The order of the presented results is as it is in the Sections 4.2-4.6. For each experiment, the results of the object detection and semantic segmentation will be presented as graph visualization and more detailed in the form of Tables. In addition to that, the cost-efficiency graphs are given as well. In Section 5.6 the best methods are compared against each other on the three datasets NuImages, A9 and Cityscapes. This Section is divided into a quantitative evaluation in Section 5.6.1, which presents the mAP@0.5 and mIU achieved by the evaluated methods. The methods are also qualitatively evaluated in the following Section 5.6.2.

## 5.1 Aggregation Methods

In the first experiment, three different aggregation methods have been compared against each other on two different methods. The aggregation methods are the sum, maximum and average of all box confidence scores. The evaluated methods are Least Confidence and 1-vs-2 margin Sampling, which are described in Section 3.1.1 and 3.1.2. Brust et al. state in their publication [BKD19] that sum is the best aggregation method. They also mention that the superior performance of the sum aggregation method comes at higher labelling costs, as this method tends to select the samples containing a lot of objects. This finding can be confirmed by the results obtained using the least confidence approach. As can be seen in Figure 5.1, the average aggregation has the worst performance on both, object detection and semantic segmentation. The results of sum and max, on the other hand, are close. A larger difference between the two methods is visible when looking at the cost efficiency shown in Figure 5.2. Here it becomes clear that the max aggregation method achieves the same results with fewer required annotations, thus making it the preferred method for this application and dataset. For the 1-vs-2 margin Sampling approach, however, the obtained results allow other conclusions to be drawn. As can be seen in Figure 5.3, using average as the aggregation method results in the highest performance for object detection in the later training cycles. In earlier training cycles the max aggregation has the best results for object detection. On the segmentation task, all aggregation methods are achieving close performance, with again max being best in early cycles and the sum being best in the last cycle. Even though the differences in cost efficiency are significantly smaller than with the least confidence approaches, in Figure 5.4 it can be seen that maximum is the most efficient aggregation method for 1-vs-2 margin sampling as well. The general deviation of the results between the two methods of least confidence and 1-vs-2 margin sampling could be explained by the fact that not enough experiments could be conducted per method due to the limited time. Therefore, the presented results may contain a certain statistical variance. The exact results of the two approaches and their aggregation variants are given in Table 5.1 for the

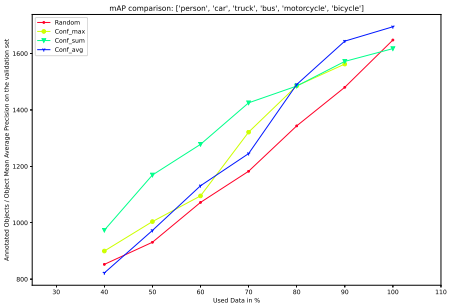



**(a)** Object Detection mAP@0.5

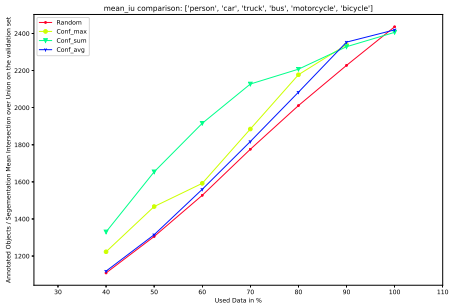



**(b)** Semantic Segmentation mIU

**Figure 5.1:** A result comparison of the least confidence approaches on the NuImages validation dataset. The plotted values are the average of multiple runs.

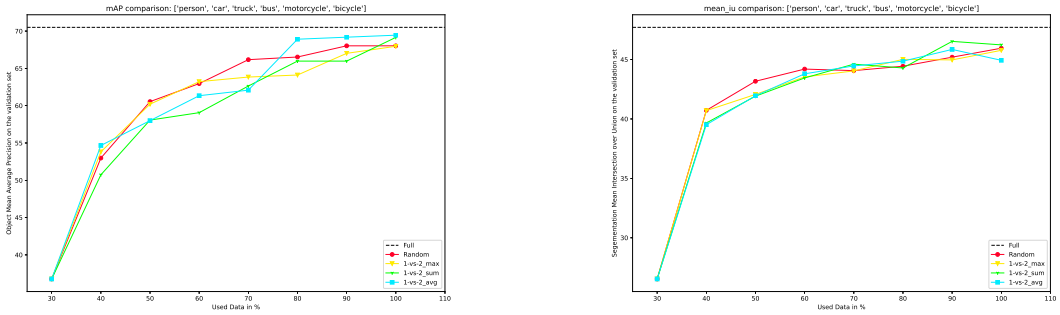



**(a)** Object Detection



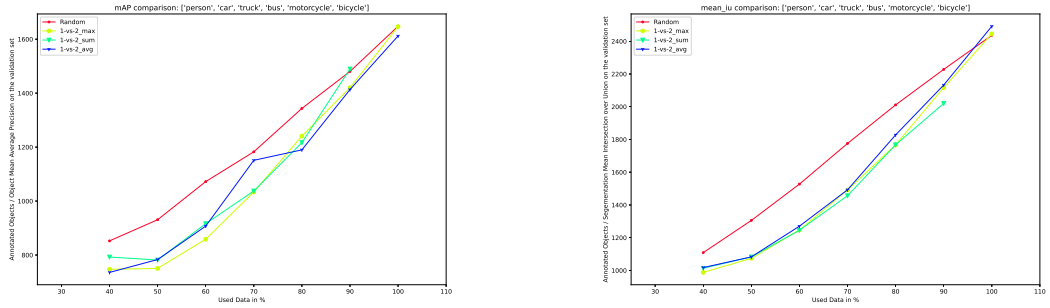

**(b)** Semantic Segmentation

**Figure 5.2:** A cost efficiency comparison of the least confidence approaches on the NuImages validation dataset. Lower values indicate a better efficiency.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.3:** A result comparison of the 1-vs-2 margin sampling approaches on the NuImages validation dataset. The plotted values are the average of multiple runs.




**(a)** Object Detection **(b)** Semantic Segmentation

**Figure 5.4:** A cost efficiency comparison of the 1-vs-2 margin sampling approaches on the NuImages validation dataset. Lower values indicate a better efficiency.

object detection and in Table 5.2 for the semantic segmentation. The two active learning approaches consistently outperform random selection and reach up to 90% of the full data performance on both tasks with just 50% of the data samples. This corresponds to just 46% required object annotations and thus annotation costs compared to the full data training. The 1-vs-2 margin approach requires even fewer annotations. With just 39% annotated objects of the full training, it achieves 85.39% of the object detection mAP@0.5 and 88.16% of the semantic segmentation mIU. As the maximum aggregation method gives the best trade-off between performance on the two tasks, as well as the required annotation effort, these two will be used in the following experiments. It must be mentioned that these aggregation variants represent the general methodological performance of least confidence and 1-vs-2 margin sampling, but might not be the best choice for other applications.

## 5.2 Inconsistency Methods

The methods of this work that use inconsistency as a selection criterion can be divided into two approaches. One approach focuses on object detection and was described in Section 3.1.3. The other focuses on semantic segmentation and was discussed in Section 3.1.4. Both approaches are similar in their basic functionality. Both tasks are predicted for the origi-

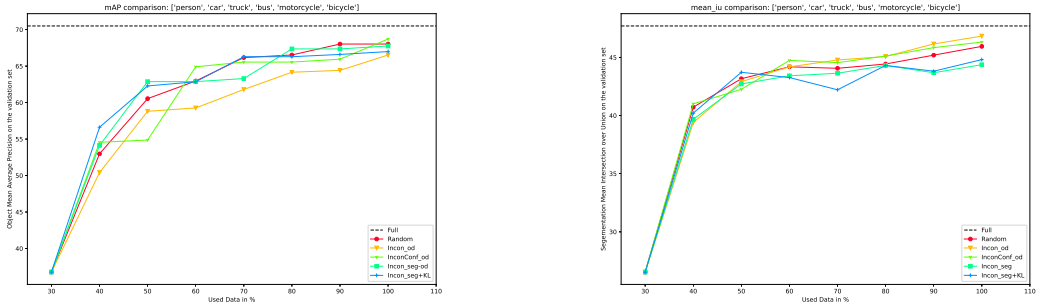
| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | 66.17 | 66.53 | 68.03 | 67.94 |
| $Conf_{max}$ | 36.77 | 54.98 | **63.61** | **65.68** | 65.99 | 67.78 | **69.54** | **69.54** |
| $Conf_{sum}$ | 36.77 | **55.28** | 61.85 | 65.27 | **66.72** | **69.19** | 69.19 | 69.23 |
| $Conf_{avg}$ | 36.77 | 55.35 | 57.5 | 60.68 | 65.43 | 65.43 | 66.04 | 66.07 |
| $1vs2_{max}$ | 36.77 | 53.81 | **60.2** | **63.22** | **63.55** | 64.11 | 67.02 | 67.96 |
| $1vs2_{sum}$ | 36.77 | 50.73 | 58.07 | 59.06 | 62.63 | 64.36 | 64.36 | 69.16 |
| $1vs2_{avg}$ | 36.77 | **54.66** | 58.0 | 61.34 | 61.34 | **68.91** | **69.19** | **69.45** |

**Table 5.1:** The object detection mAP@0.5 results of least confidence and 1-vs-2 margin sampling with the various aggregation methods on the NuImages validation dataset at each cycle.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | **40.72** | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Conf_{max}$ | 26.53 | 40.45 | 43.55 | **45.18** | **46.26** | 46.26 | 46.45 | **46.83** |
| $Conf_{sum}$ | 26.53 | 40.46 | **43.72** | 43.98 | 44.69 | **46.53** | **46.71** | 46.71 |
| $Conf_{avg}$ | 26.53 | **40.68** | 42.53 | 44.01 | 44.83 | 46.08 | 46.13 | 46.28 |
| $1vs2_{max}$ | 26.53 | **40.7** | **42.06** | 43.54 | 44.04 | **45.01** | 45.01 | 45.77 |
| $1vs2_{sum}$ | 26.53 | 39.66 | 41.92 | 43.45 | **44.61** | 44.61 | **46.53** | **46.53** |
| $1vs2_{avg}$ | 26.53 | 39.51 | 41.95 | **43.8** | 44.46 | 44.86 | 45.85 | 45.85 |

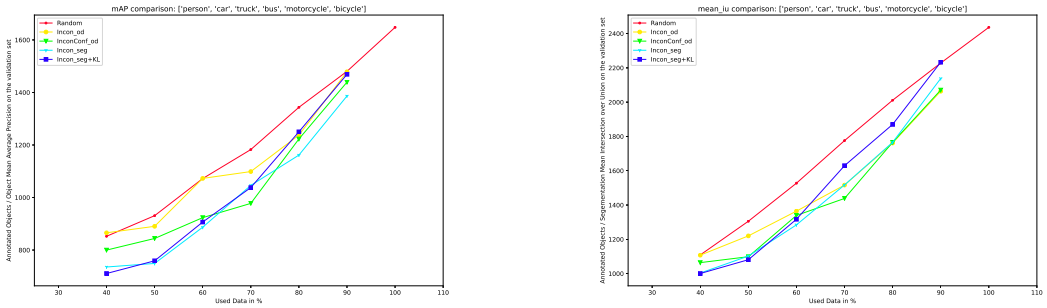
**Table 5.2:** The average semantic segmentation mIU results of least confidence and 1-vs-2 margin sampling with the various aggregation methods on the NuImages validation dataset at each cycle.

nal image and the same image but horizontally flipped. The results of both predictions are then compared with each other and their inconsistency is used to determine the next cycle's data selection. The results of the inconsistency approaches are visualized in Figure 5.5 and exact values are given in Table 5.3 for the object detection task and in Table 5.4 for the semantic segmentation task. One can see that the confidence-based approach $InconConf_{od}$ outperforms $Icon_{od}$ in most of the cycles, implying that taking confidence into account has a beneficial impact on the end result. This becomes more clear if one looks at the cost-efficiency of both methods in Figure 5.6. Here, the confidence method has a better cost efficiency for both tasks. The performance of the $Incon_{seg}$ method and the more sophisticated $Incon_{seg+KL}$ method is more or less on par. However, $Incon_{seg}$ is simpler to compute and has a better cost efficiency at later cycles, which makes it the preferable option. An interesting observation is that the segmentation focused methods $Incon_{seg}$ and $Incon_{seg+KL}$ have better results on the object detection task, while the object detection focused methods $Incon_{od}$ and $InconConf_{od}$ have better results on the semantic segmentation task. This indicates that one task can benefit from the other task. Nonetheless, the results are quite close together and could potentially be affected by statistical variance. However, inconsistency-based methods are generally better than random selection and undeniably have better cost-efficiency. However, in terms of performance, they are subject to the least confidence approach and in terms of cost-efficiency, they are subject to the 1-vs-2 margin sampling approach. The inconsistency methods are thus a trade-off between performance and cost-efficiency. Due to the simpler score computation, the $Incon_{seg}$ method has an inferior computation time, reducing the overall training time. As the performance of this method is close to or even better than the other methods, $Incon_{seg}$ will be used in the following experiments and represents the inconsistency-based methods.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.5:** A result comparison of the inconsistency approaches on the NuImages validation dataset. The plotted values are the average of multiple runs.




**(a)** Object Detection **(b)** Semantic Segmentation

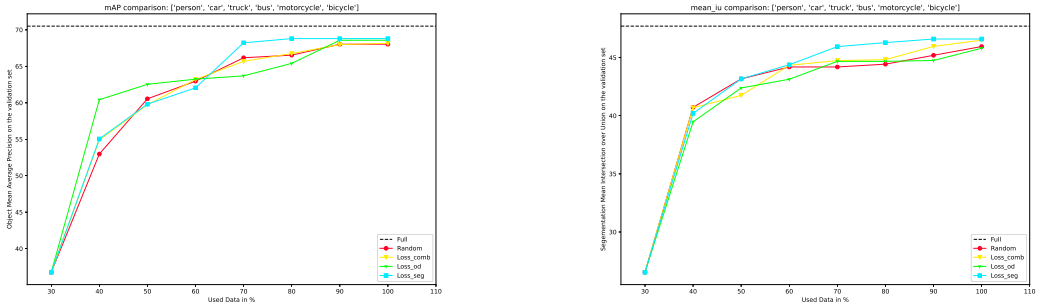
**Figure 5.6:** A cost efficiency comparison of the inconsistency approaches on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | 66.17 | 66.53 | **68.03** | 67.94 |
| $Incon_{od}$ | 36.77 | 50.41 | 58.81 | 58.81 | 61.79 | 64.17 | 64.42 | 66.53 |
| $InconConf_{od}$ | 36.77 | 54.56 | 55.01 | **64.91** | 65.55 | 65.55 | 65.95 | **68.71** |
| $Incon_{seg}$ | 36.77 | 54.14 | **62.88** | 62.88 | 63.3 | **67.36** | 67.36 | 67.76 |
| $Incon_{seg+KL}$ | 36.77 | **56.63** | 62.28 | 62.86 | **66.29** | 66.29 | 66.6 | 66.99 |

**Table 5.3:** The average object detection mAP@0.5 results for the various inconsistency methods on the NuImages validation dataset at each cycle based on multiple runs.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | 40.72 | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Incon_{od}$ | 26.53 | 39.4 | 42.92 | 44.19 | **44.78** | 45.08 | **46.16** | **46.84** |
| $InconConf_{od}$ | 26.53 | **41.01** | 42.26 | **44.75** | 44.75 | **45.12** | 45.84 | 46.32 |
| $Incon_{seg}$ | 26.53 | 39.68 | 42.72 | 43.42 | 43.64 | 44.31 | 44.31 | 44.38 |
| $Incon_{seg+KL}$ | 26.53 | 40.22 | **43.72** | 43.72 | 43.72 | 44.31 | 44.31 | 44.81 |

**Table 5.4:** The average semantic segmentation mIU results for the various inconsistency methods on the NuImages validation dataset at each cycle.

