Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation

Yang He[1,2], Shadi Rahimian[1], Bernt Schiele[2], and Mario Fritz[1]

[1] CISPA Helmholtz Center for Information Security
[2] Max Planck Institute for Informaticss
Saarland Informatics Campus, Germany
{yang.he, shadi.rahimian, fritz}@cispa.saarland, schiele@mpi-int.mpg.de

Abstract. Today's success of state of the art methods for semantic segmentation is driven by large datasets. Data is considered an important asset that needs to be protected, as the collection and annotation of such datasets comes at significant efforts and associated costs. In addition, visual data might contain private or sensitive information, that makes it equally unsuited for public release. Unfortunately, recent work on membership inference in the broader area of adversarial machine learning and inference attacks on machine learning models has shown that even black box classifiers leak information on the dataset that they were trained on. We show that such membership inference attacks can be successfully carried out on complex, state of the art models for semantic segmentation. In order to mitigate the associated risks, we also study a series of defenses against such membership inference attacks and find effective counter measures against the existing risks with little effect on the utility of the segmentation method. Finally, we extensively evaluate our attacks and defenses on a range of relevant real-world datasets: Cityscapes, BDD100K, and Mapillary Vistas. Our source code and demos are available at https://github.com/SSAW14/segmentation_membership_inference.

Keywords: Membership Inference; Data Privacy & Security; Forensics; Semantic Segmentation

1 Introduction

The availability of large datasets is playing a key role in today's state of the art computer vision methods ranging from image classification (e.g. ImageNet [7]), over semantic segmentation [6,21,35], to visual question answering [2]. Therefore, research and industry alike have recognized the importance of large-scale datasets [7,15,31,39] to push performance of computer vision algorithms. However, data collection and in particular annotation and curation of large datasets comes at a substantial cost. There are sizable efforts from the research community [6,11,35], and also industry has picked up the task of collection (e.g. [21])

as well as providing annotation services such as Amazon MTurk, which in turn can be monetized and constitutes important assets to companies.

Consequently, such assets need protection e.g. as part of intellectual property and it should be controlled which parts are made public (e.g. for research purposes) and which part remain private. Based on these datasets, high performing models are trained and then made public (e.g. as black box models) via an API or as part of a product. One might assume that the information of the training set remains contained within the trained parameters of the model and therefore remains private. Beyond the aspect of intellectual property, data might also include private information that were captured as part of the data collection process, which are sensitive and important for safe and clean services.

Unfortunately, recent work on membership inference attacks [26,27,29] has shown that even a black box model leaks information of the training data, *aiming to infer if a particular sample was used as part of the training data or not.* Such approaches have shown high success rates on a range of **classification** tasks and have equally proven to be hard to fully prevent (= defend). While this constitutes a potential threat to the machine learning model, it can also potentially be used as a forensics technique to detect a potentially unauthorized use of data.

However, we are still missing even a basic understanding on if and how these membership attack vectors extend to semantic segmentation, which is a basic computer vision task and has broad applications [4,13,16,17,37]. Hence, we propose and study first membership inference attacks and defenses for semantic segmentation. To reach this goal, we design an attack pipeline based on per-patch analysis, and discover (1) not all the areas of an input are helpful to membership inference, (2) structural information itself leaks membership privacy and (3) effective defense mechanisms exists that can reduce the effectiveness of these attacks substantially. Accordingly, we highlight our contributions to **segmentation** task and review relevant work.

1.1 Contributions

Our main contributions are as follows. (1) We present the first work on membership inference attacks against semantic segmentation models under different data/model assumptions. (2) We show structural outputs of segmentation have severe risks of leaking membership. Our proposed structured loss maps achieve the best attack results. (3) We present a range of defense methods to reduce membership leakage. In the end, we show feasible solutions to protect against membership attacks. (4) Extensive comparisons and ablation studies are provided in order to shed light on the core challenges of membership inference attacks for semantic segmentation.

1.2 Related Work

Recent attacks against machine learning models have drawn much attention to communities focusing on attacking model functionality (e.g., adversarial attacks [10,18,19,23,30,34]), or stealing functionality [24] or configurations [22] of a model. In this paper, we detail the topics of data privacy and security in the following.

Membership inference attack. Membership inference attacks have been successfully achieved in many problems and domains, varying from biomedical data [3], locations [25], purchasing records [27], and images [29].

It has been shown that machine learning models can be attacked to infer the membership status of their training data. Shokri et al. [29] proposed membership inference attacks against classification models utilizing multiple shadow models to mimic behaviors of the victim model. Shadow models were trained by querying the victim model using examples with higher confidences from the victim model. Hence, a binary classifier was trained with information from shadow models, and applied to attack the victim. Further, Salem et al. [27] demonstrated only one shadow model is enough to reach similar results rather than multiple shadow models. They also show that underlying distributions of data used to train shadow models and the victim can be different, which allows for attacks under relaxed assumptions. In addition, learning free attacks were proposed, which constitutes a low-skill attack without knowledge about the model and data distribution priors. Salem et al. [27] proposed to directly set a threshold on the confidence scores of predictions to recognize memberships. Sablayrolles et al. [26] set a threshold on loss values and achieved quite successful results. While prior work has only studied classification models so far, our contribution is to show the differences between segmentation and classification models and present the first study of attacks and defenses on semantic segmentation models based on new methods. Although the segmentation problem can be understood as pixelwise classification, it turns out the derived information is weak and needs to be aggregated over a patch or even the full image for a successful attack. Beyond this, we propose the first dedicated attacks that fully leverage the information of the full segmentation output and hence lead to even stronger attack vectors.

