Adapting Neural Networks to Complex Changing Visual Conditions with Minimal Supervision

Tilek Chubakov, Simona Aksman, Zuang Yu, Fangjun Yi, Mao Li
Advisors: Alberto Sangiovanni-Vincentelli and Xiangyu Yue

Contents

1. Introduction

Neural networks can achieve state-of-the-art performance on computer vision tasks when they have ample data and the training and deployment environments are similar [1–4]. However, in real-world scenarios (e.g. autonomous driving or medical diagnosis), labeled data for training is often limited and expensive to obtain [5]. In addition, the environments in which models are deployed can rapidly evolve in unexpected ways. In such complex changing environments, existing neural network models can fail to achieve adequate predictive performance.

To meet these real-world challenges, we design domain adaptation algorithms that aim to better generalize neural networks to previously-unseen visual settings, given a minimal amount of data and human supervision. Our methods help neural networks identify informative unseen data for labeling and adapt to new conditions in real time. In particular, we extend both semi-supervised (active) and unsupervised (continual, continuous) domain adaptation methods to the task of semantic segmentation, by which images are automatically labeled with classes at the pixel level. Semantic segmentation is a critical task for autonomous driving and therefore currently an important and highly active area of research in computer vision. Finally, we propose a novel combined method, active continual continuous domain adaptation (ACCDA), that enables neural networks to learn on the fly within continuously changing visual settings. Our work therefore improves upon the training efficiency and scalability of autonomous driving models.

In the following section we conduct a literature review that provides context for our work. Then, in section 3, we discuss different methodologies we make use of, and discuss how we contribute to and extend these methodologies. In section 4, we detail our experiments and results, and then finally, in section 5, we summarize our findings and discuss next steps for our work.

2. Context and Literature Review

2.1. Semantic Segmentation

Semantic segmentation is a computer vision task that classifies classes in images (i.e. objects, people, textures) at the pixel level. State-of-the-art algorithms for semantic segmentation use deep learning because of the high quality results that these algorithms have produced in recent years [7]. These models typically apply a method of scaling down, or down-sampling, input images to produce feature representations, and then apply a method to scale up, or up-sample, these feature representations and produce predictions at the original image's dimensions. See Figure 2 for a visual ex-

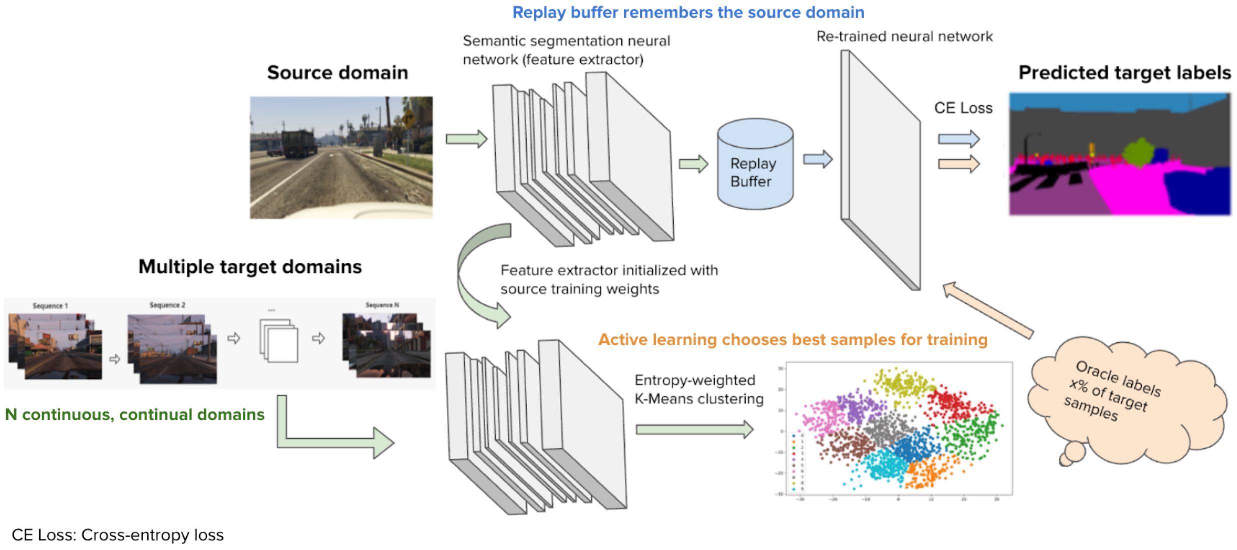

Replay buffer remembers the source domain
Semantic segmentation neural network (feature extractor)
Re-trained neural network
Source domain
Predicted target labels
CE Loss
Replay Buffer
Feature extractor initialized with source training weights
Multiple target domains
Sequence 1
Sequence 2
...
Sequence N
N continuous, continual domains
Active learning chooses best samples for training
Entropy-weighted K-Means clustering
Oracle labels x% of target samples
CE Loss: Cross-entropy loss

Figure 1. An illustration of our proposed Active Continuous Continual Domain Adaptation (ACCDA) algorithm.