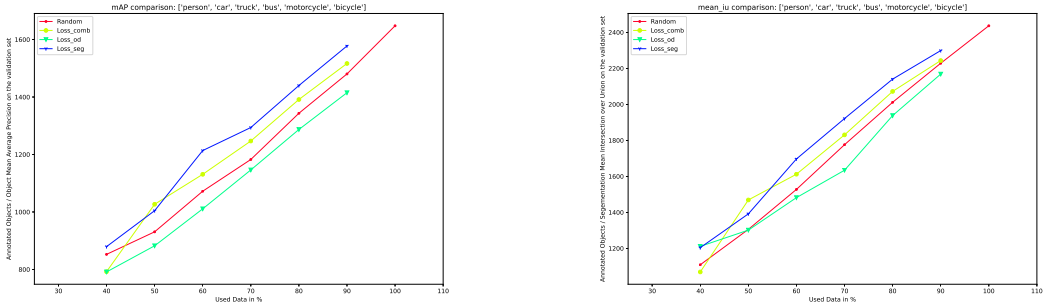



**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.7:** A result comparison of the loss prediction module approaches on the NuImages validation dataset. The plotted values are the average of multiple runs.

## 5.3 Loss Prediction Module

In Section 3.1.5, three different methods were explained how a loss prediction module can be built and used. The results of these methods are presented below. The first approach, $Loss_{od}$, predicts the loss for only the object detection and uses it to select the next training samples. The second approach, $Loss_{seg}$, works just like $Loss_{od}$ but predicts the loss for the semantic segmentation task instead. In the early training phase, the intuition that the task-focused method stand out from the other task-focused method in the respective focused task could be confirmed. But as can be seen in Figure 5.7 and in Table 5.5, this changes as soon as more than 60% of the data is used. In the later training phase, when more data is available, $Loss_{seg}$ outperforms $Loss_{od}$ in both tasks. Towards the end, the two methods converge again in their results on object detection, but $Loss_{seg}$ is still ahead on the semantic segmentation task as can be seen in Table 5.6. This could indicate, that a sample selection strategy that focuses solely on the semantic segmentation task could have a more beneficial impact on the object detection performance than a strategy that just focuses only on object detection. However, the increased performance of the $Loss_{seg}$ method could be explained by the fact, that this method selects samples that contain more objects. In Figure 5.8, one can see that the segmentation focused loss prediction module approach has the worst cost efficiency. At 70% data, the by $Loss_{seg}$ selected subset of samples contains 99,063 objects, which corresponds to 88.46% of the objects contained in the full training dataset. On the other side, the from $Loss_{od}$ selected subset at the same cycle contains only 84,165 objects, which is just 75.16% of the full training dataset objects. The combined variant $Loss_{combined}$ behaves as expected and offers a good trade-off performance on both tasks as well as in regards to the cost-efficiency. At 70% data, the by $Loss_{combined}$ selected sample subset contains 93,139 objects, which is however slightly higher compared to random selection which has 89,375 objects in that respective subset. Nonetheless, the combined loss prediction module achieves the best overall performance at 100% data on both tasks, with +0.21% mAP@0.5 and +0.55% mIU, compared to random selection. The $Loss_{combined}$ method has therefore been selected for the remaining experiments.




**(a)** Object Detection **(b)** Semantic Segmentation

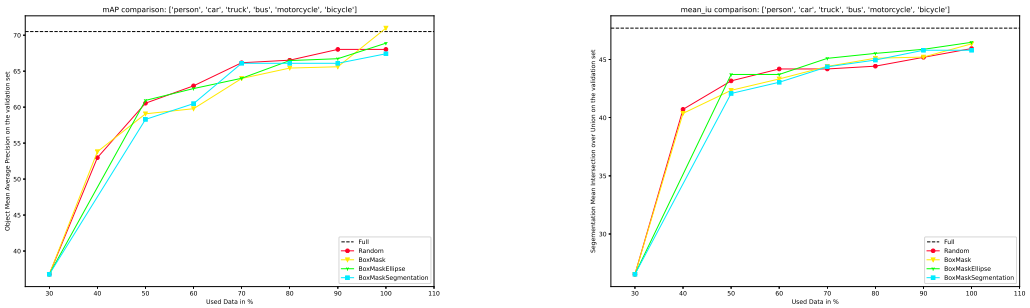
**Figure 5.8:** A cost efficiency comparison of the loss prediction module approaches on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | 66.17 | 66.53 | 68.03 | 67.94 |
| $Loss_{od}$ | 36.77 | **60.42** | **62.53** | **63.24** | 63.69 | 65.4 | 68.57 | 68.57 |
| $Loss_{seg}$ | 36.77 | 55.04 | 59.81 | 62.05 | **68.22** | **68.8** | **68.8** | **68.8** |
| $Loss_{combined}$ | 36.77 | 54.91 | 59.73 | 63.19 | 65.68 | 66.75 | 67.98 | 68.15 |

**Table 5.5:** The average object detection mAP@0.5 results for the various loss prediction module methods on the NuImages validation dataset at each cycle.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | 40.72 | **43.17** | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Loss_{od}$ | 26.53 | 39.48 | 42.39 | 43.14 | 44.68 | 44.68 | 44.76 | 45.81 |
| $Loss_{seg}$ | 26.53 | 40.2 | **43.17** | **44.38** | **45.94** | **46.29** | **46.6** | **46.6** |
| $Loss_{combined}$ | 26.53 | **40.69** | 41.75 | 44.34 | 44.74 | 44.82 | 45.96 | 46.5 |

**Table 5.6:** The average semantic segmentation mIU results for the various loss prediction module methods on the NuImages validation dataset at each cycle.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.9:** A result comparison of the different mask generation approaches applied to the $BoxMask$ method on the NuImages validation dataset. The plotted values are the average of multiple runs.

## 5.4 BoxMask Methods

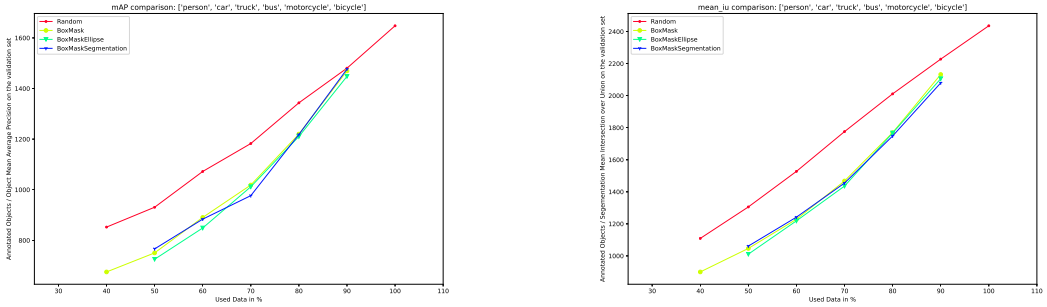
The results of the BoxMask methods are split into two parts. The first one will present the results achieved by the different mask generation approaches described in Section 3.1.6. The second part will then focus on the different normalization methods applied to the simple $BoxMask$ approach, which are described in more detail in Section 3.1.6 as well.

### 5.4.1 BoxMask - Mask Generation Methods

In the first $BoxMask$ experiment the effect of the various mask generation approaches was investigated. The approaches are described in detail in Section 3.1.6. As can be seen in the Figure 5.9 and Table 5.7, the $BoxMask$ approaches are not outperforming $Random$ selection on the object detection task for most of the training cycles, no matter what method is used to generate the mask. However, the methods do perform on-par during most of the cycles. On the segmentation task on the other hand, the $BoxMask$ achieves better mIU results for most of the time. In particular the $BoxMaskEllipse$ approach shows to be a promising method. If then the cost efficiency is considered, all $BoxMask$ approaches are more efficient than $Random$ selection. This is visualized in Figure 5.10. At 60% used data samples, the $BoxMaskEllipse$ method required 12% less annotations as $Random$ selection and reached a similar accuracy. At this training stage the $BoxMaskEllipse$ approach achieved 88.76% of the full data object detection accuracy and even 91.64% of its accuracy on the semantic segmentation task. The $BoxMask$ approach is the only method, that outperformed the full data training on the object detection task. The average mAP@0.5 of two runs is 0.48% better than the full data training. On the segmentation task however, no BoxMask approach reaches the accuracy of the full data training.

### 5.4.2 BoxMask - Normalization Methods

In many cases in the field of machine learning, normalization helps and achieves better results. In this case, however, this general assumption could not be confirmed. The simple $BoxMask$ approach uses no normalization and achieves the best results for both object detection and semantic segmentation. The normalization by the number of detected objects $BoxMask_{Norm}$ leads to significantly worse results, this is shown in Figure 5.11. Normalization




**(a)** Object Detection **(b)** Semantic Segmentation

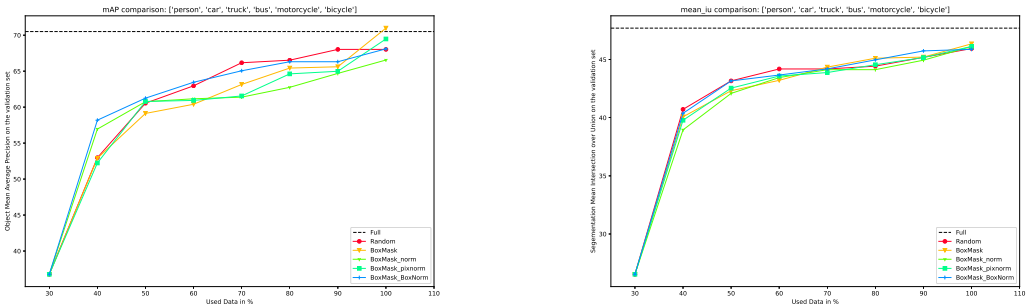
**Figure 5.10:** A cost efficiency comparison of the different mask generation approaches applied to the $BoxMask$ method on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | 70.5 |
| *Random* | 36.77 | 52.98 | 60.55 | **62.96** | 66.17 | 66.53 | **68.03** | 68.03 |
| *BoxMask* | 36.77 | **53.8** | 59.07 | 59.78 | 63.98 | 65.43 | 65.61 | **70.98** |
| *BoxMaskEllipse* | 36.77 | 51.98 | **60.93** | **62.58** | 64.01 | 66.47 | 66.73 | 68.89 |
| *BoxMaskSegmentation* | 36.77 | 52.82 | 58.29 | 60.49 | **66.9** | **66.9** | **66.9** | 67.42 |

**Table 5.7:** The average object detection mAP@0.5 results for the various mask generation approaches applied to the $BoxMask$ method on the NuImages validation dataset at each cycle.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | **40.72** | 43.17 | **44.2** | 44.2 | 44.44 | 45.2 | 45.95 |
| *BoxMask* | 26.53 | 40.37 | 42.36 | 43.34 | 44.41 | 45.12 | 45.22 | 46.36 |
| *BoxMaskEllipse* | 26.53 | **40.62** | **43.72** | **43.72** | **45.11** | **45.53** | **45.89** | **46.49** |
| *BoxMaskSegmentation* | 26.53 | 39.77 | 42.09 | 43.05 | 44.38 | 44.96 | 45.81 | 45.49 |

**Table 5.8:** The average semantic segmentation mIU results for the various mask generation approaches applied to the $BoxMask$ method on the NuImages validation dataset at each cycle.



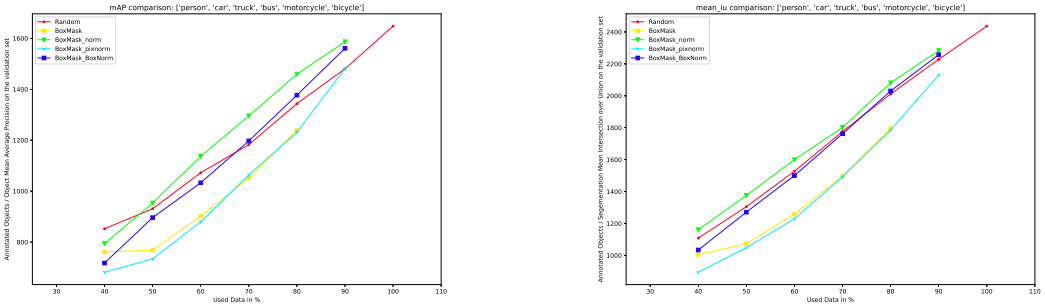

**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.11:** A result comparison of the different normalization approaches applied to the $BoxMask$ method on the NuImages validation dataset. The plotted values are the average of multiple runs.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| $Full$ | - | - | - | - | - | - | - | 70.5 |
| $Random$ | 36.77 | 52.98 | 60.55 | 62.96 | **66.17** | **66.53** | **68.03** | 68.03 |
| $BoxMask$ | 36.77 | 52.9 | 59.13 | 60.4 | 63.15 | 65.43 | 65.61 | **70.98** |
| $BoxMask_{Norm}$ | 36.77 | 56.95 | 60.76 | 61.16 | 61.39 | 62.76 | 64.69 | 66.55 |
| $BoxMask_{PixelNorm}$ | 36.77 | 52.24 | 60.79 | 60.93 | 61.55 | 64.63 | 64.99 | 69.48 |
| $BoxMask_{BN}$ | 36.77 | **58.19** | **61.25** | **63.46** | **65.05** | **66.31** | **66.17** | 68.1 |

**Table 5.9:** The average object detection mAP@0.5 results for the various normalization methods applied to the $BoxMask$ method on the NuImages validation dataset at each cycle.

by the number of pixels in the image $BoxMask_{PixelNorm}$ performs better than the previously mentioned normalization but is not better than the approach that does not use normalization. The exact results are given in the Tables 5.9 and 5.10. If one looks at the cost-efficiency in Figure 5.12, one can see that the $BoxMask_{Norm}$ normalization leads to higher annotation costs. The performance and annotation costs of the two methods $BoxMask$ and $BoxMask_{PixelNorm}$ are close together. This suggests that normalization by the number of pixels of the entire image has no effect. One possible explanation is that the number of pixels for each image is the same, so the score of an image is only scaled and not normalized. In $BoxMask_{BN}$ the number of unequal pixels within a bounding box are normalized by the size of the bounding box. This approach achieves the best accuracy on the object detection task for all but the last training cycle, if compared to the other normalization methods. It is also better performing than random selection for the first three cycles, reaching 90.01% of the full data training object detection mAP@0.5 accuracy with just 56.9% of the available data used. On the semantic segmentation task, the $BoxNorm$ (BN) approach is the best normalization for most of the training cycles. The highest accuracy towards the end is achieved by the $BoxMask$ experiment, which did not apply normalization at all. This approach also has, together with $PixelNorm$ the best cost efficiency, requiring only 48.61% annotations of all available annotations at the 60% cycle. When the great efficiency of the not normalized $BoxMask$ approach was finally confirmed, the remaining experiments with the $BoxNorm$ normalization were already started, as it looked the most promising at that time. Experiments with the pure $BoxMask$ approach remain for future research.