Privacy-preserving machine learning. The goal of these techniques is to reduce information leakage with limited access to training data, which have been applied to deep learning [1,28]. Differential privacy [9] allows learning the statistical properties of a dataset while preserving the privacy of the individual data points in it. Jayaraman et al. [14] discussed the connection between the effectiveness of differential privacy and membership inference in practice. Besides, Nasr et al. [20] provided membership protection for a classifier by training a coupled attacker in an adversary manner. Zhang et al. [36] obfuscated training data before feeding them to the model training task, which hides the statistical properties of an original dataset by adding random noises or providing new samples. In our work, we compare a series of defense approaches to mitigate membership leakage in semantic segmentation.

2 Attacks against Black-box Semantic Segmentation Models

Membership inference is to attack a **victim**, aiming to determine whether a particular data point was part of the training data of the victim. Such attacks exploit overfitting artifacts on training data [29,27,26]. Typical machine learning models tend to be overconfident on data points that were seen during the training. Such overfitting issues lead to characteristic patterns and distributions of confidence scores [29,27] or loss values [26] which has facilitated membership inference attacks. As a result, successful attacks against classification models can be achieved according to a **shadow** model trained by a malicious attacker, mimicking the overfitting patterns and distribution gaps.

We show how such attacks can equally be constructed against models for semantic segmentation with a specially designed pipeline and representations. While such models can be understood as pixel-wise classification, it turns out that the information that can be derived from a single pixel is rather weak. Hence, we develop a method that aggregates such information over patches and full images to arrive at stronger attacks. We first describe our pipeline for attacking segmentation models, and then present two attack settings exploited in our study, which have different constraints during attacks. Furthermore, we discuss our evaluation methodology, and then show evaluation results.

Algorithm 1 Training an attacker

```
Input: $\mathcal{D}^S = \{(\mathrm{X}_i, \mathrm{Y}_i)\}_i$, $\mathbf{V}$, $Epoch$
Output: Per-patch attacker $\mathbf{A_P}$
Query $\mathcal{D}^S$ with $\mathbf{V}$;
Partition $\mathcal{D}^S$ into $\mathcal{D}^S_{in}$, $\mathcal{D}^S_{out}$;
Train a shadow model $\mathbf{S}$ with $\mathcal{D}^S_{in}$;
Initialize $\mathbf{A_P}$;
for $i = 1; i \leq Epoch; i++$ do
    for $j = 1; j \leq |\mathcal{D}^S|; j++$ do
        Crop a patch $(\hat{\mathrm{X}}_j, \hat{\mathrm{Y}}_j)$ from $(\mathrm{X}_j, \mathrm{Y}_j)$;
        if $((\mathrm{X}_j, \mathrm{Y}_j) \in \mathcal{D}^S_{in})$ then
            $\mathbf{A_P}(\mathbf{S}(\hat{\mathrm{X}}_j), \hat{\mathrm{Y}}_j) \xrightarrow{\text{learn}} 1$
        else
            $\mathbf{A_P}(\mathbf{S}(\hat{\mathrm{X}}_j), \hat{\mathrm{Y}}_j) \xrightarrow{\text{learn}} 0$
        end if
    end for
end for
return $\mathbf{A_P}$;
```

Algorithm 2 Testing (Membership Inference)

```
Input: Testing pair $(\mathrm{X}, \mathrm{Y})$, $\mathbf{V}$, $\mathbf{A_P}$, $N$, $\tau$
Output: Image-level inference result $\mathbf{A}$
$\mathbf{A} = 0$; $i = 0$;
while $i < N$ do
    Crop $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$ from $(\mathrm{X}, \mathrm{Y})$; // patch selection
    if Mean$(\mathbf{V}(\hat{\mathrm{X}}) \otimes \hat{\mathrm{Y}}) > \tau)$ then
        continue; // reject too confident patches
    end if
    $\mathbf{A} = \mathbf{A} + \mathbf{A_P}(\mathbf{V}(\hat{\mathrm{X}}), \hat{\mathrm{Y}})/N$;
    i++;
end while
return $\mathbf{A}$;
```

2.1 Methods

Our approach infers image-level membership information based on observing predictions of segmentation models and correct labels. In this section, we describe our membership inference pipeline based on per-patch analysis, as summarized in Algorithm 1 and 2. Further, several design choices are discussed that significantly contribute to the success of the attack and help to understand the essence in attacking semantic segmentation models with structured outputs.

Notation. We define the notation used through the paper. Let $\mathcal{D}^{\{V,S\}} = \{(\mathrm{X}_i, \mathrm{Y}_i)\}_i$ be two datasets including images $\mathrm{X} \in \mathcal{R}^{H \times W \times 3}$ and densely annotated GTs $\mathrm{Y} \in \mathcal{R}^{H \times W \times C}$ with one-hot vectors, where C is the number of predefined labels. For each dataset, we partition it into two parts for proving different membership status, i.e., $\mathcal{D}^{\{V,S\}} = \mathcal{D}^{\{V,S\}}_{in} \cup \mathcal{D}^{\{V,S\}}_{out}$. The victim model, which is trained on $\mathcal{D}^{V}_{in}$ and we aim for attacking, is denoted as $\mathbf{V}$. To achieve attacks, we build a shadow semantic segmentation model $\mathbf{S}$ with $\mathcal{D}^{S}_{in}$, for training an attacker. Let P be the posterior of a segmentation output, i.e., $\mathrm{P} = \mathbf{S}(\mathrm{X})$ or $\mathbf{V}(\mathrm{X})$, depending on the stages of membership inference. Our per-patch attacker is denoted as $\mathbf{A_P}(\mathrm{P}, \mathrm{Y})$, taking P and Y as the inputs and outputs a binary classification score for membership status. Finally, the image-level attacker is denoted as $\mathbf{A}$.