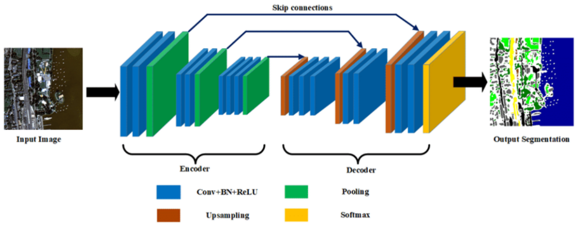


Figure 2. An example of an “encoder-decoder” semantic segmentation model which “encodes”, or down-samples, to compress the image to a set of features, and then “decodes”, or up-samples, to produce the pixel-wise output [6].

ample of this process. One of the most popular semantic segmentation models today is called DeepLabv3+ [8]. We use this model extensively in our work.

2.2. Active Learning

In active learning, a learning algorithm can interactively query a human (often referred to as an oracle) to obtain ground-truth labels [9]. This approach is useful when there is a substantial quantity of unlabeled data and the task of labeling that data is expensive. Such is the case for the task of semantic segmentation, where ground-truth labels are defined at the pixel level, and it can therefore take many hours to annotate a batch of labels for training.

State-of-the-art active learning tends to make use of either synthesized or pool-based methods for generating samples [10]. For instance, with a synthesized approach, generative adversarial networks (GANs) can be used to generate new, informative samples based on the sample distributions of existing data. Pool-based approaches, on the other hand, use representative samples from the unlabeled data and suggest them to an oracle for labeling. Pool-based approaches often use uncertainty and diversity cues to determine how best to sample data, with many recent approaches using both uncertainty and diversity cues [11]. We study several pool-based approaches in this paper, such as Active Adversarial Domain Adaptation (AADA) [12] and ADA-CLUE [11], in order to identify effective active learning strategies that can be applied to semantic segmentation. We provide further details about these methodologies in section 2.3.1.

2.3. Domain Adaptation

Domain adaptation (DA) is a technique which takes an algorithm trained on one domain, called the source domain, and optimizes its performance to a new target domain [13]. The target domain is typically unlabeled or partially labeled, and the target data distribution can be accessed during DA. Within the context of semantic segmentation, unsupervised DA is a popular approach as it does not require any target domain labels.

One application that we explore in our work is autonomous driving. To use supervised learning for this task, an extensive number of well-labelled data is typically required. In addition, data should span different environ-

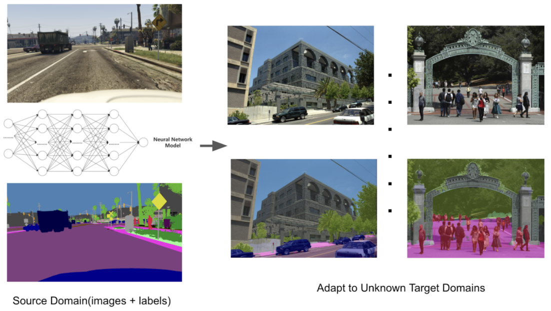

Neural Network Model
Source Domain(images + labels)
Adapt to Unknown Target Domains

Figure 3. Neural network trained on well-labeled source domains can be adapted to perform well on unseen target domains.

ments, weather conditions, and lighting conditions. The potential cost of collecting and labeling such data therefore makes supervised domain adaptation difficult to deploy and scale.

In comparison, unsupervised, or semi-supervised domain adaptation methods are aimed at quantifying the discrepancy between the distributions of a source domain and target domain, and maximally aligning features from the two domains with minimal external supervision.

One widely-used discrepancy measurement for unsupervised DA is maximum mean discrepancy (MMD). We implement it in our continuous DA model, and discuss technical details in section 3.4.

Other typical techniques include adversarial approaches using discriminative adversarial neural networks and self-training with pseudo-labels [14]. As one of essential baseline models in our work, AdaptSegNet [15] is a popular unsupervised DA algorithm which uses an adversarial approach. We provide more details about this approach in section 3.1.

2.3.1 Active Domain Adaptation

Recently, researchers have begun to combine active learning and domain adaptation for image classification, object detection, and semantic segmentation tasks [11, 12, 16]. This helps improve DA performance on challenging domain shifts and therefore allows DA to be more applicable within real-world settings. As was the case for DA, active DA approaches tend to follow one of two patterns: adversarial learning and pseudo-label cluster-based learning. Active DA has only recently been applied to the semantic segmentation task, so there is still much room for improvement. Our contribution to this area is to improve upon these active DA approaches for the task of semantic segmentation.

For instance, the first method to successfully apply active domain adaptation to the task of semantic segmentation is MADA, which utilizes a multi-anchor strategy to characterize source and target distributions in two stages [16]. We experiment with making improvements to MADA in section 3.3.

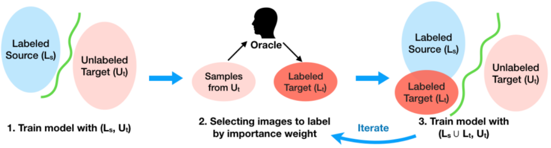


Figure 4. A visual representation of the active domain adaptation algorithm from [12]. The model selects a few target samples for an oracle (usually a human) to acquire labels.