**(a)** Object Detection **(b)** Semantic Segmentation

**Figure 5.12:** A cost efficiency comparison of the different normalization approaches applied to the $BoxMask$ method on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| $Full$ | - | - | - | - | - | - | - | **47.71** |
| $Random$ | 26.53 | **40.72** | **43.17** | **44.2** | 44.2 | 44.44 | 45.2 | 45.95 |
| $BoxMask$ | 26.53 | 40.08 | 42.32 | 43.22 | **44.35** | **45.12** | 45.22 | **46.36** |
| $BoxMask_{Norm}$ | 26.53 | 38.95 | 42.09 | 43.45 | 44.14 | 44.0 | 44.95 | 46.13 |
| $BoxMask_{PixelNorm}$ | 26.53 | 39.78 | 42.55 | 43.59 | 43.89 | 44.56 | 45.18 | 46.13 |
| $BoxMask_{BN}$ | 26.53 | **40.38** | **43.17** | **43.67** | 44.2 | 44.98 | **45.75** | 45.88 |

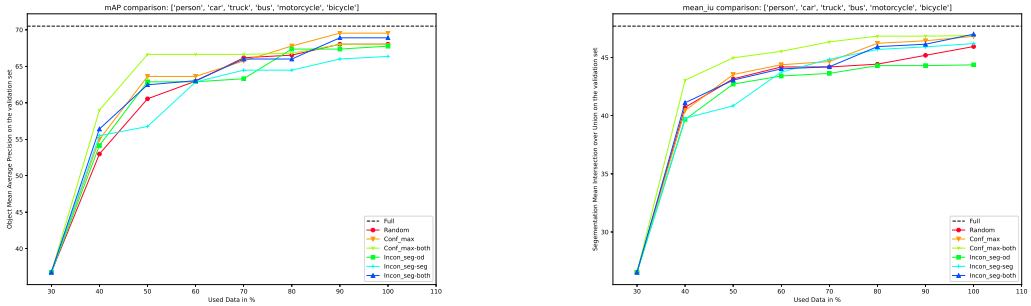
**Table 5.10:** The average semantic segmentation mIU results for the various normalization methods applied to the $BoxMask$ method on the NuImages validation dataset at each cycle.

## 5.5 Combined Methods

As already mention in Section 4.7, multiple combinations of methods are feasible. And not only the sample selection strategy can vary, the checkpoint selection has various options to choose from. This section will present the results of the experiments described in the subsections of Section 4.7. The order is the same, starting with the experiments on the checkpoint selection strategies in Section 5.5.1, followed by the results of the $HalfSplit$ approaches in Section 5.5.2. In the Sections 5.5.3 and 5.5.4 the alternating method experiments are presented. The results of the remaining alternating methods can be found in Section 5.6.1.

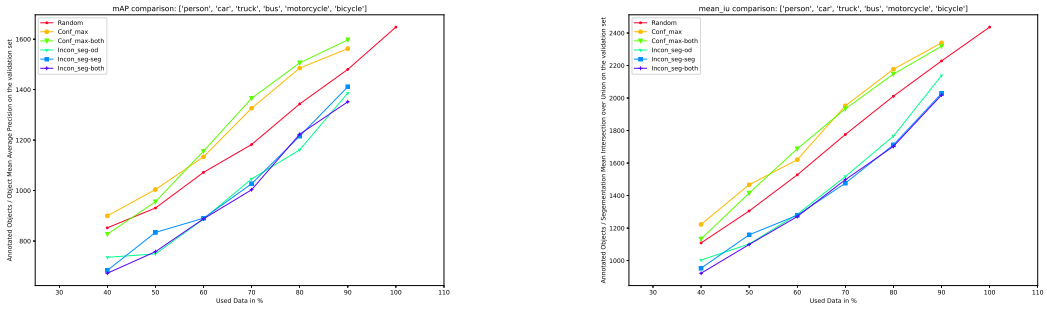
### 5.5.1 Checkpoint Selection

If the checkpoint selection based on the object detection is compared against the checkpoint selection based on the semantic segmentation, no clear conclusion can be made. Both selection strategies achieve results that are close together over all training steps as shown in Figure 5.13. In some cycles, the object detection based strategy outperforms the segmentation based one on both tasks, and in some cycles, it's the opposite way around. What strategy is performing best in which cycles is marked in Table 5.11 and 5.12 . However, the checkpoint selection that considers both tasks has a clear advantage on both tasks towards the end of the training if compared to the other two approaches. The $Incon_{seg}$ method with a combined checkpoint selection outperforms not only the two single-task checkpoint selection strategies, but also random selection by approximately 1% on both the object detection and semantic segmentation at 100% used training data. This might not seem like a huge improvement, but




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.13:** A result comparison of the different checkpoint selection approaches applied to the $Incon_{seg}$ method on the NuImages validation dataset. The plotted values are the average of multiple runs.




**(a)** Object Detection **(b)** Semantic Segmentation

**Figure 5.14:** A cost efficiency comparison of the different checkpoint selection approaches applied to the $Incon_{seg}$ on the NuImages validation dataset. Lower values indicate a better efficiency.

considering the cost, this combined selection strategy shows its full potential. As visualized in Figure 5.14, this approach has the best cost efficiency and requires only 92.46% annotated objects compared to random selection. For the $Conf_{max}$ method the combined checkpoint selection is outperforming the object detection based checkpoint selection most of the time. While it is better for all cycles on the segmentation task, the object detection based checkpoint selection achieves better accuracy in the later cycles on the object detection task. The required objects are on-par for both object detection based and combined checkpoint selection with the $Conf_{max}$ method with a slight favour towards the combined selection.

### 5.5.2 Half Split

Two experiments were conducted to investigate the effect of an alternation of two methods at roughly the half of the active learning training cycles. In both experiments one method was used to the select the data for the labeling until 60% data has been used. From there on a different method was used to determine the best samples for the remaining labeling. The intuition was that a method that has a good performance in the early training phase, but a converging performance in the later phase, could be boosted by switching the sample selection strategy after 60% data have been selected. In $Loss + 1vs2$ this intuition was not validated. Even though the performance on the object detection task was boosted after the

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | 66.17 | 66.53 | 68.03 | 68.03 |
| $Incon_{seg-od}$ | 36.77 | 54.14 | **62.88** | 62.88 | 63.3 | **67.36** | 67.36 | 67.76 |
| $Incon_{seg-seg}$ | 36.77 | 55.29 | 60.77 | 62.85 | **64.47** | 64.67 | 66.06 | 66.06 |
| $Incon_{seg-both}$† | 36.77 | 56.41 | 62.46 | **63.03** | **66.0** | 66.0 | **68.9** | **68.9** |
| $Conf_{max-od}$ | 36.77 | 54.98 | 63.61 | 65.68 | 65.99 | **67.78** | **69.54** | **69.46** |
| $Conf_{max-both}$† | 36.77 | **58.99** | **66.63** | **66.63** | **66.63** | 66.79 | 68.01 | 68.01 |

**Table 5.11:** The average object detection mAP@0.5 results for the various checkpoint selection approaches applied to the $Incon_{seg}$ and $Conf_{max}$ method on the NuImages validation dataset at each cycle.

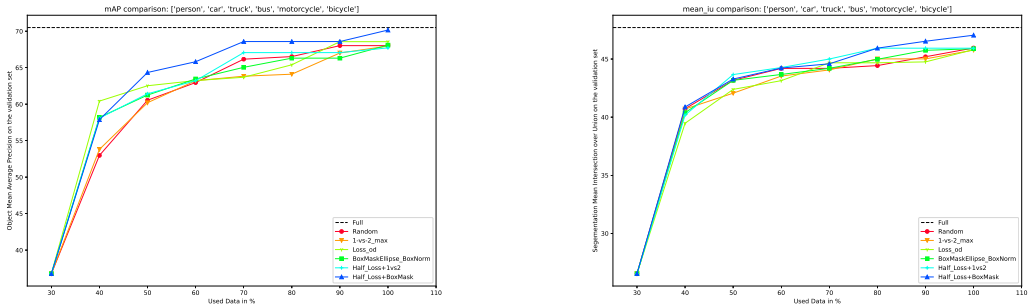
| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | 40.72 | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Incon_{seg-od}$ | 26.53 | 39.68 | 42.72 | 43.42 | 43.64 | 44.31 | 44.31 | 44.38 |
| $Incon_{seg-seg}$ | 26.53 | 39.44 | 41.86 | 43.75 | **44.85** | 45.23 | 45.62 | 46.16 |
| $Incon_{seg-both}$† | 26.53 | **41.12** | **43.06** | **44.03** | 44.21 | **45.94** | **46.15** | **47.01** |
| $Conf_{max-od}$ | 26.53 | 40.45 | 43.55 | 45.18 | 46.26 | 46.23 | 46.45 | 46.83 |
| $Conf_{max-both}$† | 26.53 | **43.06** | **44.98** | **45.54** | **46.35** | **46.85** | **46.85** | **46.89** |

**Table 5.12:** The average semantic segmentation mIU results for the various checkpoint selection approaches applied to the $Incon_{seg}$ and $Conf_{max}$ method on the NuImages validation dataset at each cycle.

60% cycle, it is not outperforming the single $1vs2_{avg}$ method at these cycles as can be seen in Table 5.13. On the segmentation task, the combination of the two methods lead to a small improvement of less than 0.1% compared to its single trained components, shown in Table 5.14. The combination of $Loss_{od}$ with $BoxMaskEl_{BN}$ outperforms its best performing single trained component $BoxMaskEl_{BN}$ by +2.06% mAP@0.5 and +1.17% mIU at 100% used data. This indicates, that combining two methods at a certain point during the training does help to increase the performance, but it does matter which methods are combined. However, it must be mentioned that the $Loss + BoxMask$ has a lower cost efficiency compared to its single trained components as shown in Figure 5.16. The higher annotation costs could be an explanation for the higher accuracy. A comparison of the object detection accuracy of both the $HalfSplit$ approaches and their components is visualized in Figure 5.15.

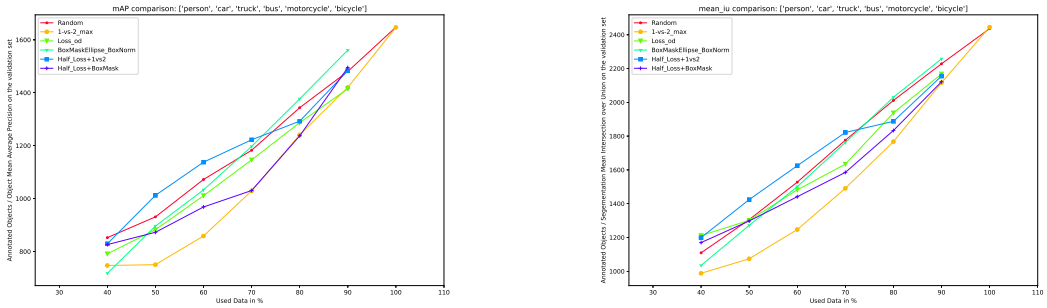
| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | 66.17 | 66.53 | 68.03 | 68.03 |
| $1vs2_{avg}$ | 36.77 | 54.66 | 58.0 | 61.34 | 62.08 | **68.91** | **69.19** | **69.45** |
| $Loss_{od}$ | 36.77 | **60.42** | **62.53** | 63.24 | 63.69 | 65.4 | 68.57 | 66.17 |
| $BoxMaskEl_{BN}$ | 36.77 | 58.19 | 61.25 | **63.46** | **65.05** | 66.31 | 66.17 | 68.1 |
| $Loss + 1vs2$ | 36.77 | **58.09** | 61.43 | 63.22 | 67.07 | 67.07 | 67.07 | 67.72 |
| $Loss + BoxMask$ | 36.77 | 57.87 | **64.35** | **65.82** | **68.59** | **68.59** | **68.59** | **70.16** |

**Table 5.13:** The average object detection mAP@0.5 results for the $HalfSplit$ approaches compared to their single trained components on the NuImages validation dataset at each cycle.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.15:** A result comparison of the $HalfSplit$ approaches against its single trained components on the NuImages validation dataset. The plotted values are the average of multiple runs.

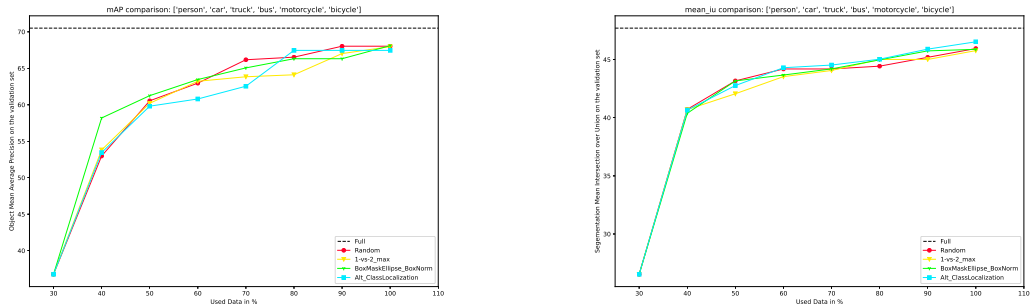



**(a)** Object Detection **(b)** Semantic Segmentation

**Figure 5.16:** A cost efficiency comparison of the $HalfSplit$ approaches against its single trained components on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| $Full$ | - | - | - | - | - | - | - | **47.71** |
| $Random$ | 26.53 | 40.72 | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $1vs2_{avg}$ | 26.53 | 39.51 | 41.95 | 43.8 | **45.37** | **45.37** | **45.85** | 45.85 |
| $Loss_{od}$ | 26.53 | 39.48 | 42.39 | 43.14 | 44.68 | 44.68 | 44.76 | 45.81 |
| $BoxMaskEl_{BN}$ | 26.53 | **40.38** | **43.17** | **43.67** | 44.2 | 44.98 | 45.75 | **45.88** |
| $Loss + 1vs2$ | 26.53 | 40.18 | **43.66** | **44.27** | **45.0** | 45.93 | 45.93 | 45.93 |
| $Loss + BoxMask$ | 26.53 | **40.88** | 43.28 | 44.23 | 44.59 | **45.94** | **46.53** | **47.05** |

**Table 5.14:** The average semantic segmentation mIU results for the $HalfSplit$ approaches compared to their single trained components on the NuImages validation dataset at each cycle.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

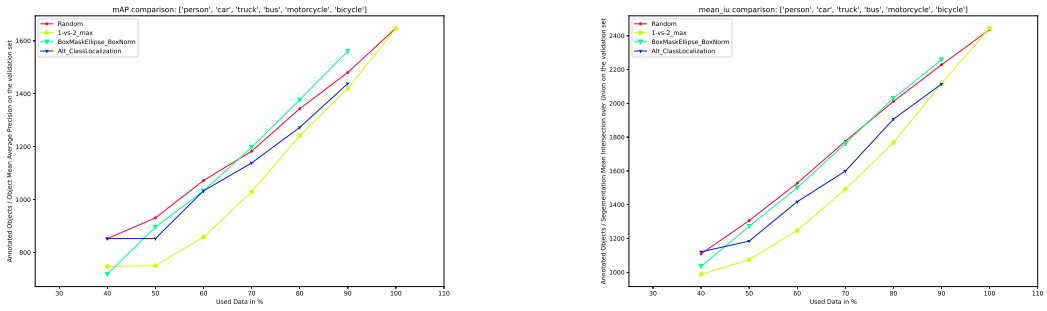
**Figure 5.17:** A result comparison of the alternating classification and localization optimizer approach against its non-alternating components on the NuImages validation dataset. The plotted values are the average of multiple runs.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **70.5** |
| *Random* | 36.77 | 52.98 | 60.55 | 62.96 | **66.17** | 66.53 | **68.03** | 68.03 |
| $1vs2_{max}$ | 36.77 | 53.81 | 60.2 | 63.22 | 63.84 | 64.11 | 67.02 | 67.96 |
| $BoxMaskEl_{BN}$ | 36.77 | **58.19** | **61.25** | **63.46** | **65.05** | 66.31 | 66.31 | **68.1** |
| $ClassLocalization$ | 36.77 | 53.45 | 59.81 | 60.8 | 62.54 | **67.45** | **67.45** | 67.45 |

**Table 5.15:** The average object detection mAP@0.5 results for the alternating classification and localization optimizer approach against its non-alternating components on the NuImages validation dataset at each cycle.

### 5.5.3 Alternation of Low Correlating Methods

Alternating two methods that focus on the optimization of the classification and the localization, respectively, should in theory boost the accuracy of both the classification and the localization of objects and following the improved accuracy, the overall performance should be increased as well. However, the results presented in Figure 5.17 and Table 5.15 show that this is not the case for the object detection task. The $ClassificationLocalization$ method is only at 80% used data better than all three other methods. During all other cycles, there is always one method performing better. In the early training cycles, $BoxMaskEl_{BN}$ is outperforming the other methods with a large margin achieving an mAP of 58.19%, which is nearly 5% higher than the achieved mAP of the alternating approach. And also in the last cycle the not alternating $BoxMaskEl_{BN}$ is the best performing method, outreaching the alternating method by 0.65% and performing on-par with random selection. On the semantic segmentation, however, the alternating training schema results in an improvement of the accuracy. $ClassLocalization$ is the best performing method in all but the first two cycles. Looking at the cost-efficiency shown in Figure 5.18, combining the two methods in an alternating fashion is a trade-off between the two methods. This indicates, that the alternating training schema has the potential to make use of both a strong, but expensive method and a bit weaker, but cheap method. This achieves a good performance while keeping the annotation effort low.