Training. Our method is built upon a per-patch attacker $\mathbf{A_P}$, as described in Algorithm 1. In line with previous work on membership inference [29,27], we construct a shadow model $\mathbf{S}$ that is to some extent similar to $\mathbf{V}$ and therefore is expected to exhibit similar behaviour and artifacts w.r.t. membership. In addition, $\mathbf{S}$ aims to capture semantic relations and dependencies between different classes in structured outputs and provide training data to the patch classifier with known membership labels. We prepare a dataset $\mathcal{D}^{S}$ with the same label space to $\mathcal{D}^{V}$, and then $\mathbf{S}$ is trained on $\mathcal{D}^{V}_{in}$. The exact assumptions of our knowledge on $\mathbf{V}$ that inform the construction of $\mathbf{S}$ are detailed in 2.2.

Construction of per-patch attacker. $\mathbf{S}$ provides training data for the per-patch attacker $\mathbf{A_P}$, as we have complete membership information of $\mathbf{S}$. This allows us to train $\mathbf{A_P}$ by achieving the binary In/Out classification on the data pairs from $\mathcal{D}^{S}_{in}$ and $\mathcal{D}^{S}_{out}$. $\mathbf{A_P}$ can be any architecture taking image-like data as inputs, and we discuss different data representations to train it as follows.

Data representation. We apply two representations of a data pair (X, Y) over segmentation models, as the inputs of a per-patch attacker. In other words, we train a classifier to compare the differences between $\mathrm{P}{=}\mathbf{S}(\mathrm{X})$ and Y to determine the membership status of (X, Y) in training the classifiers.

1. *Concatenation.* We concatenate P and Y over the channel dimension, leading to a representation with size $H \times W \times 2C$.

2. *Structured loss map.* The structured loss map ($\mathrm{SLM} \in \mathbf{R}^{H \times W \times 1}$) computes the cross-entropy loss values at all the locations (i, j), where $\mathrm{SLM}(i, j) = -\sum_{c=1}^{C} \mathrm{Y}(i, j, c) \cdot \log(\mathrm{P}(i, j, c))$. Previous work [26] shows the success of applying a threshold on the loss value of an image pair for image classification, and this method can be easily applied to semantic segmentation. Despite this, we show keeping structures of loss maps is still crucial to the success of attacking semantic segmentation.

Testing. Given a data pair (X, Y) to determine if it was used to train $\mathbf{V}$ (i.e., $(\mathrm{X}, \mathrm{Y}) \in \mathcal{D}^{V}_{in}$), we are able to crop a patch $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$ from the pair, and thus the inference result is $\mathbf{A_P}(\mathbf{V}(\hat{\mathrm{X}}), \hat{\mathrm{Y}})$ according to the representation used in the training. In order to further amplify the attack, we aggregate the information of the per-patch attack on an image-level, therefore, the final inference result is calculated by

$$\mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A_P}(\mathbf{V}(\hat{\mathrm{X}}^i), \hat{\mathrm{Y}}^i), \tag{1}$$

where $(\hat{\mathrm{X}}^i, \hat{\mathrm{Y}}^i)$ is the i-th cropped patch from $(\hat{\mathrm{X}}, \hat{\mathrm{Y}})$.

Selection of patches. As our method is based on scoring each patch, the selection of patches plays an important role in obtaining stronger attacks. Besides, it also helps us to understand which patches are particularly important for determine the membership status of an example. Therefore, we study the influence of different patch selection schemes with the following choices:

1. *Sliding windows.* We crop patches on a regular grid with a fixed step size.
2. *Random locations.* We sample patches uniformly across the image.
3. *Random locations with rejection.* We emphasize the importance of different patches for recognizing membership is not alike, therefore, this scheme aims to reject patches, which do not contribute to final results or even provide misleading information. We observe the patches with very strong confidences or very small loss should be omitted. For example, road area counts for most pixels of an image and are segmented very well, therefore, this scheme tends to select the bordering areas between a road and other classes, instead of the center of a road.

As summarized in Algorithm 2, we construct image-level membership inference attacks according to per-patch attacks, allowing us to leverage distinct patches for successful attacks. Our pipeline is flexible to image sizes and aspect ratios if different image sizes exist in a dataset and even cross multiple datasets.

2.2 Attack Settings

In our method, we train a shadow segmentation model **S** and an attacker **A** for attacking a victim segmentation model **V**. Our two attack settings differ in the knowledge on data distribution and model selection for training **V** and **S**.

Data & model dependent attacks: This attack assumes that the victims model can be queried at training time of an attacker. Besides, this setting allows to train a shadow model with the same architecture to the victim. Specifically, **S** and **V** have the same learning protocol and post-processing techniques during inference. Further, this attack assumes the data distributions of $\mathcal{D}^V$ and $\mathcal{D}^S$ are also identical, which comes from the same database. Last, query with a victim model is allowed to split $\mathcal{D}^S$ into $\mathcal{D}^S_{in}$ and $\mathcal{D}^S_{out}$, as listed in the 1-st line of Algorithm 1, that we use the examples with stronger confidences to build $\mathcal{D}^S_{in}$.