2.3.2 Continuous Domain Adaptation

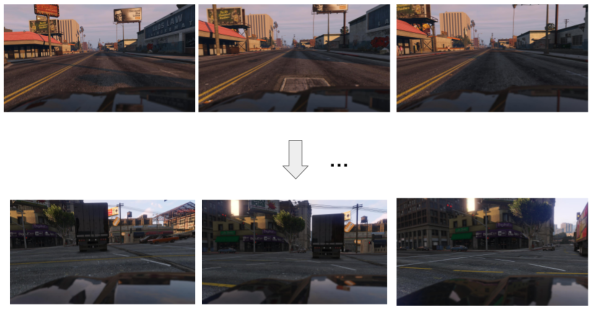


Figure 5. Significant differences caused by changes in lighting conditions. There is a large gap between the appearance of daytime and nighttime environments, but it is caused by small shifts and gradual changes.

In general, when the visual environment significantly changes, like from day to night, a model's performance degrades. However, large gaps usually come from the accumulation of continuous shifts, which can be shown in Fig. 5. We want our model to capture these small shifts by learning from seen tasks, and adapting to multiple and gradually changing target domains. In the case of learning from seen tasks, catastrophic forgetting [17] is a dominant problem for continuous domain adaptation. Traditional neural network structures have a tendency to overwrite past knowledge with the latest knowledge, which can lead to poor performance in a continual setting.

Therefore, there is a need for a buffer structure to replay previously-seen images. A replay buffer is a popular method in reinforcement learning to prevent forgetting, improve network's robustness by sampling, and boost calculation efficiency. There are usually two stages to a replay buffer - how to store experience and how to replay old knowledge. Accordingly, we present a novel replay buffer

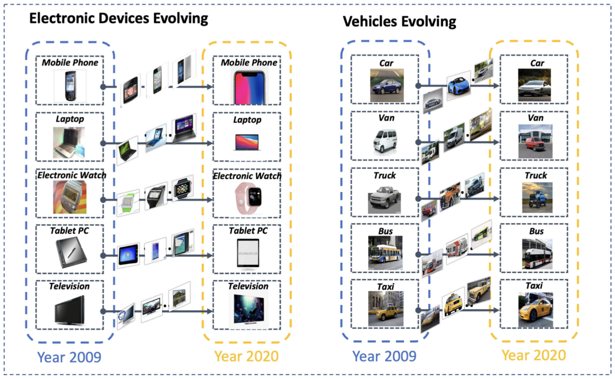


Figure 6. Illustration of two domains that gradually shift from source (images from the year 2009) to target (images from the year 2020) [18].

that is entropy-based and image-based to further enhance its performance. More details are discussed in section 3.4.

2.3.3 Active Gradual Domain Adaptation

One piece of research that is relevant to our proposed AC-CDA algorithm is an image classification algorithm is called Active Gradual Domain Adaptation (AGDA) [18]. This approach makes use of an uncertainty and diversity based active learning strategy to adapt to a set of gradually shifting visual environments, which are illustrated in Figure 6. This figure provides examples of images in the source and target datasets used for AGDA. It illustrates how different classes of objects, such as electronic devices and vehicles, evolve slowly over time between 2009 and 2020. The AGDA algorithm is able to outperform models trained only on data from the source domain as well as unsupervised domain adaptation models trained on the target domain. Performance of these other methods has been shown to degrade relative to AGDA as the distance between source and target domains grows.

2.4. Continual Learning

Continual learning, or lifelong learning, is a machine learning approach that aims to progressively adapt to new target domains by accumulating knowledge from previous experience [19–21]. Existing research on continual learning attempts to address the problem of catastrophic forgetting while adjusting to new tasks/domains. Another interesting approach is the dynamic expansion of neural network capacity to adapt to new tasks/domains [22]. In our work, we focused on using reinforcement learning methods to train policies for optimal selection of training samples to store in the replay buffer that can be used to to avoid catastrophic forgetting (details in section 3.4).

3. Methodology

3.1. AdaptSegNet

AdaptSegNet [15] is an adversarial DA approach which trains a fully-convolutional discriminator network **D** to learn the difference between source and target domains. The first step is to train the semantic segmentation network, which acts as the generator **G** within the adversarial learning setup. Next, **G**'s segmentation prediction is fed to **D**, which attempts to correctly classify whether it is coming from the source or target domain. The loss function for this joint learning process is:

$$L(I_s, I_t) = L_{seg}(I_s) + \lambda_{adv} L_{adv}(I_t) \tag{1}$$

where I_s and I_t are source and target images, respectively, L_{seg} is the cross-entropy loss learned by **G** in the source domain, and L_{adv} is the adversarial loss learned by **D**. λ_{adv} is a weight that balances the losses. Finally, the loss is optimized with a min-max objective:

$$\max_{\mathrm{D}} \min_{\mathrm{G}} L(I_s, I_t) \tag{2}$$

Given this objective, **G** will attempt to reduce it's loss with respect to its objective and improve predictions on the source images. At the same time, it will attempt to "fool" **D** into classifying target predictions as source predictions. This has the effect of reducing the gap between the source and target domains. We use AdaptSegNet as the first step for both our active domain adaptation and continuous, continual domain adaptation approaches.