**(a)** Object Detection **(b)** Semantic Segmentation

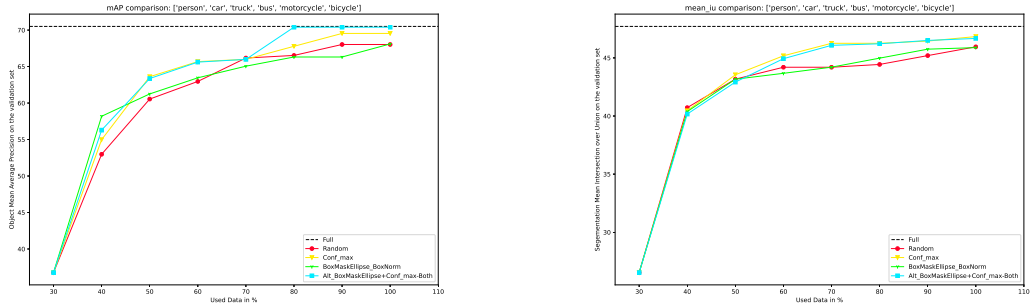
**Figure 5.18:** A cost efficiency comparison of the alternating classification and localization optimizer approach against its non-alternating components on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | - | **47.71** |
| *Random* | 26.53 | **40.72** | **43.17** | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $1vs2_{max}$ | 26.53 | **40.7** | 42.06 | 43.54 | 44.06 | 45.01 | 45.01 | 45.77 |
| $BoxMaskEl_{BN}$ | 26.53 | 40.38 | **43.17** | 43.67 | 44.2 | 44.98 | 45.75 | 45.88 |
| $ClassLocalization$ | 26.53 | 40.66 | 42.77 | **44.3** | **44.53** | **45.03** | **45.9** | **46.53** |

**Table 5.16:** The average semantic segmentation mIU results for the alternating classification and localization optimizer approach against its non-alternating components on the NuImages validation dataset at each cycle.

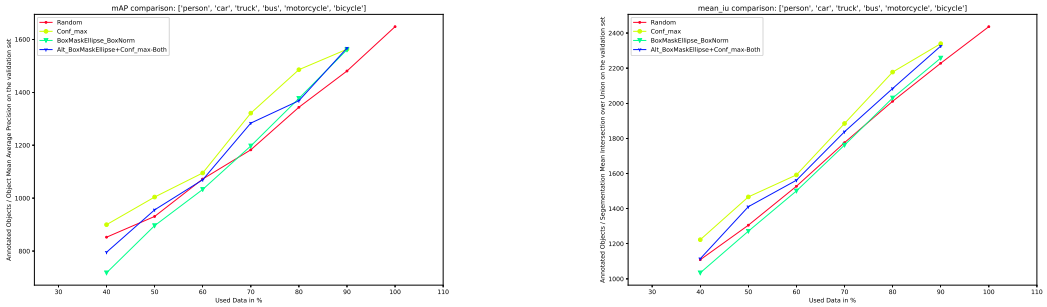
### 5.5.4 Alternation of Low Correlating Methods

The analysis of the dataset correlations of all methods with each other showed, that the two methods $Conf_{max}$ and $BoxMaskEl_{BN}$ have the lowest correlation averaged over all cycles. Therefore, these two were combined in an alternating training fashion, starting with $BoxMaskEl_{BN}$ in the first cycle. The progress of the accuracy on the two tasks compared to the non-alternating methods is visualized in Figure 5.19. On the semantic segmentation task, the alternation of the two methods does not result in an improvement. In most of the cycles, $Conf_{max}$ remains the superior method. Only in the cycle that used 80% of the data, the alternating approach is slightly better. However, the margin between the alternating approach and the non-alternating $Conf_{max}$ method is so small, that it can be neglected. On the other hand, the margin between the two approaches is larger on the object detection task. The alternating method achieves +2.62% mAP@0.5 at 80% data compared to $Conf_{max}$ and even +4.09% mAP@0.5 compared to $BoxMaskEl_{BN}$ at the same cycle. The cost-efficiency of $BoxMaskEl_{BN} - Conf_{max}$† is slightly better compared to $Conf_{max}$ and just marginally worse compared to $BoxMaskEl_{BN}$ as shown in Figure 5.20. It cannot be clearly said whether the combination of the two methods is causing the improved accuracy or the combined checkpoint selection that has been used for the $BoxMaskEl_{BN} - Conf_{max}$† experiment. This does also make the comparison against the non-alternating methods not completely fair. Nonetheless, it is the best overall method and thus, gives a good indication of what can be achieved with the alternating methods approach and the checkpoint selection strategy.




**(a)** Object Detection mAP@0.5 **(b)** Semantic Segmentation mIU

**Figure 5.19:** A result comparison of the alternating methods with the lowest dataset correlation against the non-alternating methods on the NuImages validation dataset. The plotted values are the average of multiple runs.




**(a)** Object Detection **(b)** Semantic Segmentation

**Figure 5.20:** A cost efficiency comparison of the alternating methods with the lowest dataset correlation against the non-alternating methods on the NuImages validation dataset. Lower values indicate a better efficiency.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **70.5** |
| *Random* | 52.98 | 60.55 | 62.96 | **66.17** | 66.53 | 68.03 | 68.03 |
| $Conf_{max}$ | 54.98 | **63.61** | **65.68** | 65.99 | 67.78 | 69.54 | 69.54 |
| $BoxMaskEl_{BN}$ | **58.19** | 61.25 | 63.46 | 65.05 | 66.31 | 66.31 | 68.1 |
| $BoxMaskEl_{BN} - Conf_{max}$† | 56.29 | 63.36 | 65.61 | 65.98 | **70.4** | **70.4** | **70.4** |

**Table 5.17:** The average object detection mAP@0.5 results for the alternating methods with the lowest dataset correlation against the non-alternating methods on the NuImages validation dataset at each cycle.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **47.71** |
| *Random* | **40.72** | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Conf_{max}$ | **40.45** | **43.55** | **45.18** | **46.26** | **46.26** | 46.45 | **46.83** |
| $BoxMaskEl_{BN}$ | 40.38 | 43.17 | 43.67 | 44.2 | 44.98 | 45.75 | 45.88 |
| $BoxMaskEl_{BN} - Conf_{max}$† | 40.17 | 42.92 | 44.93 | 46.08 | 46.21 | **46.5** | 46.67 |

**Table 5.18:** The average semantic segmentation mIU results for the alternating methods with the lowest dataset correlation against the non-alternating methods on the NuImages validation dataset at each cycle.

## 5.6 Method Comparison

In this section the best performing methods are compared against each other. This comparison is split into two parts. First, a quantitative comparison in Section 5.6.1, where the mAP@0.5 and the mIU on the validation subset of the datasets NuImages, A9 and Cityscapes are presented. And second, a qualitative comparison of the predictions on a selected set of input images from each dataset. In addition, the top- and lowest scored samples from each dataset are visualized for some of the methods in Section 5.6.2. For brevity the following notations were used. Alternating methods are indicated with the $-$ symbol. The $HalfSplit$ methods are displayed with the $+$ symbol. The $\dagger$ symbol indicates that the combined checkpoint selection has been used and $*$ indicates an alternating checkpoint selection.

### 5.6.1 Quantitative Results

#### NuImages

On the NuImages dataset, the active learning achieved great results as shown in Table 5.19. Random selection was outperformed by a non-alternating method for most of the cycles. Only in the cycle that used 70% of the data, the random selection approach has the highest mAP@0.5 of all non-alternating methods. The best single method is, in this experimental setting, is the least confidence approach using the maximum as aggregation method $Conf_{max}$. This method is simple, easy to apply for most object detection architectures and yet very effective. Its accuracy is highly competitive in both object detection and semantic detection. Table 5.20 shows that the least confidence method has a strong performance on the semantic segmentation as well. It even outperforms methods that were specifically designed for semantic segmentation such as $Incon_{seg}$. The only drawback of this method is the high annotation costs that this selection approach causes. The loss prediction module approach in general, and especially the combined approach $Loss_{comb}$ achieves comparable accuracy on both tasks and has a slightly better cost-efficiency. The novel $BoxMask$ approach achieves competitive performance on both tasks as well and has an even better cost-efficiency compared to the two prior mentioned methods. The other newly introduced method $Incon_{seg}$ has the best overall cost efficiency, which comes at the cost of a decreased object detection accuracy. The conducted experiments show that the alternation of two or more methods boosts the object detection accuracy. It is not possible to make a clear decision on which methods are best combined with each other and in which order. However, it is obvious that combining two methods that perform well results in a boosted overall accuracy. Combining a method that performs well on one task with a method that performs well on another task is also a good option. An increase in accuracy on the semantic segmentation task due to the alternation of two methods is not as strongly visible as it is the case for the object detection task. Another observation is that the checkpoint selection has a great influence on the final results. The combined checkpoint selection achieves the best results, both for object detection and semantic segmentation. It is better than the single-task focused selection and also better than the alternating, method-specific selection. Finally, it must be said that none of the methods investigated yielded in better results than the full data training. This could have various reasons, such as the implementation, the learning rate, the dataset split and others. More on that in Section 7. It is also worth mentioning that the NuImages dataset was curated by already applying active learning. Therefore, it is impressive to see, that the proposed methods and training techniques were still able to improve the accuracy compared to random selection.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | | - | - | - | - | - | **70.5** |
| *Random* | 52.98 | 60.55 | 62.96 | **66.17** | 66.53 | 68.03 | 68.03 |
| $Conf_{max}$ | 54.98 | **63.61** | **65.68** | 65.99 | 67.78 | **69.54** | **69.54** |
| $1vs2_{max}$ | 53.81 | 60.2 | 63.22 | 63.84 | 64.11 | 67.02 | 67.96 |
| $Loss_{combined}$ | 56.06 | 62.44 | 63.19 | 65.68 | **68.07** | 68.07 | 68.15 |
| $BoxMaskEl_{BN}$ | 58.19 | 61.25 | 63.46 | 65.05 | 66.31 | 66.31 | 68.1 |
| $BoxMask_{KL}$ | **58.79** | 60.0 | 63.38 | 64.92 | 64.94 | 66.04 | 67.78 |
| $Incon_{seg-seg}$ | 55.29 | 60.77 | 62.85 | 64.47 | 64.47 | 66.0 | 66.57 |
| *ClassLocalization* | 53.45 | 59.81 | 61.08 | 63.88 | 66.33 | 66.35 | 66.48 |
| $Loss_{od} - Loss_{seg}$ | 58.74 | **64.59** | 65.4 | 66.53 | 67.2 | 67.7 | 69.59 |
| $Loss_{od} - Loss_{seg}*$ | 52.99 | 59.37 | 64.99 | 64.99 | 69.21 | 69.21 | 68.92 |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | **59.02** | 62.56 | 63.85 | 64.54 | 66.02 | 66.05 | 67.94 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 49.08 | 57.44 | 63.41 | 65.79 | 65.79 | 67.3 | 67.3 |
| $BoxMaskEl_{BN} - Conf_{max}\dagger$ | 56.29 | 63.36 | 65.61 | 65.98 | **70.4** | **70.4** | **70.4** |
| $Loss_{od} - Conf_{max}\dagger$ | 55.13 | 60.97 | **66.16** | 66.16 | 66.16 | 69.43 | 69.43 |
| $Loss_{od} + 1vs2_{avg}$ | 58.09 | 61.43 | 63.22 | 67.07 | 67.07 | 67.07 | 67.72 |
| $Loss_{od} + BoxMaskEl_{BN}\dagger$ | 57.87 | 64.35 | 65.82 | **68.59** | 68.59 | 68.59 | 70.16 |

**Table 5.19:** The average object detection mAP@0.5 results of the best of all evaluated methods on the NuImages validation dataset at each cycle.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **47.71** |
| *Random* | **40.72** | 43.17 | 44.2 | 44.2 | 44.44 | 45.2 | 45.95 |
| $Conf_{max}$ | 40.45 | **43.55** | **45.18** | **46.26** | **46.26** | **46.45** | **46.83** |
| $1vs2_{max}$ | 40.7 | 42.06 | 43.54 | 44.06 | 45.01 | 45.01 | 45.77 |
| $Loss_{combined}$ | 40.45 | 42.87 | 44.34 | 44.74 | 45.09 | 45.96 | 46.5 |
| $BoxMaskEl_{BN}$ | 40.38 | 43.17 | 43.67 | 44.2 | 44.98 | 45.75 | 45.88 |
| $BoxMask_{KL}$ | 40.19 | 42.9 | 43.88 | 43.97 | 44.73 | 44.84 | 45.87 |
| $Incon_{seg-seg}$ | 39.44 | 41.86 | 43.75 | 44.85 | 45.7 | 45.93 | 46.2 |
| *ClassLocalization* | 40.66 | 42.77 | 44.06 | 44.33 | 45.06 | 45.62 | 46.53 |
| $Loss_{od} - Loss_{seg}$ | 40.68 | 43.49 | 43.98 | 44.34 | 45.11 | 45.82 | 45.82 |
| $Loss_{od} - Loss_{seg}*$ | 40.26 | 42.4 | 44.47 | 44.84 | 45.28 | 46.39 | 46.77 |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | **41.46** | 42.92 | 43.97 | 44.76 | 45.04 | 46.01 | 46.38 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 38.33 | 42.54 | 43.48 | 44.68 | 45.38 | 45.55 | 46.85 |
| $BoxMaskEl_{BN} - Conf_{max}\dagger$ | 40.17 | 42.92 | **44.93** | **46.08** | **46.21** | 46.5 | 46.67 |
| $Loss_{od} - Conf_{max}\dagger$ | 41.4 | **43.94** | 44.47 | 45.3 | 45.35 | **47.07** | **47.07** |
| $Loss_{od} + 1vs2_{avg}$ | 40.18 | 43.66 | 44.27 | 45.0 | 45.93 | 45.93 | 45.93 |
| $Loss_{od} + BoxMaskEl_{BN}\dagger$ | 40.88 | 43.28 | 44.23 | 44.59 | 45.94 | 46.53 | 47.05 |

**Table 5.20:** The average semantic segmentation mIU results of the best of all evaluated methods on the NuImages validation dataset at each cycle.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| $Full$ | - | - | - | - | - | - | 65.48 |
| $Random$ | **46.79** | 46.79 | 46.79 | 54.15 | 55.37 | 64.06 | 64.06 |
| $Conf_{max}$ | 41.08 | 45.74 | 49.37 | 49.37 | 51.32 | 51.32 | 54.02 |
| $Conf_{max}$† | 44.69 | 44.69 | 55.84 | 55.84 | 55.84 | 55.84 | 55.84 |
| $1vs2_{max}$ | 42.17 | **59.68** | **59.68** | **59.68** | **61.48** | **66.46** | **66.46** |
| $Loss_{combined}$ | 44.51 | 44.51 | 53.67 | 53.67 | 55.02 | 55.02 | 55.02 |
| $Incon_{seg-seg}$ | 38.03 | 43.23 | 52.84 | 52.84 | 56.82 | 56.82 | 56.82 |
| $Incon_{seg-both}$† | 43.63 | 43.63 | 44.12 | 48.57 | 53.88 | 53.88 | 53.88 |
| $BoxMaskEllipse_{BN}$ | 39.28 | 41.6 | 52.74 | 56.05 | 56.05 | 56.05 | 56.05 |
| $BoxMask_{KL}$ | 43.69 | 44.49 | 53.41 | 56.19 | 56.19 | 56.19 | 56.19 |
| $Loss_{od} - Loss_{seg}$ | **51.36** | 51.36 | 51.36 | 51.36 | 51.36 | 51.36 | 60.67 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 34.39 | 41.29 | 48.07 | 48.07 | **67.31** | **69.32** | **69.32** |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | 48.85 | 48.85 | 52.62 | 52.62 | 52.62 | 52.62 | 53.3 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{od}$† | 40.87 | 50.31 | 50.31 | 52.73 | 56.07 | 56.07 | 56.07 |
| $Loss_{od} - Conf_{max}$ | 37.65 | 46.68 | 46.68 | 46.68 | 54.86 | 54.86 | 55.15 |
| $BoxMaskEl_{BN} - Conf_{max}$ | 46.24 | **63.92** | **63.92** | **63.92** | 63.92 | 63.92 | 63.92 |
| $Loss_{od} + BoxMaskEl_{BN}$† | 46.9 | 54.94 | 54.94 | 57.99 | 57.99 | 57.99 | 57.99 |

**Table 5.21:** The average object detection mAP@0.5 results of all evaluated methods on the A9 validation dataset at each cycle.