Data & model independent attacks: For this attack, we only know the victim model's functionality and a defined label space. There is no query process for constructing training set for **S**, instead, **S** is able to be trained with a dataset of the different distribution, which leads to a cheaper and more practical attack. Furthermore, the model configuration and training protocol of the victim are unknown. The goal of the shadow model is to capture the membership status for each example, and provide training data for attack model **A**. Particularly, model and data distribution are completely different to victims, even there is no query process, which might be detected on the server. Therefore, we highlight the severity of information leakage in this simplified attack.

2.3 Evaluation Methodology

We evaluate the performance of membership inference attacks with **precision-recall** curves and receiver operating characteristic (**ROC**) curves. We regard the images used during training as positive examples, and negatives if not. Therefore, given a testing set with M image pairs used to train a model and N pairs not used, random guess with probability 0.5/0.5 for both classes is able to achieve precision $\frac{M}{M+N}$ and recall 0.5. We set different thresholds in a classifier and compare its precision-recall curve to the random guess performance, to observe if attacks are successful. Similarly, we draw the random guess behavior in a ROC curve, which is the diagonal of a plot. Furthermore, to compare different attacks quantitatively, we apply maximum **F-score** ($\frac{2\cdot\text{precison}\cdot\text{recall}}{\text{precison}+\text{recall}}$) in precision-recall curves and **AUC-score** in ROC curves to evaluate attack performance. Last, our method is based on per-patch attacks, therefore, we employ the same metrics for per-patch evaluation, to help us understand and compare different attacks, as well as defense methods in section 3, exhaustively.

2.4 Evaluation Results

Data and architectures. We conduct the experiments on street scene semantic segmentation between various datasets, including *Cityscapes* [6], *BDD100K* [35] and *Mapillary Vistas* [21], which are captured in different countries under diverse weathers and image qualities, providing multiple domains. Besides, we apply PSPNet [37], UperNet [33], Deeplab-v3+ [5] and DPC [4] as our segmentation models. For per-patch attacker $\mathbf{A_P}$, we train a ResNet-50 [12] from scratch, allowing us to visualize the regions contributing to the recognition of membership for an example by class activation mapping [38]. In detail, the size of inputs for ResNet-50 is 90×90 in spatial, corresponding to 713×713 image patches.

Comparison methods. To demonstrate the effectiveness of specific considerations for segmentation models, we compare our pipeline to previous attackers for classification models [26,27]. For [27], we adapt their shadow model based attacker, by regarding each location as a classification problem. We also test their learning-free attacker by only considering the mean of confidence scores of a prediction. Besides, we compare the proposed method with [26], which employs a threshold on the loss value.

Setup for data & model dependent attacks. For dependent attacks, we conduct experiments with *Cityscapes* and PSPNet (a.k.a. PSP→PSP). We split *Cityscapes* into four parts, i.e., $\mathcal{D}^V_{in}$, $\mathcal{D}^V_{out}$, $\mathcal{D}^S_{in}$ and $\mathcal{D}^V_{out}$, where the sizes of those sets are as follows: $|\mathcal{D}^V_{in}|$ =1488, $|\mathcal{D}^V_{out}|$ =912, $|\mathcal{D}^S_{in}|$ =555 and $|\mathcal{D}^S_{out}|$ =520. We train a victim model from ImageNet [7] pretrained models and lead to 59.88 mean IoU (mIoU) for segmentation. For evaluation of per-patch attacks, we sample 29760 patches from $\mathcal{D}^V_{in}$ and 30096 patches from $\mathcal{D}^V_{out}$. Therefore, this setting leads to 62% and 49.7% precision for image-level and per-patch attacks

Table 1: Data and model descriptions of victim and shadow models for independent attacks.

Dataset	Model	Backbone	In / Out
Cityscapes (Victim)	PSPNet [37] UperNet [33]	ResNet-101 [12]	2975 / 500
BDD100K (Shadow) *Mapillary* (Shadow)	Deeplab-v3+ [5] DPC [4]	Xception-71 [32]	4k / (3k+1k) 10k / (8k+2k)

Table 2: Comparison of different attackers (in %). We compare our attackers to previous methods, including the learning-based attacker [27]* and learning-free attackers by applying a threshold on a confidence score [27]+ or a loss value [26]. "→" means the attacks with a shadow model of the left, and the victims are the right, which can be PSPNet [37], UperNet [33], DPC [4] or Deeplab-v3+ [5].