3.2. Active Adversarial Domain Adaptation (AADA)

Since AADA and AdaptSegNet both include an adversarial unsupervised DA step, AdaptSegNet can be used as a starting point for implementing AADA [12]. The output of the adversarial network is an input to the active learning sample selection step, which aims to find the most informative labels for sampling. The sampling criterion $s(x)$ for choosing labels from the unlabeled target dataset is defined as:

$$s(x) = \frac{1 - G_d(G_f(x))}{G_d(G_f(x))} \mathcal{H}(G_y(G_f(x))) \tag{3}$$

where $G_d(G_f(x))$ and $G_y(G_f(x))$ are the predictions made by discriminator **D** and generator **G**, respectively, and $\mathcal{H}(G_y(G_f(x)))$ indicates the entropy of **G**'s predictions. In this sampling strategy, $\frac{1-G_d(G_f(x))}{G_d(G_f(x))}$ is the diversity cue, and the entropy term is the uncertainty cue. This sampling strategy is applied per batch, and assumes that $G_d(G_f(x))$ and $G_y(G_f(x))$ are scalar values. This is not the case for semantic segmentation, where **G** and **D** both produce pixel predictions at the original image's dimensions.

We discuss our experiments with extending AADA to the task of semantic segmentation in the next section where we discuss experiments and results.

3.3. Multi-Anchor Domain Adaptation (MADA)

MADA, an active domain adaptation method proposed in [16], outperforms other active domain adaptation methods for the task of semantic segmentation. It uses a multi-anchor strategy to efficiently capture the multi-model distribution of pixel-level features from both source and target and thus better identifies informative, diverse unlabeled samples than other methods. MADA was able to effectively adapt the ImageNet-pretrained DeepLab model from a synthesised dataset (GTA5) to a real-world dataset (Cityscapes), using very little training data. Like AADA, MADA also builds upon AdaptSegNet and extends the adversarial strategy.

In the first stage of MADA, it generates a feature map F^s of the source data x^s through adversarial domain adaptation and groups these features into clusters represented by centroid anchors A^s_k. The unlabeled target samples selected for label acquisition $D(x^t)$ are the ones closest to source centroids using the following distance calculation:

$$D(x^t) = \underset{k}{min}||F^t(x^t) - A^s_k||^2_2 \tag{4}$$

In the second stage of MADA, the segmentation model is fine-tuned with both source data and newly-labeled target samples. Pseudo labels are computed using cross-entropy loss, and then K-means clustering is performed to find centroid anchors for the remaining unlabeled target domain data. The final model objective function for semi-supervised training is thus a combination of segmentation loss L_{seg}, pseudo label cross-entropy L_{pseudo}, and the anchor representation score measured by distance L^t_{dis}:

$$L_{semi} = L_{seg} + L_{pseudo} + L^t_{dis} \tag{5}$$

We select MADA as our main algorithm for improving the performance of active domain adaptation for semantic segmentation. We experiment with making changes to the active learning sampling criteria used in MADA. Following the best practices of active learning theory and other similar successful active DA methods, such as AADA [12] and ADA-CLUE [11], we incorporate entropy, which would indicate the uncertainty of a model's predictions, into the strategy. For some intuition for why using both uncertainty and diversity is useful for active learning, see Figure 7. The entropy-only active learning strategy will select the most uncertain regions of the class distribution for labeling, however, it appears to over-sample these uncertain portions of the distribution. A diversity-only strategy, on the other hand, will select a set of labels that best spans the feature space, but is not picking up on the most highly-uncertain data points that the entropy strategy is able to find. The combination of these strategies best spans the distribution while also identifying the most highly-uncertain data points.

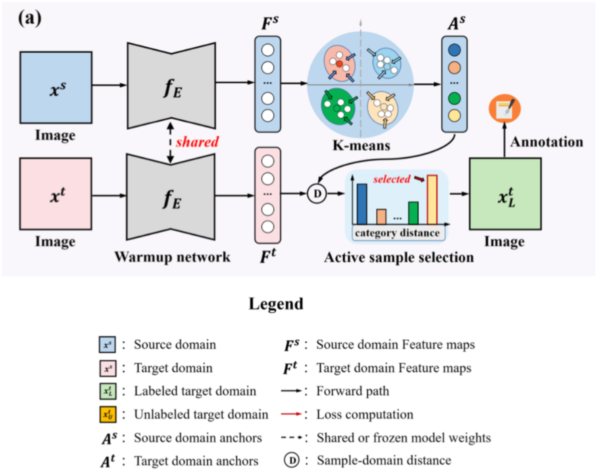

(a)
x^s
Image
f_E
shared
F^s
K-means
A^s
Annotation
x^t
Image
f_E
Warmup network
F^t
D
selected
category distance
Active sample selection
x^t_L
Image
Legend
x^s : Source domain
x^t : Target domain
x^t_L : Labeled target domain
x^t_U : Unlabeled target domain
A^s : Source domain anchors
A^t : Target domain anchors
F^s : Source domain Feature maps
F^t : Target domain Feature maps
: Forward path
: Loss computation
: Shared or frozen model weights
D : Sample-domain distance

Figure 7. A visual depiction of the first stage of the MADA algorithm [16]. This is the stage in which active learning sampling occurs.