### A9

The object detection results of the conducted experiments on the A9 dataset are given in Table 5.21. This draws a slightly different picture than the results on the NuImages dataset. On the object detection, the $1vs2_{max}$ method achieves the best accuracy and outperforms the other non-alternating methods by a large margin. This approach has the overall lowest costs as it requires only 79.43% of the full dataset annotations and yet it achieves higher accuracy than the full data training. The overall best accuracy is again reached through the method alternating training strategy. $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ has a +3.84% mAP@0.5 compared to the full data training and used only 81.03% of the full dataset annotations. This validates the finding from the NuImages dataset, that the alternating training strategy boosts the overall object detection accuracy. This can also be observed for the semantic segmentation task as shown in Table 5.22. However, the margin between the non-alternating and the alternating methods is significantly low. This indicates again, just like on the NuImages dataset, that the conducted sample selection strategies are more suitable for the object detection task. The effectiveness of active learning on a small dataset is arguable and especially for the segmentation task, the scenes in the dataset might not be diverse enough, which could be a possible explanation for the poor improvement achieved by using active learning. On the other hand, even though the accuracy of the full data training is not reached for the semantic segmentation task, the required annotation costs are much lower if active learning is used. The $Loss_{od} - Conf_{max}$ approach, for example, required just 41.60% of the full data annotations to reach 92.90% of the full data training mIU. It must also be mentioned that the full data training most likely overfitted the training and validation data. This becomes visible if looked at the qualitative results presented in Section 5.6.2.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **40.03** |
| *Random* | 30.96 | 34.55 | 36.35 | 37.48 | 37.84 | 38.11 | 38.11 |
| $Conf_{max}$ | 32.53 | 33.67 | 33.67 | 35.88 | 36.14 | 37.28 | 37.65 |
| $Conf_{max}$† | **33.24** | **34.81** | **36.63** | **38.11** | **38.11** | 38.11 | **38.79** |
| $1vs2_{max}$ | 31.3 | 33.69 | 35.08 | 35.86 | 35.86 | 37.93 | 37.93 |
| $Loss_{combined}$ | 31.73 | 33.1 | 36.09 | 36.09 | 36.09 | 37.89 | 37.89 |
| $Incon_{seg-seg}$ | 32.81 | 32.81 | 33.45 | 36.82 | 37.97 | 38.17 | 38.57 |
| $Incon_{seg-both}$† | 29.53 | 32.64 | 35.7 | 36.52 | 38.0 | **38.34** | 38.34 |
| $BoxMaskEllipse_{BN}$ | 30.89 | 34.3 | 35.84 | 35.84 | 37.53 | 37.53 | 37.53 |
| $BoxMask_{KL}$ | 32.55 | 34.0 | 35.7 | 37.46 | 37.46 | 38.08 | 38.13 |
| $Loss_{od} - Loss_{seg}$ | 32.37 | 32.37 | 34.4 | 34.44 | 34.44 | 37.4 | 37.4 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 28.86 | 33.45 | 34.57 | 37.04 | **38.34** | 38.34 | 38.34 |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | 33.02 | 33.59 | 34.07 | 36.02 | 36.17 | 38.01 | 38.61 |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{od}$† | 30.48 | 32.42 | 35.78 | 36.72 | 36.98 | 37.62 | 37.83 |
| $Loss_{od} - Conf_{max}$ | **33.69** | **35.42** | **37.19** | 37.25 | 37.28 | 38.7 | 38.91 |
| $BoxMaskEl_{BN} + Conf_{max}$ | 33.54 | 35.23 | 36.72 | 37.44 | 38.14 | **38.96** | **38.96** |
| $Loss_{od} + BoxMaskEl_{BN}$† | 31.74 | 34.9 | 35.82 | 36.54 | 37.65 | 37.97 | 38.47 |

**Table 5.22:** The average semantic segmentation mIU results of all evaluated methods on the A9 validation dataset at each cycle.

#### Cityscapes

Due to the large dataset size and the resulting training time, fewer experiments could be conducted on this dataset. The mean average precision of the object detection is shown in Table 5.23 and the mean intersection over union results of the semantic segmentation can be found in Table 5.24. While in the early three cycles always at least one active learning method outperforms random selection on the object detection task, the opposite is the case for the remaining cycles. This indicates, that active learning is only beneficial until a certain amount of data has been used. From that point on, the effectiveness of active learning converges and is no better than random selection. Nonetheless, active learning helps to reduce the annotation costs, while keeping the mAP@0.5 relatively competitive. The $BoxMaskEl_{BN}$ method achieves 93.26% of the full data trained mAP@0.5 with just using 89.72% of the available fine annotated objects. The segmentation results achieved by the various methods are again closely together. The combined loss prediction module is outperforming random selection on the segmentation task by +0.26% and required only 91.81% of the fine annotated objects to reach 95.69% of the full data trained mIU.

### 5.6.2 Qualitative Results

In this section qualitative results on the three datasets NuImages, A9 and Cityscapes are presented. For each dataset, there are the predictions of both object detection and semantic segmentation on four randomly selected images from the test sets visualized. The predictions are taken from the models trained using the various sample selections strategies. The results are shown for two training states. One is after 40% of the dataset has been used to train the model and the other is after 100% of the data samples have been used during training. Table 5.25 shows to which class a colour belongs in the prediction. In addition to the prediction results, the samples with the highest and lowest scores for each sample selection method

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **58.57** |
| *Random* | 40.62 | 46.13 | 48.73 | **52.08** | **54.24** | **56.93** | **57.27** |
| $Conf_{max}$ | **44.82** | 44.82 | 46.25 | 47.67 | 51.2 | 52.26 | 54.58 |
| $1vs2_{max}$ | 39.39 | 44.91 | 45.54 | 49.6 | 50.08 | 52.39 | 54.69 |
| $Incon_{seg-seg}$ | 42.79 | **47.02** | 49.9 | 51.97 | 53.28 | 53.28 | 53.28 |
| $Loss_{combined}$ | 42.68 | 46.49 | 50.14 | 50.3 | 51.88 | 52.27 | 52.97 |
| $BoxMaskEllipse_{BN}$ | 41.88 | 45.7 | **50.57** | 50.6 | 52.03 | 54.62 | 56.48 |
| $Loss_{od} - Loss_{seg}$ | **42.82** | **46.84** | **49.5** | 50.53 | 50.53 | 53.43 | **53.64** |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 39.52 | 45.86 | 48.32 | 48.32 | 48.32 | 53.41 | 53.48 |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | 40.32 | 44.34 | 46.17 | **51.67** | **52.38** | **53.63** | 53.63 |

**Table 5.23:** The average object detection mAP@0.5 results of all evaluated methods on the Cityscapes validation dataset at each cycle.

| **Experiment** | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| *Full* | - | - | - | - | - | - | **51.25** |
| *Random* | 44.8 | 46.56 | **47.66** | 47.66 | 48.35 | 48.65 | 49.03 |
| $Conf_{max}$ | **45.78** | **46.65** | 46.94 | 47.69 | 48.25 | 48.32 | 49.05 |
| $1vs2_{max}$ | 45.15 | 46.26 | 47.11 | 47.51 | 47.97 | 48.8 | 49.2 |
| $Incon_{seg-seg}$ | 43.4 | 46.5 | 47.16 | 47.72 | 48.16 | 48.71 | 49.1 |
| $Loss_{combined}$ | 44.81 | 46.53 | 47.59 | **48.08** | **48.36** | **49.04** | **49.29** |
| $BoxMaskEllipse_{BN}$ | 45.21 | 46.53 | 46.57 | 47.45 | **48.36** | 48.97 | 48.97 |
| $Loss_{od} - Loss_{seg}$ | **46.08** | **46.58** | **47.17** | **48.31** | **48.8** | **48.82** | **49.11** |
| $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$ | 44.69 | 46.36 | 47.11 | 47.62 | 48.05 | 48.76 | 48.76 |
| $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$ | 45.61 | 46.44 | 47.5 | 47.65 | 48.15 | 48.43 | 49.05 |

**Table 5.24:** The average semantic segmentation mIU results of all evaluated methods on the Cityscapes validation dataset at each cycle.

| | | | |
|---|---|---|---|
| Pedestrian | Car | Motorcycle | Ignore |
| Bus | Truck | Bicycle | |
| Traffic Sign | Rider | Sidewalk | Traffic Light |
| Road | Vegetation | Terrain | Sky |

**Table 5.25:** The color codes for both the object detection and semantic segmentation.

are presented at the same training states. The highest score does not necessarily mean the absolute score value but lists the images most likely to be selected for labelling. This is due to the fact that some methods favour lower scores, while other methods selected the highest scored samples. The results and selected samples from the other states can be found online [Fri22].

#### NuImage

The images for the qualitative analysis on the NuImages dataset are randomly selected from the test set. The ground truth for those images is not publicly available. The results of the state at the first active cycle, which corresponds to 40% used data, and the last active cycle, where 100% of the data is used, are shown in the Figures 5.21 and 5.22, respectively. In Table 5.25 the color of each class is visualized. The confidence threshold for the NMS was set to 60% during the inference. The presented qualitative results show that the model can produce decent results at just 40% data used, no matter what method is applied. While the difference between the methods might not look much at first, they do exist. One example where the difference is well observable is the first column. In this image, a van is parked in the left-hand side background. Only $Conf_{max}$ and $Inconseg_{seg}$ detect and classify this object correctly at 40% data. At the later training stage where 100% data was used, the models of all methods detect this object. This shows, that the two prior mentioned methods have an advantage against the others regarding objects like this particular van. More interesting is the selection made by the presented methods. The samples at the first cycle with the highest scores are shown in Figure 5.23 and the images with the lowest scores are shown in Figure 5.24. The highest scores samples at the last cycle are displayed in Figure 5.25 and the lowest scored samples of the same cycle can be found in Figure 5.26. Looking at these selected images, the $Inconseg_{seg}$ is particularly interesting. The lowest scored images are all taken at night, showing no objects at all. $1vs2_{max}$ is similar, as the lowest scored images contain no objects as well. In general, one can see that images that contain one large object blocking the scene are more likely to have a low score. The image of the FedEx car, for example, is listed as one of the lowest scored images for many methods. The well-performing methods $BoxMaskEl_{BN} - Conf_{max}$† and $Loss_{od} + BoxMaskEl_{BN}$† stand out because of their selection of very diverse scenes. The set of highly scored images contains scenes at night, in difficult weather conditions and scenes with few and many objects in them. Some images are selected from many methods equally as either highly scored or low scored. This indicates that there exists a subset in the dataset that can be considered a good or bad selection, regardless of the methods used. A combination of the methods at the same cycle could extract those easily. An experiment in this direction remains for future work.

#### A9

For the qualitative analysis of the methods on the A9 dataset, a sequence of unseen data taken from each of the four cameras has been used. The presented images here are a random selection from this set. The inference results on the first cycle that used 40% of the training data are shown in Figure 5.27. The predictions on the same images using the models that

used 100% of the training data are visualized in Figure 5.28. The NMS threshold for the inference on the A9 dataset was set to 30%. The poor results of the full data trained model in the first coloured row indicate clearly, that the model overfitted the training and validation data. This was hard to avoid as the number of iterations should match the iterations performed for all experiments to make a fair comparison possible. This also shows, that if only a small amount of data is available, active learning can help to prevent overfitting and still reach high accuracy. In between the methods, the differences in the prediction quality are minor. The shadows of the overhead structure in the first column image are often detected as an object in the semantic segmentation. The same accounts for the green verge between the two driving lanes in the third column image. These incorrect classifications are solved in the last training cycle. The object detection is more accurate as well in general, but a clear distinction between the selection methods cannot be made. Again, more interesting insights can be gained if looked at the Figures 5.29 and 5.30 showing the highest and lowest scored images at the first training cycle. In general, most of the methods select crowded scenes for the next training cycle and score the images with no or only a few objects lower. The two methods $BoxMaskEllipse_{BN}$ and $Loss_{od} - Loss_{seg}$ both selected an image from the scene of an accident, which is from the human annotator perspective an interesting sample for the training of a model. The $BoxMaskEllipse_{BN} - Conf_{max}$ method selected multiple images from the scene of a motorcycle as the highest scored images. This again, just like on the NuImages dataset, shows that this method selects a diverse set of objects. This furthermore shows that the scoring methodology works as similar scenes have a similar score. However, this is usually not desired, as two completely different scenes promise a higher training effect. This will be taken into account in future developments of the method. The highest and lowest scored images at the last cycle are presented in Figure 5.31 and 5.32, respectively. Here again, images from the car accident sequence are selected by multiple methods. This confirms the assumption made in Section 5.6.2 that there is a specific subset that is considered especially valuable for the training progress, which could be defined by using multiple methods in the same cycle.

#### Cityscapes

The predictions on a randomly selected subset of the Cityscapes test data is shown in Figure 5.33 for the first cycle and Figure 5.34 shows the results of the last cycle on the same set of images. While the semantic segmentation results are quite accurate with just 40% used data, the object detection results are relatively bad at this training stage. This is most likely due to not optimized NMS threshold, which was set to 40% during the inference of this qualitative evaluation. The accuracy of both tasks is improving a lot over all training cycles resulting in a good accuracy at 100% used data. The samples at the first cycle with the highest scores are shown in Figure 5.35 and the images with the lowest scores are shown in Figure 5.36. The highest scored samples at the last cycle are displayed in Figure 5.37 and the lowest scored samples of the same cycle can be found in Figure 5.38. The selection on the Cityscapes dataset correlates to the one on the NuImages and A9 dataset. The higher scored images are mainly crowded scenes containing many cars and pedestrians. The lowest scored images are for most of the methods scenes with no objects and a neutral background. In contrast to that stands one high scored image from the $Loss_{od} - Loss_{seg}$ method shown in the last column of the third-last row of Figure 5.35. This image contains no objects, but has a very interesting background. This indicates that a model might not only be uncertain about an object shape or color, but the background of a scene can also add a lot of uncertainty to the predictions, especially for the semantic segmentation task. Another insight that can be gained by looking at Figure 5.37 is that towards the end of the training, most methods select less crowded scenes as the highest scored images, compared to the first cycle selection. However, the

scenes either contain special objects, such as a person on a bicycle extending his arm to turn, or more diverse lightning conditions, such as a reflective road surface or a lot of shadows. Just like on the other two datasets, the methods often select the same samples as either high or low scored. This further supports the assumptions made earlier.