Methods	Dependent		Independent							
	[37]→[37]		[5]→[37]		[5]→[33]		[4]→[37]		[4]→[33]	
	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC
Adapted [27]*	77.2	67.2	92.4	63.5	92.3	62.6	–	–	–	–
Adapted [27]+	77.4	62.0	92.3	63.4	92.3	59.2	92.3	63.4	92.3	59.2
Adapted [26]	82.2	74.9	94.4	81.4	93.0	72.4	**94.4**	**81.4**	93.0	72.4
Ours (C+GT, Full)	80.6	81.2	94.5	85.0	92.8	71.8	93.2	73.5	92.6	68.8
Ours (Loss, Full)	84.2	82.6	**95.7**	89.1	93.2	76.3	93.1	73.5	92.4	68.3
Ours (C+GT, Random)	83.4	82.7	95.0	86.1	95.4	88.5	92.9	74.9	**94.4**	**85.5**
Ours (Loss, Random)	**84.8**	**84.6**	**95.7**	**90.8**	**95.8**	**94.3**	94.0	77.7	93.3	79.4
Ours (C+GT, Rejection)	83.3	83.0	94.9	86.3	95.3	91.2	93.5	76.3	**94.4**	**86.1**
Ours (Loss, Rejection)	**86.7**	**87.1**	**95.9**	**91.1**	**96.2**	**94.9**	**94.1**	**77.8**	**93.5**	82.0

in random guess. The resulting F-scores for image-level and per-patch attacks are 55.36% and 49.85% respectively, which are drawn in Fig. 1.

Setup for data & model independent attacks. For independent attacks, we employ different segmentation models for shadow models and victims, as summarized in Table 1. Particularly, *BDD100K* has completely compatible label space to *Cityscapes* of 19 classes, but *Mapillary Vistas* has 65 labels. We merge the some classes from *Mapillary Vistas* into *Cityscapes*, and ignore the others. For victim models, we train a PSPNet and an UperNet using the official split of *Cityscapes*, leading to 79.7 and 76.6 mIoU for segmentation. For shadow models, we apply our splits with balanced In/Out distribution to train a binary classifier. In the end, the F-score of random guess for image-level independent attack is 63.13%. Comparing it to the numbers in Table 2, we observe all the attackers obtain much higher F-score than 63.13%, which shows the severe information leakage of semantic segmentation models.

Results. Results of the different versions of our model as well as comparision to previous work in presented in Table 2. While previous work on membership inference targets classification models [27,26], we facilitate a comparison to these approaches by extending them to the segmentation scenario. [27] proposes a learning-based attacker and a learning-free attacker. We train their learning-based attacker with 1×1 vector inputs, and test on all pixel locations. Final image-level attacks are obtained by averaging the binary classification scores of all locations. Similar to our method, we test different settings, and it fails to achieve attacks with the shadow model DPC [4] in Table 1. Besides, we test their learning-free attacker by averaging the confidence scores of all locations. Equally,

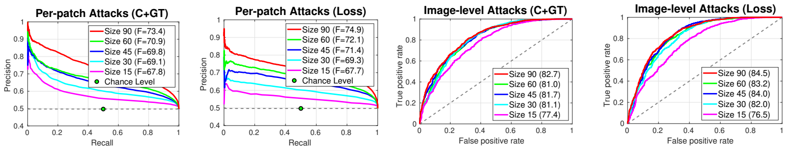


Fig. 1: Evaluation of the **importance of spatial structures** for PSP→PSP, starting from our final model (Size 90).

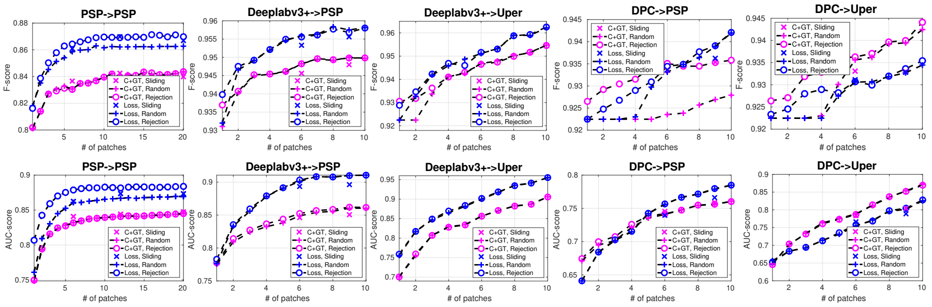


Fig. 2: Image-level comparison results w.r.t. **patch selection** and **data representations**, under varying patch numbers.

we facilitate a comparison to [26] where we use the loss map for the segmentation output. For our methods, we report the numbers for last two patch selection strategies with sampling 10 patches. Besides, we also perform attacks with full image inputs using our binary classifiers, which have a global average pooling in the end and are able to handle different sizes of inputs. We emphasize that the ratio of In/Out testing examples are different for dependent and independent attacks, therefore, the numbers between them cannot be compared. We conclude that recent models for semantic segmentation are susceptible to membership inference attacks with AUC scores of the attacker up to 87.1% in the dependent and 94.9% in the independent setting. Overall, we observe that our loss-based method with rejection scheme performs best in most settings and measures.

Importance of spatial structures. Key to strong attacks is exploiting the structural information of an output from a segmentation model. Hence, we conduct attacks with gradually reduced structural inputs in our dependent attacks in order to analyze the importance of this structural information for our goal. Our final model takes 90×90 blocks as inputs for per-patch attackers. Therefore, we crop sub-blocks from our final model with input sizes of 60, 45, 30, 15 for providing different level of structures. We compare the precision-recall curves for per-patch attacks and ROC curves for image-level attacks in Fig. 1. We note that all the feature vectors in the blocks of different sizes have the same scale of receptive fields. We apply the same architecture of per-patch attacker for sizes 90-30, but modify the ResNet-50 with fewer pooling operations for size 15, because its spatial size is too small. First, we compare the per-patch attack performance, and are able to observe that attacks become harder with decreasing patch sizes, where smaller patches provide less structures. Second, we compare image-level attacks for them, where random selection strategy is applied to integrate all patches. We sample 5, 20, 20, 30, 30 random patches for size 90, 60, 45, 30, and 15 to integrate image-level results. Consequently, size 90 achieves the best performance, even though other attackers obtain very close image-level results. Last, we highlight that our concatenation-based attacker degenerates to previous work [27] with 1×1 vector inputs. We observe that 1×1 inputs keep this decreasing trend and achieve worse results than size 15, which can be found in Table 2. From this results, we conclude that structures are of great importance in membership inference attacks for semantic segmentation, so that an attacker is able to mine some In/Out confidence or loss patterns over an array input.