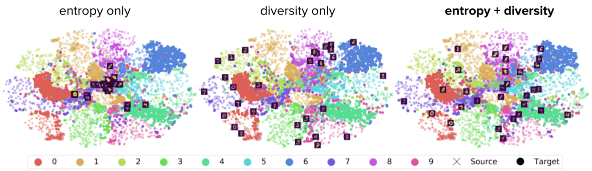


Figure 8. Three different active learning strategies, entropy only, diversity only and entropy + diversity, select different samples from the target data for an oracle to label [11].

Our final sampling criteria is as follows:

$$D(x^t) = \underset{S,k}{\text{argmax}} \sum_{k=1}^{K} \frac{1}{Z_k} \sum_{x \epsilon X_k} \mathcal{H}(F^t(x^t))||F^t(x^t) - A^t_k||^2 \tag{6}$$

where S is the set of target samples we select; A^t_k are the cluster centroids in the target domain; $\mathcal{H}(F^t(x^t))$, the entropy term, provides an estimate of the uncertainty of a segmentation model's predictions within the target domain; and Z_k is the sum of the entropy terms for each image, such that we calculate the mean entropy for each image.

3.4. Continuous, Continual Domain Adaptation

3.4.1 Sequential Unsupervised Adaptation

We propose an adaptation model based on the previous work [23] for multiple continuously shifting domains. Most existing domain adaptation methods assume a single source and a single target domain distribution. Usually, these methods rely on feature alignment between the source and target domains, or equivalently learning a mapping between distributions that minimize some distance measure. Some of the commonly used distance measures are: Kullback-Leibler divergence, Maximum Mean Discrepancy, correlation alignment, and adversarial loss.

After a mapping that minimizes the distance between the source and target domains is learned, the original semantic segmentation classifier for the source distribution can be applied directly to the target domain after feature alignment. Any standard classification loss function, such as cross-entropy loss, is used for the optimization task.

In the case of multiple continuously shifting target domains, running the alignment and adaptation procedure on a single bundled target domain results in poor performance. Thus, we explore adaptation schemas that sequentially adapt a segmentation model to multiple continuously (gradually) changing target domains. At each iteration, the current segmentation model will be adapted to the next target domain. Intuitively, the small changes in target distributions should make the feature alignment step easier.

In our work, we focus on various experiments using different distance functions between source and target distributions, as well as methods for selecting, storing, and replaying samples from previous domains to mitigate catastrophic forgetting.

3.4.2 Our replay buffer

During the adaptation stage, there are multiple target domains that a neural network needs to adapt to. As the adaptation stage progresses, the model will apply more weight to the most recently seen target domains, which may cause precision loss to domains it encountered previously due to catastrophic forgetting. Thus, to solve the problem of catastrophic forgetting, a **replay buffer** is introduced to our model. This replay buffer stores the previous target domain statistics, and is used to replay images from previous domains [24].

Since the replay buffer stores image statistics from different domains, we need to employ a sampling strategy when dealing with the replay buffer. **Random sampling** is one of the most common sampling strategies, but the result is not representative of the whole domain statistics stored in the replay buffer. Thus, we adapt a sampling method based on **entropy** loss values. Entropy loss value measures the degree of chaos. A lower entropy loss value indicates a sample with a lower degree of chaos. Recently, researchers tend to sample data with the lowest entropy loss values, because it is easy for models to train on the less chaotic data. Nonetheless, the less chaotic data cannot represent the whole dataset. Sampling the data with the lowest and highest entropy loss values, where 90% of the sampled result consists of least chaotic data and 10% of it consists of the most chaotic data, can better represent the whole dataset and improve the performance of our adaptation stage [25]. We experiment with applying the model with and without a replay buffer, and also experiment with different sampling methods in section 4.

One of our baseline models, Unsupervised Model Adaptation for Continual Semantic Segmentation, used a statistic-based replay buffer to store the previously learned knowledge [26]. Specifically, the unsupervised model adaptation used a Gaussian Mixture Model (GMM) to compute the means and co-variances learned so far. Although the memory requirement for the statistics-based replay buffer might be less demanding, we need to re-run the algorithm every time we perform the adaptation stage. Also, this algorithm is not compatible with our proposed sampling algorithm, so we need to convert the original statistics-based replay buffer.

To solve the problems specified above, we converted the original statistics-based replay buffer to a image-based replay buffer. To accomplish this, we removed the Gaussian Mixture Model (GMM), which is used to compute the means and co-variances. Instead, we directly read the original images in our source domain and put the images into our replay buffer. Every time we are in the adaptation stage, we can directly use the images in our replay buffer to feed into our proposed algorithm.