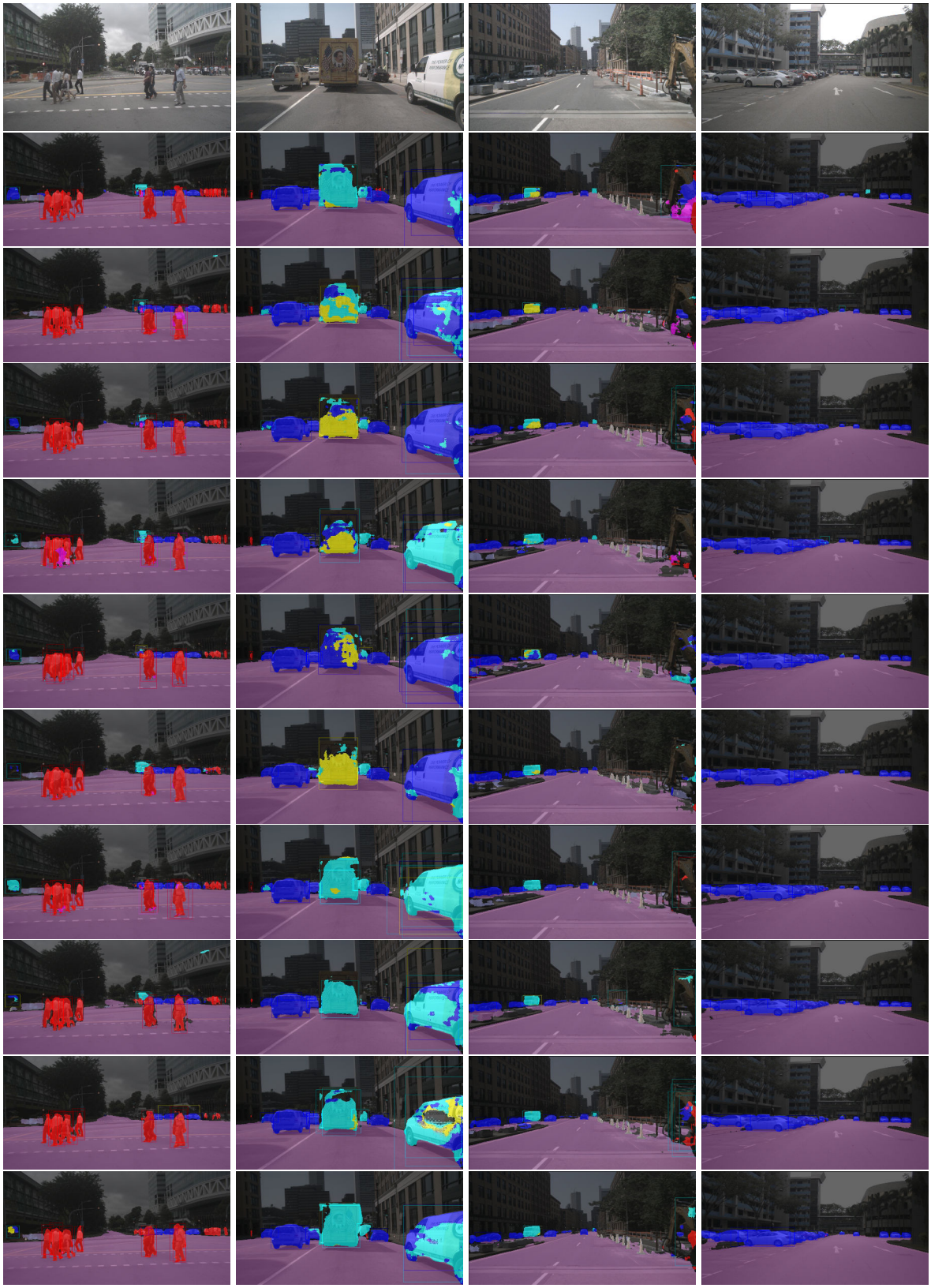


**Figure 5.21:** Qualitative results on the NuImage dataset with 40% data used in total. The first colored row are the results from the $Full$ method using 100% data from the beginning, the other rows show these applied selection strategies. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}$†, $Loss_{od} + BoxMaskEl_{BN}$†.

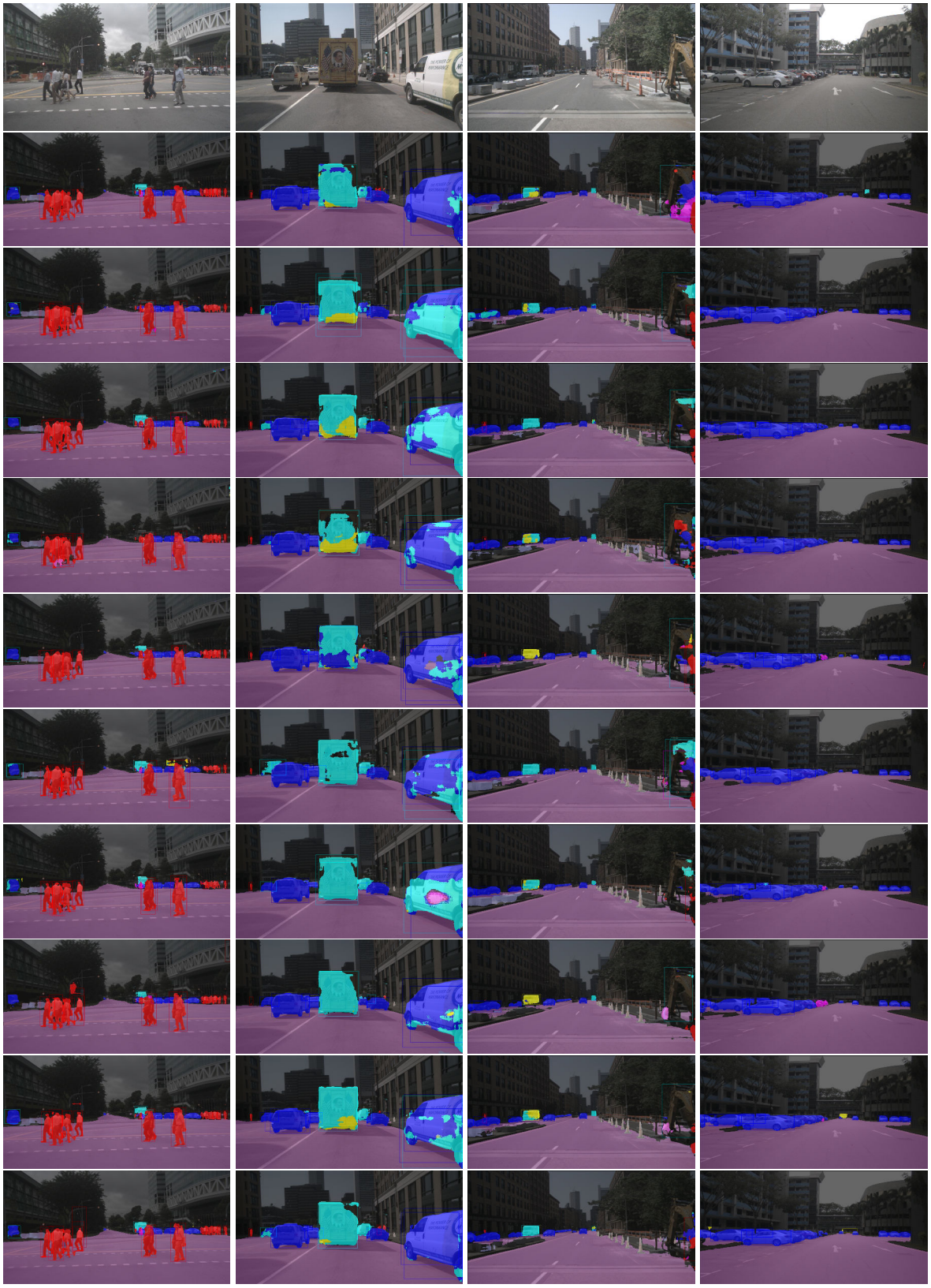


**Figure 5.22:** Qualitative results on the NuImage dataset with 100% data used in total. Each row is a different sample selection approach. These are from top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}\dagger, Loss_{od} + BoxMaskEl_{BN}\dagger$.

**Figure 5.23:** The remaining NuImage samples at the first cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}$†, $Loss_{od} + BoxMaskEl_{BN}$†.


**Figure 5.24:** The remaining NuImage samples at the first cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}\dagger, Loss_{od} + BoxMaskEl_{BN}\dagger$.

**Figure 5.25:** The remaining NuImage samples at the last cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}$†, $Loss_{od} + BoxMaskEl_{BN}$†.

**Figure 5.26:** The remaining NuImage samples at the last cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}\dagger, Loss_{od} + BoxMaskEl_{BN}\dagger$.

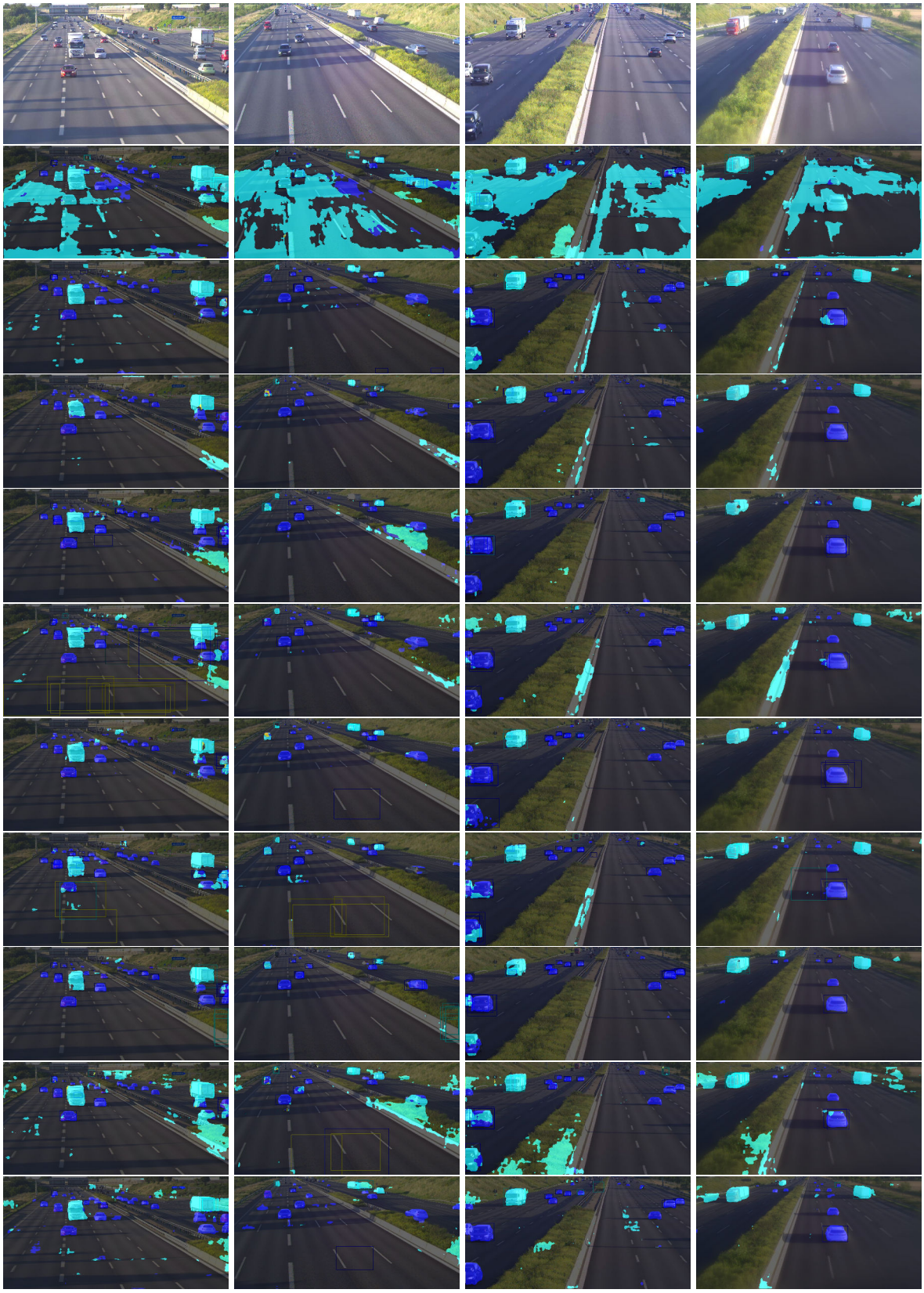


**Figure 5.27:** Qualitative results on the A9 dataset with 40% data used in total. The first colored row are the results from the $Full$ method using 100% data from the beginning, the other rows show these applied selection strategies. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}, BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}, BoxMaskEl_{BN} - Conf_{max}$.

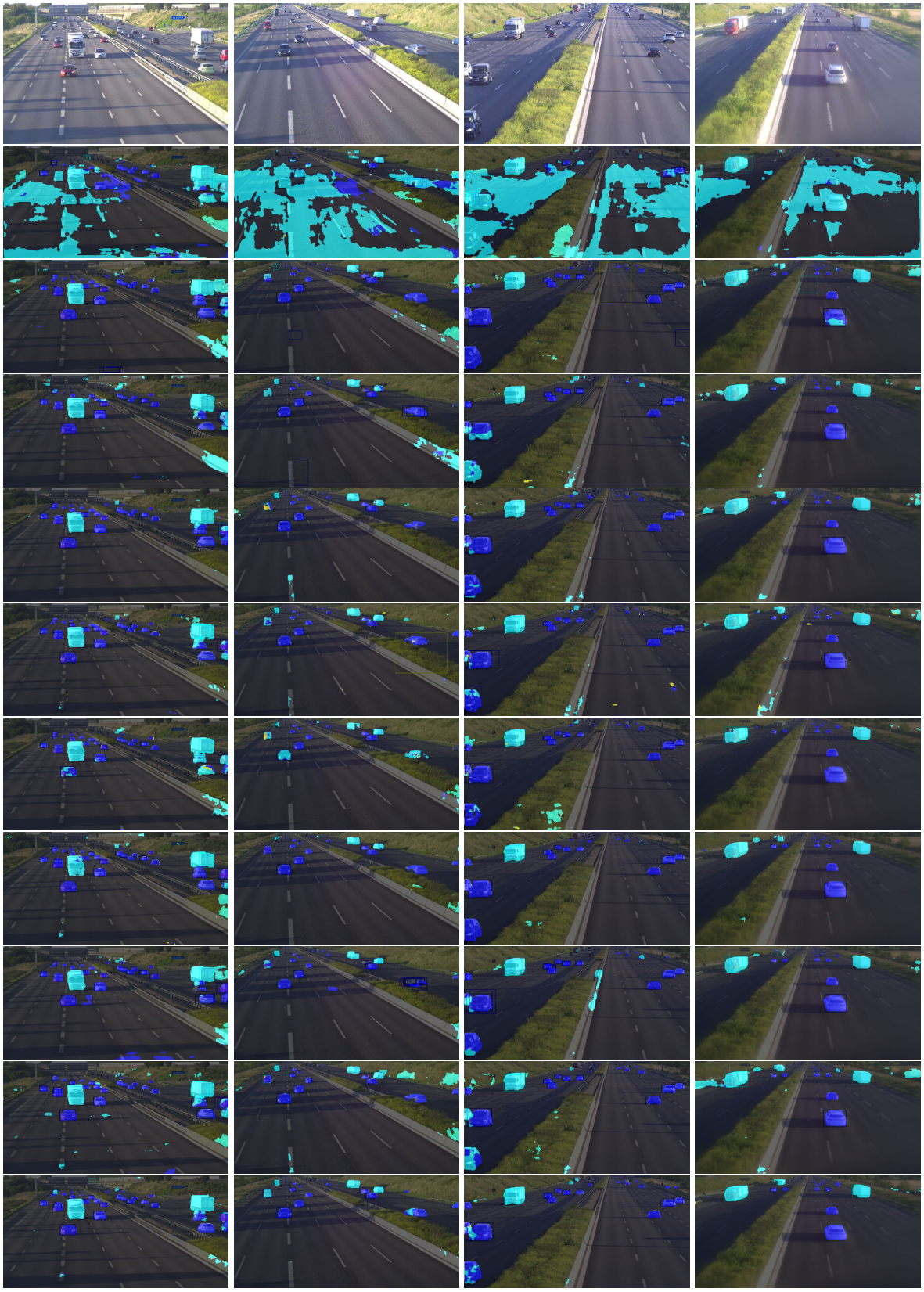


**Figure 5.28:** Qualitative results on the A9 dataset with 100% data used in total. Each row is a different sample selection approach. These are from top to bottom: $Full$, $Random$, $Conf_{max}$, $1vs2_{max}$, $Inconseg_{seg}$, $Loss_{combined}$, $BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Conf_{max}$.