Analysis of patch selection and data representation. We test our three sampling strategies and two representations as depicted in section 2.1. Fig. 2 plots the image-level comparison results. For sliding windows, we sample at least 6 patches to guarantee an entire image can be covered. For random locations, we sample different numbers of patches for image-level attacks to observe the influence of patch numbers, starting from one patch. We conduct this experiments for 3 times and report the mean. In summary, we observe these two strategies achieves comparable performance when the same numbers of patches are used. Specifically, sliding windows perform better on dependent attacks with loss maps, and random locations are better for independent attacks (Deeplab-v3+ →PSP, and Deeplab-v3+ →Uper), which may be caused by inconsistent data distributions or different behaviors of segmentation models. Last, we test our random locations with rejection strategy. To avoid the affect of random seeds, we sample the same locations to previous random locations if a patch is not rejected. We can see clear improvements if we sample very few patches, whose results are sensitive to sampled locations. In street scenes, road has a large portion of pixels, therefore, it tend to sample a road patch, which has the highest accuracy over all the classes and less discrimination for In/Out classification. After ignoring those patches, performance is improved because the rejection helps us avoid those less informative patches. To conclude, not all the regions contribute to successful attacks for segmentation, that we need a regime to determine membership status of an image, instead of processing the whole like previous work for classification.

Comparing our patch-based attacks to the full image attacks, we realize using full images as inputs makes performance significantly decreased, even though the same classifier is applied. The classification for full images may be affected by misleading areas. Hence, partitioning an image into many patches helps focus on local patterns and makes a better decision. Besides, we observe that our rejection scheme achieves better performance than random scheme, which further supports our argument on the difference between segmentation and classification. In addition, our concatenation-based attacker outperforms [27], which demonstrates the importance of spatial structures, similar to Fig. 1. From our results, [26] is able to obtain acceptable performance but worse than our structured loss map-based attackers, which hold the structural information. Finally, our novel structured loss maps achieve better results than concatenation and other methods [27,26] in most cases.

3 Defenses

To mitigate the membership leakage and protect the authority of a model, we study several defenses for semantic segmentation models, while maintaining their utility with little performance degeneration. When a model is deployed, a service provider has all rights to access the model and data. Our work shows for the first time a feasible solution for protecting very large semantic segmentation model. As a consequence, we manipulate the model in training or testing stages by reducing the distribution gaps between training data and others w.r.t. confidence scores of predictions or loss values, including Argmax, Gauss, Dropout and DPSGD. The first two methods can be applied in any segmentation models and last two can be applied in deep neural networks.

Settings. We analyze the performance of image-level attacks according to random locations in this section, which are easily compared to the results without defenses in Table 2. Because Gaussian noises, or dropout will change output distributions, rejection scheme may sample different patches, we only test the random location schemes and sample patches at the same locations for different defenses, and keep consistent to previous attacks.

For dependent attacks, our shadow and victim models have the same post-processing and learning protocol, as claimed in section 2.2. In other words, we employ the same defenses and strength factors for them in this setting. For independent attacks, we report the settings of Deeplab-v3+ →PSP and Deeplab-v3+ →Uper, which are the most successful. We only employ defenses on victim models as protections for released black-box semantic segmentation models.

Evaluation methodology. Due to the different ratios for In/Out examples in various settings, we do not report their F-scores in this section. Instead, we only employ AUC-score to compare different defense methods, expecting to reduce the original attacks' AUC-scores to 0.5, that random guess in all the settings hold this number. Furthermore, an ideal defense is supposed to make attacks hard and preserve segmentation utility at the same time. Therefore, we apply mIoU [17] to evaluate the segmentation performance and jointly compare different defense methods w.r.t. capability of membership protection and utility of segmentation.

3.1 Methods and Results

Argmax. It only returns predicted labels instead of posteriors for an image. We use one-hot vectors to complete attacks for our methods and others [27,26]. Obviously, previous learning-free attacker [27] based on confidence scores fails to recognize membership states, because every example has confidence 1. In Fig. 3, we show the comparison results for all the other methods. Because argmax is very easy to be noticed, we train binary classifiers for independent attacks with argmax operation as well. In general, argmax only reduces membership leakage in segmentation models a little for all the attackers. A model already leaks information when it only returns predicted labels. To conclude, we highlight the difference to protecting classification, that argmax cannot successfully protect the membership privacy for segmentation.