4. Experiments and Results

4.1. Extending AADA to semantic segmentation

Our first attempt at creating an active DA method that addresses the task of semantic segmentation was to extend the AADA method, which was originally designed for image classification tasks and uses adversarial unsupervised DA. In order to extend this method to semantic segmentation, we used AdaptSegNet, an adversarial unsupervised DA method designed for semantic segmentation [12,15]. In the original version of AADA, the active learning sampling criteria takes as input a set of scalar values that are produced by the generator and discriminator during the adversarial unsupervised DA step. However, in semantic segmentation, the outputs of the generator and discriminator in AdaptSegNet are now pixel maps. So our first task was to determine a method for compressing these values into scalars that could be used for active learning sampling. We also tested additional approaches for computing the sampling criteria. By making these changes, we hoped to make AADA more suitable to the semantic segmentation task. However, we found that we could not improve the performance of this method to exceed a random sampling strategy. Around this time, we also discovered that a novel method called MADA [16] had just been developed for semantic segmentation, and the au-

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bicycle	mIoU
Random	92.8	64.5	85.8	38.0	34.8	43.7	50.1	56.9	87.9	40.4	87.7	69.0	30.8	89.4	51.1	43.8	21.7	29.9	59.4	56.7
AADA	92.2	59.9	87.3	36.4	45.7	46.1	50.6	59.5	88.3	44.0	90.2	69.7	38.2	90.0	55.3	45.1	32.0	32.6	62.9	59.3
MADA	92.4	61.4	87.4	39.5	45.9	45.2	50.6	57.5	87.8	42.4	89.2	72.7	44.9	90.0	54.7	50.5	43.4	47.8	66.9	61.6
Our Proposed	93.4	64.9	86.7	39.0	47.8	45.4	50.0	56.7	87.3	44.0	89.9	72.0	44.6	90.4	55.8	54.9	44.0	46.0	65.5	62.3

Table 1. Results of different active domain adaptation methods on GTA5 → Cityscapes in terms of IoU across categories and the overall mean (mIoU). Each method was trained for 50 epochs, starting from the GTA5 dataset and adapting to the Cityscapes dataset.

thors of the MADA approach developed a version of AADA for semantic segmentation while working on the MADA approach. They were able to create a version of AADA that could outperform a random sampling strategy, but their proposed MADA approach performs even better. The first three rows of Table 1 detail these results in terms of IoU. Note that these results have been scaled to a range of 0-100, and therefore represent percentage values of IoU. All of these tests are conducted over 50 epochs using the GTA5 dataset as the source domain and the Cityscapes dataset as the target domain. In the next section we focus our attention on improving the MADA method given its stronger performance.

4.2. Improving active learning strategy in MADA

We experimented with several formulations of the active learning sampling strategy in MADA [16]. For instance, instead of using source cluster centroids A_k^s to determine the diversity cue, we used target cluster centroids A_k^t. We made this change because it more closely aligns with the design of other active learning sampling approaches that work well, such as ADA-CLUE, and is more appropriate for use in a continuous continual setting, where information from the target domain is more relevant than information from the source domain.

We also tried different ways of integrating the entropy cue $\mathcal{H}(F^t(x^t))$ into the active learning strategy. One method we tried was the approach used in ADA-CLUE, where an entropy weighting is applied during the K-means clustering step during which A_k^t is computed. Another approach is to run the K-means clustering step first, and then apply the entropy weight after. We experimented with both and found that the latter approach worked better. Table 1 provides our results for this latter approach, and compares our proposed approach to several benchmarks: the existing MADA method, a random sampling strategy, as well as the version of AADA that the authors of MADA adapted to the semantic segmentation task. In our testing, our source domain dataset is the GTA5 dataset and our target domain dataset is the Cityscapes dataset. We use these datasets in order to compare our work to the baseline method, MADA. All experiments are run over 50 training epochs with a 5% active learning sampling rate.

Our proposed method for active learning outperforms the existing MADA method by a small margin overall, achieving an mIoU of 62.3% as compared to the mIoU of 61.6% that MADA's active learning strategy achieves. In addition, our method seems to perform significantly better than MADA at identifying some specific categories of moving vehicles that can often appear unexpectedly on the road. For instance, it outperforms MADA at classifying buses, trucks, and trains. We plan to continue making improvements to this method to further improve performance.

4.3. Continuous, Continual Domain Adaptation

We next construct our continual and continuous domain adaptation model by using unsupervised learning and our replay buffer on the basis of [23, 26], and select AdaptSegNet [15] as our baseline model. For continuous shifts within the target domain, we choose 5000 images from the VIPER dataset that contains driving scenes with lighting conditions that gradually shift from afternoon to night. Therefore, models are expected to be adapted from the SYNTHIA dataset to the VIPER dataset. For baseline model - AdaptSegNet and our source model only trained in SYNTHIA, we input continuous target dataset as a single bundled batch. Fig. 9 shows samples of the prototypical distribution trained on SYNTHIA.

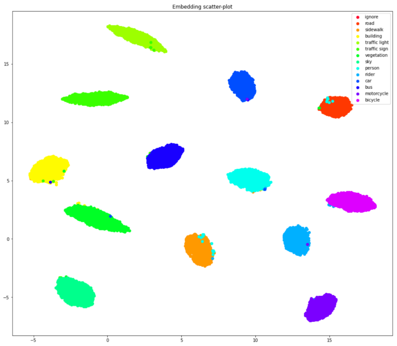


Figure 9. Samples distribution in embedding space.

We generate the embedding space in Fig. 9 using UMAP, a visualization method that maps high dimensional data to lower dimensions [27]. Each color corresponds to a different class. As discussed in 3.4.2, when training our adaptation model, we replace the Gaussian Mixture Model (GMM) with real images and store image samples in the replay buffer.