**Figure 5.29:** The remaining A9 samples at the first cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Conf_{max}$.

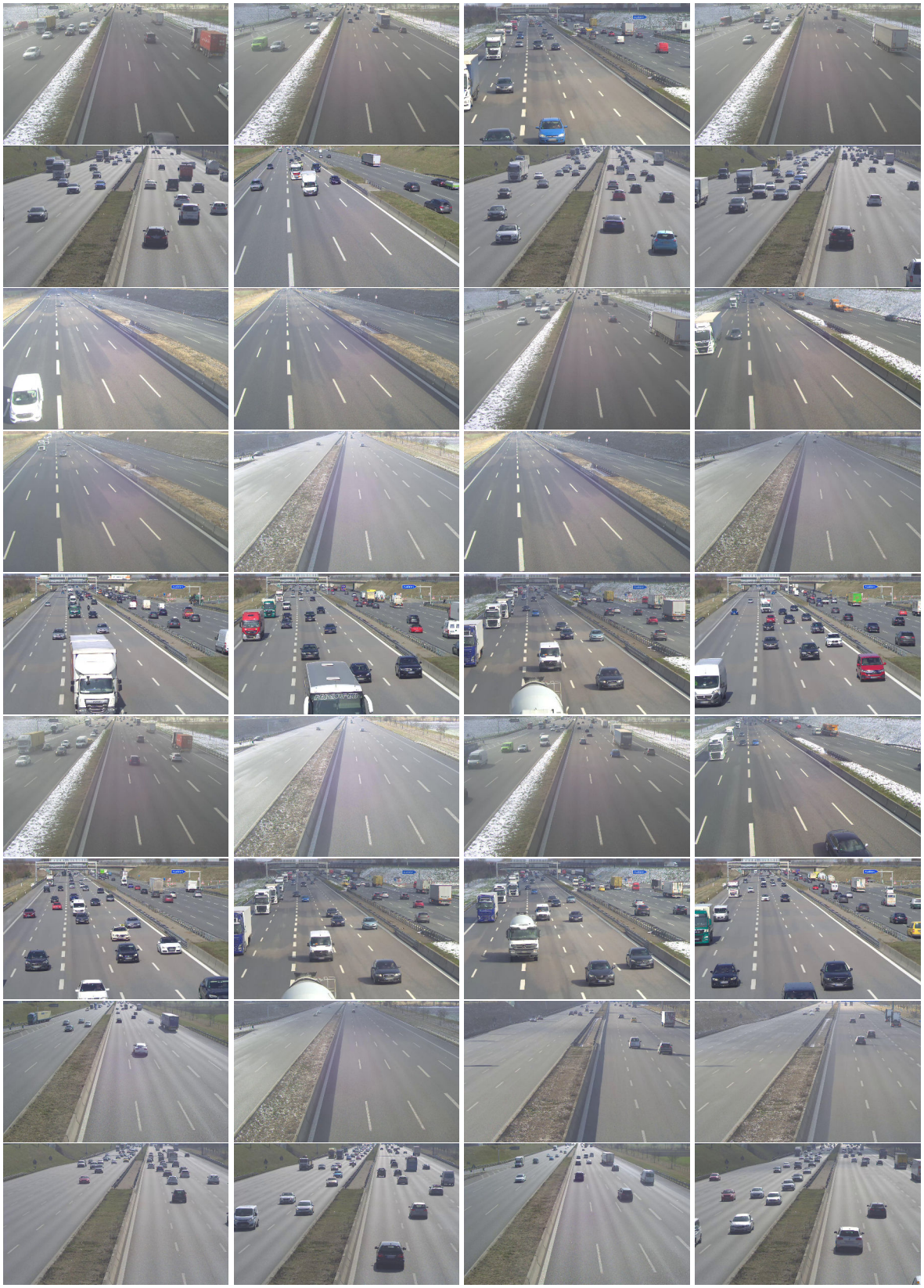


**Figure 5.30:** The remaining A9 samples at the first cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Conf_{max}$.

**Figure 5.31:** The remaining A9 samples at the last cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Conf_{max}$.

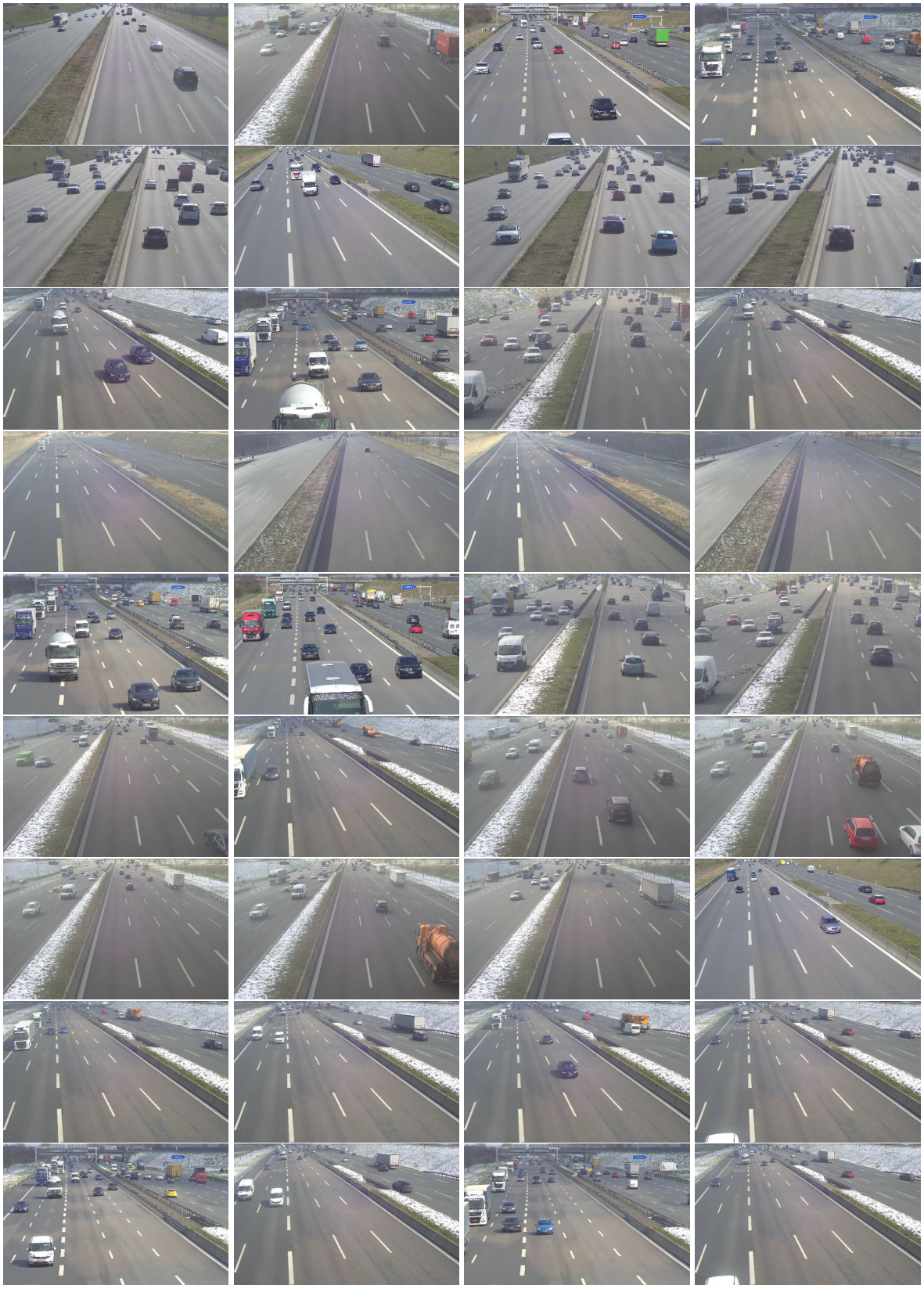


**Figure 5.32:** The remaining A9 samples at the last cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEllipse_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Conf_{max}$.

**Figure 5.33:** Qualitative results on the Cityscapes dataset with 40% data used in total. The first colored row are the results from the $Full$ method using 100% data from the beginning, the other rows show these applied selection strategies. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}, BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}, Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

**Figure 5.34:** Qualitative results on the NuImage dataset with 100% data used in total. Each row is a different sample selection approach. These are from top to bottom: $Full$, $Random$, $Conf_{max}$, $1vs2_{max}$, $Inconseg_{seg}$, $Loss_{combined}$, $BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

**Figure 5.35:** The remaining Cityscapes samples at the first cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

**Figure 5.36:** The remaining Cityscapes samples at the first cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

**Figure 5.37:** The remaining Cityscapes samples at the last cycle that have the highest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

**Figure 5.38:** The remaining Cityscapes samples at the last cycle that have the lowest score according to proposed methods. Each row represents a different selection method. From top to bottom: $Random, Conf_{max}, 1vs2_{max}, Inconseg_{seg}, Loss_{combined}, BoxMaskEl_{BN}$, $Loss_{od} - Loss_{seg}$, $BoxMaskEl_{BN} - Loss_{od} - Loss_{seg}$, $Loss_{od} - BoxMaskEl_{BN} - Loss_{seg}$.

# Chapter 6

# Discussion

In the following, the results shown in the previous chapter will be discussed. The sections of this chapter are structured by the general findings of this thesis. Starting with the influence of the checkpoint selection on the overall accuracy in Section 6.1, followed by the discussion of the KL-divergence in Section 6.2. The alternating training schema will be discussed in Section 6.3 and in Section 6.4 the possible influence of the size and split of a dataset is discussed. In Section 6.5 the effect of a single-task method on the respective other task is analysed. Finally, some of the unsuccessful experiments are presented in Section 6.6.

## 6.1 Checkpoint Selection

Selecting the intermediate checkpoint based on both tasks gives a good trade-off between the two tasks. A weighted selection could also be possible here if one task is more important than the other one. On the $Incon_{seg}$ method the combined checkpoint selection resulted in better overall performance on both tasks. This confirms the common theory, that two complement tasks can each boost the other's performance. On the NuImage dataset, the combined checkpoint selection on $Incon_{seg}$ outperforms the single task selections towards the end. This indicates that if a single task focused checkpoint selection is used, the model specializes itself on that specific single task in the later training phase resulting in a good performance on the focused task and a reduced performance on the other task. A combined checkpoint selection remedies this and keeps the performance on both tasks high, even if the training progresses. On the A9 dataset, object detection is also improved by using the combined checkpoint selection. The semantic segmentation on this dataset is +2.46% better if the single segmentation based checkpoint selection is applied. One reason for that could be the applied pre-training using the Cityscapes dataset. Here, the single object detection focused checkpoint selection is used. This way, the initial weights are already kind of specialized for the object detection task. Therefore, a combined selection has not the same effect as on the NuImages dataset and a single segmentation checkpoint selection results in a better performance on the segmentation task. This leads to the conclusion that if a model is trained from scratch, the combined checkpoint selection is the best choice. If pre-trained weights are used, one should consider how these were trained and use the opposite task during the checkpoint selection.

## 6.2 KL-Divergence

In both the approaches $Incon_{seg}$ and $BoxMask$ the KL-divergence was used to compute the inconsistency between two predicted class probability distributions. Even though it is a more

sophisticated approach, the results were not better than its much simpler counterpart which often just counts the unequal class predictions. The achieved accuracy of the KL-divergence approach is on-par or sometimes slightly worse compared to the simpler method. This is the case for both object detection and semantic segmentation. This suggests that looking at the most likely class prediction is enough and the full distribution must not be considered. The KL-divergence approaches are slightly more complex to implement, as the predictions must be a probability distribution. This thesis aims to reduce the annotation cost, which cannot be achieved with the KL-divergence. On both the NuImages dataset, as well as on the A9 dataset, the approaches that used the KL-divergence have higher annotation costs. In addition to that comes an increased computation time, which is important to consider if one wants to reduce the overall costs. If the selection of the samples and therefore the training of the network takes more time, the required resources are occupied for a longer period, which then leads to higher costs as well. In the used implementation, the computation of the KL-divergence takes roughly factor 10 of the time that is needed for the simpler approach. The combination of equal accuracy, worse annotation costs and worse computation time, lead to the suggestion that the KL-divergence should not be used and that a simpler methodology is better instead.

## 6.3 Alternating Training Schema

As proposed in the publications by Reichart et al. [Rei+08] and Ikhawantri et al. [Ikh+18] alternating two or more selection strategies can boost the overall performance of active learning for natural language processing. The presented results confirm this hypothesis in the context of autonomous driving as well and experiments applying this training methodology achieved improved results in object detection and semantic segmentation. However, the results conducted in this thesis also show, that the choice of methods matters depending on the dataset that is used. Randomly selecting a method at each cycle might result in a good performance, but it is questionable whether this is the result of the methodology or rather a statistical coincidence. Especially on the object detection task, an alternation of methods resulted in a large improvement in accuracy compared to the non-alternating methods. This could also be confirmed on the A9 dataset. The segmentation accuracy improvement of the alternating training approach was not as large as for the object detection but is still noticeable. On the Cityscapes dataset, the alternation of methods was not successful. However, the evaluation on this dataset was not as extensive due to the limited time. Combining more methods and evaluating them on the Cityscapes dataset is an interesting addition for future work.

## 6.4 Dataset Size and Split

If the dataset size is small, as the A9 dataset, for example, active learning is hardly outperforming random selection. The same accounts for too large datasets. The results on the Cityscapes dataset show, that active learning is helpful in the earlier training cycles but loses its advantage to random selection in the later cycles. On both datasets, the poor performance of all the active learning methods can be due to a miscalibration of the hyper-parameters. Possible tweaking points could be the size of the randomly selected initial dataset which is used to train the model before active learning methods are applied. In the conducted experiments this was set to 30% of the full dataset. This value was selected as it resulted in the best performance on the NuImages dataset. However, for a small dataset like A9, or a

large dataset like Cityscapes, this value possibly should be de- or increased. The remaining training cycles always select an additional 10% of the data and add them to the training pool. Again, this value is the result of the experiments on the NuImages dataset. Another value could result in higher accuracy on the other datasets. It would be interesting how the data split and initial data amount affect the performance of active learning methods compared to random selection. This research remains for future work.

## 6.5 Effect of Task-Focused Sample Selection on the Other Task

Another interesting discussion point is the effect of a sample selection strategy which is focusing on one task on the other task. Such single-task focused methods are $Conf$, $1vs2$, $Incon_{od}$ and $Loss_{od}$ for the object detection task. The methods $Incon_{seg}$ and $Loss_{seg}$ were solely using semantic segmentation features to estimate the uncertainty of the model. The intuitive assumption that the object detection focused methods perform best on the object detection, and the segmentation focused ones perform best on the semantic segmentation could only be partially confirmed. The $Loss_{seg}$ method for example was able to achieve higher accuracy than the $Loss_{od}$ method on the object detection in the later training cycles as can be seen in Table 5.5. This is also the case for the $Incon_{seg}$ method which outperforms the object detection accuracy of the $Incon_{od}$ method at all cycles, shown in Table 5.3. In contrast to that stand, however, the performance of the inconsistency methods on the semantic segmentation, which is flipped as presented in Table 5.4. The generated results do not lead to a clear conclusion on whether a specific task should be focused more on compared to the other task. But they definitely show that a combination of both tasks leads to an increased overall accuracy on both tasks. The $BoxMask$ approaches use the gained information from both tasks which is leading to a strong performance already. The combination of multiple methods, however, results in the overall best accuracy on both tasks. One can thus conclude, that single-task focused methods can boost the performance of the other task if alternated with a different task-focused method.