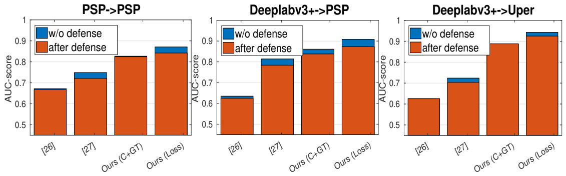


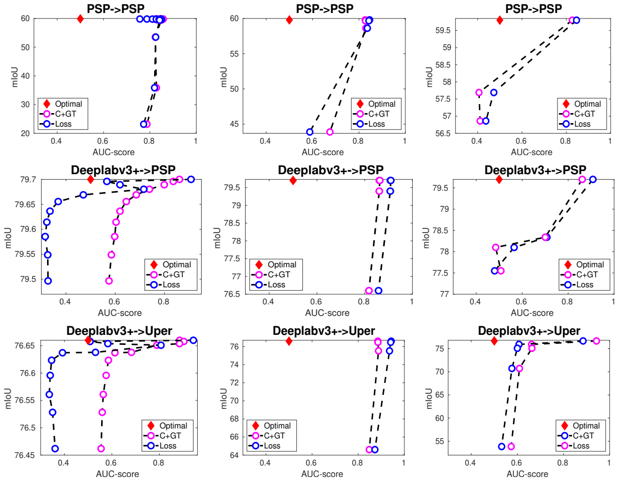
Fig. 3: Performance comparison for Argmax defense.

Gauss. To hide overfitting artifacts or patterns, we add Gaussian noises on the posteriors with different variances, varying from 0.01 to 0.1 with step 0.01 for independent attacks. To further test the defense for dependent attacks, we add very strong noises to variance 0.4. After noising, we set the values into 0 in case they are smaller than 0, and then normalize each location individually. Segmentation performance is decreased with stronger noises, therefore, we show the joint privacy-segmentation plots in Fig. 4(a) to observe the defense behaviors as well as the maintained utility of the segmentation method. First, we observe Gauss protects PSPNet and UperNet in independent attacks successfully, which reduces AUC-scores from 0.9 to less than 0.6, while only losing 0.2 mIoU. Second, we observe our loss-based attackers are more sensitive to Gaussian noises. Despite stronger attacks of structured loss maps, they are easier to protect with Gaussian noises. Finally, we realize this defense is hard to mitigate leakage for dependent attacks. Even though we employ very strong noises for this, losing mIoU from 59.88 to 23.17, it still has more than 0.75 AUC-scores for both attacks. To conclude, Gauss is hard to protect a model when the noises of the same distribution are added to victim and shadow models, and binary classifiers can pick useful information from noisy inputs.

Dropout. It is used to avoid overfitting in training a deep neural networks, that we applied in training our victim model with dropout ratio 0.1. However, it does not hide membership from our studies in section 2. Therefore, we enable dropout operation during testing to blur a prediction. We realize a network still produces decent results when we use a different dropout ratio. Hence, we apply dropout ratio 0.1, 0.5 and 0.9 to obfuscate a prediction at different degrees. We show the joint plots in Fig. 4(b). From our study, we observe enabling dropout during test is able to slightly mitigate membership leakage, but segmentation performance decreases a lot when a large ratio is applied.

DPSGD. Differential Privacy SGD (DPSGD) [1] adds Gaussian noises on the clipped gradients for individual examples of a training batch, in a way that the learnt parameters and hence all derived results such as predictions are differentially private. We apply DPSGD in our study to protect a model. Before training, we collect gradient statistics over entire training data for different layers of a network, and set individual clipping factors for all the layers. Next, we train PSPNet with Gaussian variances 10^{-3}, 4×10^{-3} for dependent settings, and variances 10^{-3}, 4×10^{-3}, 8×10^{-3} for independent settings. For UperNet, we train with 10^{-6}, 10^{-3}, 3×10^{-3}, 6×10^{-3}. Theoretically, the Gaussian noises used in our model is not enough to guarantee a tight differential privacy bound [8]. However, there is a gap between theoretical garuantees and emperical defenses. prior work has shown practical defenses from small gaussian noise and hence loose bounds [14]. In our work, we demonstrate this in semantic segmentation models and show the utility-privacy plots in Fig. 4(c). We observe that DPSGD successfully protect memberships in all the settings, in particular, it will not hamper the utility of segmentation models which only reduces 1.12, 1.36 and 0.75 mIoU when noises with 1e-6 are applied in three rows. Therefore, we recommend DPSGD to train a segmentation model for protecting membership privacy in practise.

Summary of defenses. In spite of the success of membership inference under various settings, we point out feasible solutions which can significantly reduce the risk of information leakage. (1) Adding Gaussian noises helps prevent leakage in independent settings from unknown attackers. Trade-off between model degeneration and information leakage is able to be considered to choose a suitable noise level. Hence, we recommend this method as a basic protection without further costs to prevent potential independent attacks, which are very cheap to implement. (2) For neural networks, we suggest applying DPSGD to train a model, which mitigates the leakage in all the settings with limited model degeneration, even though it adds noises on the gradients during training and hence requires increased training time.



(a) Gauss (b) Dropout (c) DPSGD

Fig. 4: Joint plots for different defenses. x-axis is the AUC-score for membership protection and y-axis is the mIoU for segmentation utility. optimal defenses achieve 0.5 AUC-score while preserving segmentation utility, as drawn with the red diamond.

3.2 Interpretability

One of difference between attacking segmentation and classification is on the input form of the binary classifier, where the input of segmentation can be regarded as an image. Hence, our method can provide interpretations for different examples, indicating important regions for recognizing membership status. Besides, interpretations also help us to understand and compare different defenses. We apply class activation maps (CAMs) [38] to highlight the areas that help to detect examples/patches from training set in Fig. 5. Besides, we also compare the activation areas before and after defenses with structured loss maps.