Moreover, we divide the target dataset into ten continual sequences and adapt to a single sequence in each iteration.

Method	road	sidewalk	building	traffic light	traffic sign	vegetation	sky	person	car	bus	mIoU
AdaptSegNet	36.3	24.3	32.2	9.4	1.4	27.49	68.24	4.0	27.23	16.4	25
Source-only	46.6	.09	47.5	.09	.01	13.3	73	19.7	86.3	.02	30
Unsupervised	83	38.6	55.4	15.3	0	26.3	84	14.5	85.5	0	41
ACCDA	89.3	57	65.6	55.4	12.1	36.3	91.8	36.8	98.1	.05	58

Table 2. Performance comparison on SYNTHIA → VIPER continuous, continual domain shift between a source-only model (without domain adaptation), an unsupervised continuous domain adaptation approach ("Unsupervised"), and our semi-supervised active continuous continual domain adaptation approach ("ACCDA") which samples 20% of target data per epoch for labeling, and uses a batch size per epoch of 20 samples. Results are presented in terms of IoU across categories and the overall mean (mIoU).

Features are pre-computed for the source domain and applied to distance calculations during training. Intersection over Union (IoU) is a statistical measurement for overlapping area between predictions and ground truth [28], ranging from 0–1 (0–100%).

Experiment results are shown in Table 2. AdaptSegNet [15] achieves 25% mean IoU (mIoU) in this case. Our model obtains 30% mIoU before adaptation and reaches 41% mIoU after adaptation, which significantly outperforms the baseline model.

4.4. Our novel method: ACCDA

Our proposed algorithm, which we are calling Active Continuous Continual Domain Adaptation or ACCDA, is similar to Active Gradual Domain Adaptation [18] in that it also uses an entropy and diversity based active learning strategy to learn a set of gradually shifting target domains. However, our approach also makes use of a replay buffer to remember the source domain, and our task is semantic segmentation rather than image classification. To our knowledge, this is the first active domain adaptation approach designed for continually-shifting domains for the task of semantic segmentation.

Figure 1 provides an illustration to help describe ACCDA. As you can see from this illustration, our algorithm contains three main components: the replay buffer which remembers the source domain; a continuous, continual series of target domains; and an active learning strategy which chooses the best samples for training from the target domain. Finally, the algorithm is re-trained with labels selected by active learning and data from our replay buffer in order to minimize cross entropy loss and produce semantic segmentation predictions.

The final final loss function for our ACCDA method is the following:

$$L_{ACCDA} = \lambda_1 L_{replay} + \lambda_2 L_{active} \tag{7}$$

$$L_{replay} = L_{CE}(f(x_s), y_s) \tag{8}$$

$$L_{active} = L_{CE}(f(x_t), y_t) \tag{9}$$

where L_{replay} and L_{active} represent the replay buffer loss and active learning loss, respectively; L_{CE} is the cross-entropy loss; $f(x_s)$ is a semantic segmentation prediction on source domain data; y_s are the source domain labels; $f(x_t)$ is a semantic segmentation prediction on target domain data; y_t are the target domain labels; and λ_1 and λ_2 are the loss function weights that trade-off between learning source data via the replay buffer and target data via active learning, respectively.

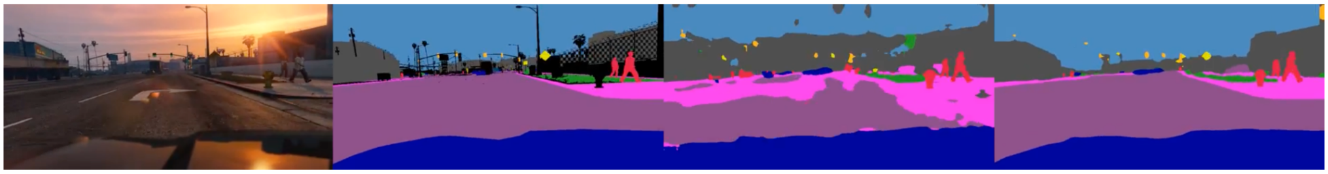
We found that tuning λ_2 for our active learning loss function helped to reduce the overall model's loss more quickly, which translated to improvements in our primary metric of interest, mean IoU, as illustrated in Figure 11. In addition to loss lambdas, we also tuned other hyperparameters, including the active learning rate and batch size for training on the target domain.

The results of applying our ACCDA method with a 20% active learning rate and a batch size of 20 samples for the SYNTHIA → VIPER domain shift are in Table 2. For this domain shift, ACCDA significantly improves model performance, from 41 to 58 mean IoU, when compared with an unsupervised method, which uses a replay buffer but does not have access to target labels. When compared to a model that was only trained on the source domain and does not include either a replay buffer or active learning, our approach improves mean IoU from 30 to 58.

While the ACCDA model improves performance across each category, it shows especially strong improvements for certain semantic classes that are important to pay attention for on the road, such as cars, people, and traffic lights. The performance for cars is particularly good, at 98 out of 100 mIoU. Note that buses did not actually appear in the training data, hence the low performance.