## 6.6 Unsuccessful Experiments

One experiment investigated if a good performance of a method at a given training cycle is related to the method itself or the amount of the available data. The intuition was that a specific method could always outperform other methods at a certain amount of data, but be worse in other cycles. To validate this thought, an alternating training was started, which always used the selection strategy of the method with the highest mAP at this specific cycle. However, this resulted in very poor results. This indicates that the success of a selection strategy is highly related to its previous selection and that various methods can not be combined arbitrarily. Another idea was to alternate the methods that have the highest mAP increase at the respective training cycle, compared to the previous cycle. At the time of starting this experiment, the resulting alternation schema was a combination of the loss prediction module, the least confidence, and the 1-vs-2 margin sampling method. The performance on the semantic segmentation task was on-par with random selection, but on the object detection, this approach performed much worse compared to random selection. This again supports the conclusion made earlier, that methods cannot be combined based on their performance and should rather be combined by other indications like dataset correlation or their focused task.

# Chapter 7

# Conclusion & Future Work

The extensive analysis and evaluation of existing methods showed, that active learning can have a beneficial impact on the overall accuracy of a multi-task network on both trained tasks. In addition to that, a novel methodology has been developed which combined the domain knowledge from both tasks and the experiments showed that with that combined knowledge the accuracy of both tasks can be improved. All that while keeping the annotation costs lower than the traditional random selection of samples. Even though the accuracy of a fully trained model could not be reached with the active learning approaches, they achieve a large fraction of the accuracy at a much earlier time and with much less consumed data. Not only the sample selection strategies have been investigated in this thesis but also a novel training schema. Alternating multiple methods and combining the two tasks at the checkpoint selection pushed the accuracy even higher. Although a similar and more favourable accuracy could be achieved as that of the full data training, its accuracy was unfortunately not surpassed. The question of why this is the case remains for future work. Various possibilities come into question. The most likely reason is that during the full data training the learning rate was continuously reduced. This was not the case during the active learning cycles, as the learning rate was kept constant throughout the training. Also, other hyper-parameters, such as the number of iterations, were not optimised, which would certainly lead to a further improvement in accuracy. Especially on the two less studied datasets A9 and Cityscapes, it would be interesting to see to what extent the accuracy could be improved if the data split were adjusted to the respective dataset size. In the experiments with the NuImages dataset, it has been shown that an initial pool of 30% randomly selected data yields the best results. However, the NuImages dataset has already been curated using Active Learning methods, so a different split might work better for the A9 and Cityscapes datasets. Unfortunately, in the time frame of this thesis, it was not possible to try out any number of combinations of the sample selection strategies presented. An extensive investigation in this respect would certainly be interesting. And the scheme of alternating the methods could also have more potential. For example, instead of changing the method in every cycle, one could only alternate them every second cycle. Another idea would be to use two methods in one cycle and give the highest rated 50% of the samples from both methods to the annotation. Finally, it should be mentioned that the architecture used offers a good trade-off between accuracy and speed. However, it would be interesting to see what impact the presented methods have on other detectors and if the state-of-the-art performance can be further improved by using the findings presented here.

# Appendix A

# Appendix 1

## A.1 List of Acronyms

**BetG** Bet Gradient . . . . . 8

**DNN** Deep Neural Network . . . . . iv

**GAN** Generative Adversarial Network . . . . . 5

**IoU** Intersection over Union . . . . . 25

**KL-divergence** Kullback–Leibler-divergence . . . . . 8

**K-NN** K-Nearest Neighbours . . . . . 10

**LiDAR** Light Detection And Ranging . . . . . 22

**mAP** Mean Average Precision . . . . . 12

**MC Dropout** Monte-Carlo Dropout . . . . . 6

**mIU** Mean Intersection over Union . . . . . 10

**MSAC** Multi-Scale Atrous Convolution . . . . . 12

**MSE** Mean Squared Error . . . . . 16

**MTAL** Multi-Task Active Learning . . . . . 11

# Bibliography


[Aga+20] Agarwal, S., Arora, H., Anand, S., and Arora, C. “Contextual Diversity for Active Learning”. In: (Aug. 2020). URL: http://arxiv.org/abs/2008.05723.

[Agh+19] Aghdam, H. H., Gonzalez-Garcia, A., Weijer, J. van de, and López, A. M. “Active Learning for Deep Detection Neural Networks”. In: (Nov. 2019). URL: http://arxiv.org/abs/1911.09168.

[Bel+21] Belharbi, S., Ben Ayed, I., McCaffrey, L., and Granger, E. “Deep Active Learning for Joint Classification & Segmentation With Weak Annotator”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 3338–3347.

[Bel+18] Beluch, W. H., Genewein, T., Nurnberger, A., and Kohler, J. M. “The Power of Ensembles for Active Learning in Image Classification”. In: IEEE, June 2018, pp. 9368–9377. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00976. URL: https://ieeexplore.ieee.org/document/8579074/.

[Ben+21] Bengar, J. Z., Weijer, J. van de, Twardowski, B., and Raducanu, B. “Reducing Label Effort: Self-Supervised meets Active Learning”. In: (Aug. 2021). URL: http://arxiv.org/abs/2108.11458.

[BKD19] Brust, C. A., Käding, C., and Denzler, J. “Active learning for deep object detection”. In: *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* 5 (2019), pp. 181–190. DOI: 10.5220/0007248601810190.

[Cae+19] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. “nuScenes: A multimodal dataset for autonomous driving”. In: *arXiv e-prints*, arXiv:1903.11027 (Mar. 2019), arXiv:1903.11027. arXiv: 1903.11027 [cs.LG].

[Cai+21] Cai, L., Xu, X., Liew, J. H., and Foo, C. S. “Revisiting Superpixels for Active Learning in Semantic Segmentation with Realistic Annotation Costs”. In: (2021), pp. 10988–10997.

[CPL18] Cao, J., Pang, Y., and Li, X. *Triply Supervised Decoder Networks for Joint Detection and Segmentation*. 2018. DOI: 10.48550/ARXIV.1809.09299. URL: https://arxiv.org/abs/1809.09299.

[Che+18] Chen, L., Yang, Z., Ma, J., and Luo, Z. “Driving Scene Perception Network: Real-Time Joint Detection, Depth Estimation and Semantic Segmentation”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2018. DOI: 10.1109/wacv.2018.00145. URL: https://doi.org/10.1109%5C%2Fwacv.2018.00145.

[Cho+21] Choi, J., Elezi, I., Lee, H.-J., Farabet, C., and Alvarez, J. M. “Active Learning for Deep Object Detection via Probabilistic Modeling”. In: (Mar. 2021). URL: http://arxiv.org/abs/2103.16130.

[Col+21] Colling, P., Roese-Koerner, L., Gottschalk, H., and Rottmann, M. "MetaBox+: A new region based active learning method for semantic segmentation using priority maps". In: Oct. 2021, pp. 51–62. ISBN: 9789897584862. DOI: 10.5220/0010227500510062.

[Cor+16] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Cre+22] Creß, C., Zimmer, W., Strand, L., Fortkord, M., Dai, S., Lakshminarasimhan, V., and Knoll, A. "A9-Dataset: Multi-Sensor Infrastructure-Based Dataset for Mobility Research". In: 2022. URL: https://github.com/providentia-project/a9-dev-.

[Den+09] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[DB20] Desai, S. V. and Balasubramanian, V. N. "Towards Fine-grained Sampling for Active Learning in Object Detection". In: IEEE, June 2020, pp. 4010–4014. ISBN: 978-1-7281-9360-1. DOI: 10.1109/CVPRW50498.2020.00470. URL: https://ieeexplore.ieee.org/document/9151048/.

[Dvo+17] Dvornik, N., Shmelkov, K., Mairal, J., and Schmid, C. *BlitzNet: A Real-Time Deep Network for Scene Understanding*. 2017. DOI: 10.48550/ARXIV.1708.02813. URL: https://arxiv.org/abs/1708.02813.

[Ele+21a] Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixe, L., and Alvarez, J. M. "Towards Reducing Labeling Cost in Deep Object Detection". In: (2021). URL: http://arxiv.org/abs/2106.11921.

[Ele+21b] Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixe, L., and Alvarez, J. M. "Not All Labels Are Equal: Rationalizing The Labeling Costs for Training Object Detection". In: (June 2021). URL: http://arxiv.org/abs/2106.11921.

[Eve+15] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111 (1 Jan. 2015), pp. 98–136. ISSN: 0920-5691. DOI: 10.1007/s11263-014-0733-5. URL: http://link.springer.com/10.1007/s11263-014-0733-5.

[Fen+21] Feng, D., Harakeh, A., Waslander, S. L., and Dietmayer, K. "A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving". In: *IEEE Transactions on Intelligent Transportation Systems* (Nov. 2021), pp. 1–20. ISSN: 1524-9050. DOI: 10.1109/tits.2021.3096854.

[Fre65] Freeman, L. C. *Elementary Applied Statistics: For Students in Behavioral Science*. John Wiley & Sons, Dec. 1965.

[Fri22] Friedrich, P. *MTAL for Object Detection and Semantic Segmentation*. https://gitlab.lrz.de/ge97jum/thesis. 2022.

[GG15] Gal, Y. and Ghahramani, Z. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: (June 2015). URL: http://arxiv.org/abs/1506.02142.

[GIG17] Gal, Y., Islam, R., and Ghahramani, Z. "Deep Bayesian Active Learning with Image Data". In: (Mar. 2017). URL: http://arxiv.org/abs/1703.02910.

[GLU12] Geiger, A., Lenz, P., and Urtasun, R. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[HS90] Hansen, L. K. and Salamon, P. "Neural Network Ensembles". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (10 1990), pp. 993–1001. ISSN: 01628828. DOI: 10.1109/34.58871.

[Hau+20] Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., and Alvarez, J. M. "Scalable Active Learning for Object Detection". In: Apr. 2020, pp. 1430–1435. DOI: 10.1109/IV47402.2020.9304793.

[He+15] He, K., Zhang, X., Ren, S., and Sun, J. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[Hou+11] Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. "Bayesian Active Learning for Classification and Preference Learning". In: (Dec. 2011). URL: http://arxiv.org/abs/1112.5745.

[Ikh+18] Ikhwantri, F., Louvan, S., Kurniawan, K., Abisena, B., Rachman, V., Wicaksono, A. F., and Mahendra, R. "Multi-Task Active Learning for Neural Semantic Role Labeling on Low Resource Conversational Corpus". In: Association for Computational Linguistics, 2018, pp. 43–50. DOI: 10.18653/v1/W18-3406. URL: http://aclweb.org/anthology/W18-3406.

[Kao+19] Kao, C. C., Lee, T. Y., Sen, P., and Liu, M. Y. "Localization-Aware Active Learning for Object Detection". In: vol. 11366 LNCS. Jan. 2019, pp. 506–522. ISBN: 9783030208752. DOI: 10.1007/978-3-030-20876-9_32.

[Kas+19] Kasarla, T., Nagendar, G., Hegde, G. M., Balasubramanian, V., and Jawahar, C. V. "Region-based active learning for efficient labeling in semantic segmentation". In: vol. 2019-January. Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 1109–1117. ISBN: 9781728119755. DOI: 10.1109/WACV.2019.00123.

[Li+21] Li, Y., Fan, B., Zhang, W., Ding, W., and Yin, J. "Deep active learning for object detection". In: *Information Sciences* 579 (Nov. 2021), pp. 418–433. ISSN: 00200255. DOI: 10.1016/j.ins.2021.08.019.

[Liu+21] Liu, Z., Ding, H., Zhong, H., Li, W., Dai, J., and He, C. "Influence Selection for Active Learning". In: *arXiv e-prints* (Aug. 2021), arXiv:2108.09331.

[Mac+19] Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., and Rother, C. "CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation". In: Oct. 2019. URL: http://arxiv.org/abs/1810.09726.

[Mah+18] Mahapatra, D., Bozorgtabar, B., Thiran, J.-P., and Reyes, M. "Efficient Active Learning for Image Classification and Segmentation using a Sample Selection and Conditional Generative Adversarial Network". In: (June 2018). URL: http://arxiv.org/abs/1806.05473.

[MT18] Mayer, C. and Timofte, R. "Adversarial Sampling for Active Learning". In: (Aug. 2018). URL: http://arxiv.org/abs/1808.06671.

[Mon21] Monarch, R. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Manning, 2021. ISBN: 9781617296741. URL: https://books.google.de/books?id=LCh0zQEACAAJ.

[Pen+20] Peng, J., Nan, Z., Xu, L., Xin, J., and Zheng, N. “A Deep Model for Joint Object Detection and Semantic Segmentation in Traffic Scenes”. In: IEEE, July 2020, pp. 1–8. ISBN: 978-1-7281-6926-2. DOI: 10.1109/IJCNN48605.2020.9206883. URL: https://ieeexplore.ieee.org/document/9206883/.

[Rei+08] Reichart, R., Tomanek, K., Hahn, U., and Rappoport, A. “Multi-Task Active Learning for Linguistic Annotations”. In: Association for Computational Linguistics, 2008, pp. 861–869. URL: https://aclanthology.org/P08-1098.

[RUN18] Roy, S., Unmesh, A., and Namboodiri, V. P. “Deep active learning for object detection”. In: *BMVC*. 2018.

[Sal21] Salscheider, N. *Video-based Environment Perception for Automated Driving Using Deep Neural Networks*. Karlsruher Institut für Technologie (KIT), 2021. URL: https://books.google.de/books?id=mrSszgEACAAJ.

[Sal19] Salscheider, N. O. “Simultaneous Object Detection and Semantic Segmentation”. In: (May 2019).

[Sal20] Salscheider, N. O. *Training code for "Simultaneous Object Detection and Semantic Segmentation"*. https://github.com/fzi-forschungszentrum-informatik/NNAD/tree/icpram2020. 2020.

[Sch+20] Schmidt, S., Rao, Q., Tatsch, J., and Knoll, A. “Advanced Active Learning Strategies for Object Detection”. In: 2020, pp. 871–876. DOI: 10.1109/IV47402.2020.9304565.

[SS18] Sener, O. and Savarese, S. *Active Learning for Convolutional Neural Networks: A Core-Set Approach*. 2018. arXiv: 1708.00489 [stat.ML].

[Set10] Settles, B. “Active Learning Literature Survey”. In: *Machine Learning* 15 (2 2010), pp. 201–221. ISSN: 00483931. DOI: 10.1.1.167.4245.

[Sha48] Shannon, C. E. *A Mathematical Theory of Communication*. 1948.

[SVN19] Siddiqui, Y., Valentin, J., and Nießner, M. “ViewAL: Active Learning with Viewpoint Entropy for Semantic Segmentation”. In: (Nov. 2019). URL: http://arxiv.org/abs/1911.11789.

[Wu+19] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[WSH16] Wu, Z., Shen, C., and Hengel, A. van den. *Wider or Deeper: Revisiting the ResNet Model for Visual Recognition*. 2016. arXiv: 1611.10080 [cs.CV].

[Xie+20] Xie, S., Feng, Z., Chen, Y., Sun, S., Ma, C., and Song, M. “DEAL: Difficulty-aware Active Learning for Semantic Segmentation”. In: (Oct. 2020). URL: http://arxiv.org/abs/2010.08705.

[Yan+17] Yang, L., Zhang, Y., Chen, J., Zhang, S., and Chen, D. Z. “Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation”. In: (June 2017). URL: http://arxiv.org/abs/1706.04737.

[YK19] Yoo, D. and Kweon, I. S. “Learning Loss for Active Learning”. In: (May 2019). URL: http://arxiv.org/abs/1905.03677.

[Yua+21] Yuan, T., Wan, F., Fu, M., Liu, J., Xu, S., Ji, X., and Ye, Q. “Multiple instance active learning for object detection”. In: (Apr. 2021). URL: http://arxiv.org/abs/2104.02324.