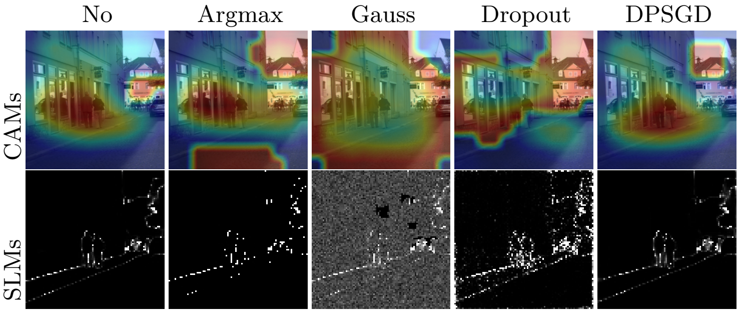


Fig. 5: Class activation maps (CAMs) and structured loss maps (SLMs) for independent attack Deeplab-v3+ →Uper.

First, we observe our attacker is able to mine some regions with specific objects or intersections between two classes, even our attacker has no interaction with a victim. Second, we compare the attacker's different behaviors for those defenses. We can see argmax can simply change the CAM to different intensities, but still hold the major layout of the original CAM. For Gaussian noises, we employ variance 0.1 here, and can apparently observe noises on the structural loss map for all the pixel locations, therefore, it makes all the examples have a similar CAM. For dropout, it will change structured loss maps in many places and then change the CAM. In particular, it changes the locations with strong loss values more than others. For DPSGD, we can see it has very similar loss maps to the original model. The only differences are on some regions hard to segment. Even DPSGD changes the loss maps a little, the final CAMs are able to change a lot for some examples, therefore, it helps defend stealing memberships while preserving segmentation performance very well.

4 Conclusion

We have provided the first membership inference attacks and defenses for semantic segmentation models by extending previous membership attacker for classification and proposing a new specific representation (i.e., structured loss maps). Our study is conducted under two different settings with various model/data assumptions. We show that spatial structures are important to achieve successful attacks in segmentation, and our structured loss maps achieve the best results among all. Besides, we study defense methods to reduce membership leakage and provide safe segmentation. As a result, we suggest to add Gaussian noises on the posteriors in inference, or apply differential privacy in training. We hope that our work contributes to the awareness of novel threats that modern deep learning models pose – such as leakage of information on the training data. Our contributions shows that such threats can be mitigated with little impact on the utility of the overall model.

Acknowledgement The authors thank Apratim Bhattacharyya, Hui-Po Wang and Zhongjie Yu for their valuable feedback to improve the quality of this manuscript.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318 (2016) 3, 12
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: ICCV (2015) 1
3. Backes, M., Berrang, P., Humbert, M., Manoharan, P.: Membership privacy in microrna-based studies. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016) 3
4. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: NeurIPS (2018) 2, 7, 8
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018) 7, 8
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 1, 7
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 1, 7
8. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008) 13
9. Dwork, C.: Differential privacy. Encyclopedia of Cryptography and Security (2011) 3
10. Fischer, V., Kumar, M.C., Metzen, J.H., Brox, T.: Adversarial examples for semantic image segmentation. arXiv preprint arXiv:1703.01101 (2017) 3
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 1
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 7, 8
13. He, Y., Chiu, W.C., Keuper, M., Fritz, M.: Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. In: CVPR (2017) 2
14. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: 28th {USENIX} Security Symposium ({USENIX} Security 19). pp. 1895–1912 (2019) 3, 13
15. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) 1
16. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017) 2
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015) 2, 11
18. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR (2017) 3
19. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR (2016) 3
20. Nasr, M., Shokri, R., Houmansadr, A.: Machine learning with membership privacy using adversarial regularization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (2018) 3

21. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017) 1, 7
22. Oh, S.J., Augustin, M., Schiele, B., Fritz, M.: Towards reverse-engineering black-box neural networks. In: ICLR (2018) 3
23. Oh, S.J., Fritz, M., Schiele, B.: Adversarial image perturbation for privacy protection a game theory perspective. In: ICCV (2017) 3
24. Orekondy, T., Schiele, B., Fritz, M.: Knockoff nets: Stealing functionality of black-box models. In: CVPR (2019) 3
25. Pyrgelis, A., Troncoso, C., De Cristofaro, E.: Knock knock, who's there? membership inference on aggregate location data. NDSS (2018) 3
26. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jegou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: ICML (2019) 2, 3, 4, 5, 7, 8, 9, 10, 11
27. Salem, A., Zhang, Y., Humbert, M., Fritz, M., Backes, M.: Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In: NDSS (2019) 2, 3, 4, 5, 7, 8, 10, 11
28. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1310–1321 (2015) 3
29. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: IEEE Symposium on Security and Privacy (SP) (2017) 2, 3, 4, 5
30. Stutz, D., Hein, M., Schiele, B.: Disentangling adversarial robustness and generalization. In: CVPR (2019) 3
31. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV (2017) 1
32. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016) 8
33. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV (2018) 7, 8
34. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: ICCV (2017) 3
35. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687 (2018) 1, 7
36. Zhang, T.: Privacy-preserving machine learning through data obfuscation. arXiv preprint arXiv:1807.01860 (2018) 3
37. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017) 2, 7, 8
38. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016) 7, 14
39. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE T-PAMI (2017) 1