5. Conclusions and Next Steps

We have proposed ACCDA, a novel domain adaptation algorithm for the task of semantic segmentation. It combines active learning and a replay buffer to effectively learn given continuous, continual shifts in visual conditions. We have also demonstrated strong results for an autonomous driving use case, improving mean IoU from 30 to 58 on the



Original Image Ground Truth Source-only **ACCDA**

Figure 10. A visual comparison of predicted semantic segmentation labels produced by ACCDA and the Source-only method shows that our proposed approach greatly improves segmentation performance relative to a method that does not employ domain adaptation. It effectively captures key semantic classes that are important to pay attention for on the road, such as people, traffic lights, and traffic signs.

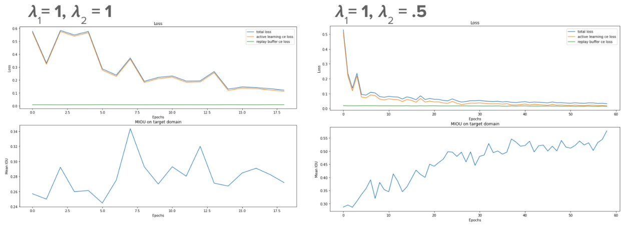


Figure 11. Example of improvements observed from tuning λ_2, the weight on the active learning loss function.

SYNTHIA → VIPER domain shift when compared with a source-only method, and 41 to 58 when compared to an unsupervised approach which uses a replay buffer. In addition, each sub-component of ACCDA, namely the replay buffer and active learning strategy, show improvements in performance compared with other state-of-the art methods.

We plan to further tune hyperparameters used in our ACCDA and conduct ablation studies to analyze the effects of these efforts. We will continue to tune the number of training epochs, learning rate, loss function weights, and batch size. We also plan to continue experimenting with methods for improving our active domain adaptation sampling strategy and will update our results accordingly. For instance, since the clusters we use for the diversity cue in our proposed strategy are easily influenced by dominant data points, our method may use a biased representation of the target domain. We therefore also plan to integrate a region-based heuristic to better capture the characteristics and spatial adjacency within an image region. This extension is largely inspired by the concept of region impurity and prediction uncertainty proposed by a recent related work [29]. By combining both pixel-level and region-level features, our active learning approach could further improve in performance relative to MADA.

6. Acknowledgements

Thank you to our research advisor, Alberto Sangiovanni-Vincentelli, and our research mentor, Xiangyu Yue, for motivating this work and for the many helpful suggestions and discussions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] D. Peng, Y. Zhang, and Guan, “End-to-end change detection for high resolution satellite images using improved unet++,” *Remote Sensing*, vol. 11, p. 1382, 06 2019.

[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, “A review on deep learning techniques applied to semantic segmentation,” *CoRR*, vol. abs/1704.06857, 2017. [Online]. Available: http://arxiv.org/abs/1704.06857

[8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016. [Online]. Available: http://arxiv.org/abs/1606.00915

[9] B. Settles, "Active learning literature survey," 2009.

[10] B. Zhang, L. Li, S. Yang, S. Wang, Z.-J. Zha, and Q. Huang, "State-relabeling adversarial active learning," 2020.

[11] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8505–8514.

[12] J. Su, Y. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, "Active adversarial domain adaptation," *CoRR*, vol. abs/1904.07848, 2019. [Online]. Available: http://arxiv.org/abs/1904.07848

[13] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to Unseen Domains: A Survey on Domain Generalization," *CoRR*, vol. abs/2103.03097, 2021.

[14] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," *CoRR*, vol. abs/2101.10979, 2021.

[15] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] M. Ning, D. Lu, D. Wei, C. Bian, C. Yuan, S. Yu, K. Ma, and Y. Zheng, "Multi-anchor active domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9112–9122.

[17] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions." *Psychological Review*, vol. 97, no. 2, p. 285–308, 1990.

[18] S. Zhou, L. Wang, S. Zhang, Z. Wang, and W. Zhu, "Active gradual domain adaptation: Dataset and approach," *IEEE Transactions on Multimedia*, vol. 24, pp. 1210–1220, 2022.

[19] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1611835114

[20] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.

[21] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *NIPS*, 2017.

[22] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *ArXiv*, vol. abs/1708.01547, 2018.

[23] A. Bobu, E. Tzeng, J. Hoffman, and T. Darrell, "Adapting to continuously shifting domains," in *ICLR*, 2018.

[24] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, and L. S. Davis, "Ace: Adapting to changing environments for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[25] B. Na, J. Mok, H. Choe, and S. Yoon, "Accelerating neural architecture search via proxy data," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 2848–2854, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2021/392

[26] S. Stan and M. Rostami, "Unsupervised model adaptation for continual semantic segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, pp. 2593–2601, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16362

[27] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[28] Wajih, A. Rosebrock, Anne, W. Ahmed, Kiran, Jsky, Abby, Aamer, Elsa, Miej, and et al., "Intersection over union (iou) for object detection," Jul 2021. [Online]. Available: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[29] B. Xie, L. Yuan, S. Li, C. H. Liu, and X. Cheng, "Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation," *arXiv preprint arXiv:2111.12940*, 2021.