Evaluate Inference Attacks: Attack and Defense against 2D Semantic Segmentation Models

YIHAN LIAO, City University of Hong Kong, China
JACKY KEUNG, City University of Hong Kong, China
JINGYU ZHANG*, City University of Hong Kong, China
YUROU DAI, Lehigh University, USA
SHUO LIU, City University of Hong Kong, China

Deep learning (DL)-based 2D semantic segmentation (SS) plays a vital role in the perception task of autonomous driving. However, the SS model relies on DL, which makes it vulnerable to inference attacks. Recent research has discovered that SS models are susceptible to the membership inference attack, yet other inference attacks remain underexplored. Our study fills this gap by comprehensively investigating the vulnerabilities of two widely used RGB image-based 2D SS models (*DeepLabV3* and *DeepLabV3+*) against three inference attacks: membership inference, attribute inference, and model inversion. We evaluate the attack effectiveness on three backbones (MobileNetV2, ResNet50, and ResNet101) across three datasets (VOC2012, CityScapes, and ADE20K), where the attack accuracy can reach up to 95% (membership inference), 40% (attribute inference), and 70% (model inversion), revealing that deeper networks are more prone to privacy leakage in inference attacks. Consequently, we introduce differential privacy and model pruning as defensive mechanisms, significantly reducing attack performance, where the average accuracy drops 20% among the three inference attacks. Our findings reveal critical privacy vulnerabilities in SS tasks and offer practical guidance for developing more robust SS models in autonomous driving.

CCS Concepts: • **Security and privacy → Privacy-preserving protocols**; • **Computing methodologies → Computer vision**; **Machine learning**.

Additional Key Words and Phrases: Semantic segmentation, Inference attacks, Differential privacy, Model pruning

1 INTRODUCTION

With the rapid proliferation of autonomous driving technology, vehicles now collect and process vast amounts of data during operation [54]. This data includes road and traffic information and potentially sensitive personal data, such as passengers' location, travel routes, and behavioral patterns. The extensive use and storage of such data heightens the risk of misuse and breaches, raising significant privacy concerns [24, 32, 50]. One of the critical tasks in autonomous driving systems is **S**emantic **S**egmentation (SS), which divides images into different semantic regions (e.g., lanes, vehicles, pedestrians). SS is typically categorized into 2D image-based and 3D point cloud-based technologies, enabling vehicles to perceive their surroundings more accurately, thereby enhancing driving safety and performance [13, 36]. This study focuses specifically on 2D image-based SS tasks.

*Corresponding author.

Authors' Contact Information: Yihan Liao, yihanliao4-c@my.cityu.edu.hk, City University of Hong Kong, Hong Kong, China; Jacky Keung, City University of Hong Kong, Hong Kong, China, jacky.keung@cityu.edu.hk; Jingyu Zhang, City University of Hong Kong, Hong Kong, China, jzhang2297-c@my.cityu.edu.hk; Yurou Dai, Lehigh University, Bethlehem, Pennsylvania, USA, jzhang2297-c@my.cityu.edu.hk; Shuo Liu, City University of Hong Kong, Hong Kong, China, sliu273-c@my.cityu.edu.hk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s).
ACM 1556-4703/2025/4-ART
https://doi.org/10.1145/3732289

Just Accepted

RGB image-based SS models, such as *DeepLabV3* [4] and *DeepLabV3+* [5], have been widely adopted due to their high segmentation performance. However, their reliance on rich contextual information and advanced feature extraction techniques makes them potentially vulnerable to inference attacks [18]. Specifically, *DeepLabV3* uses dilated convolutions to capture multi-scale context, while *DeepLabV3+* improves upon this by optimizing feature fusion and incorporating a decoder module. These enhancements boost segmentation accuracy, but they may also increase the risk of information leakage. Current research has emphasized adversarial attacks [38], which primarily aim to mislead or corrupt model predictions, and the exploration of inference attacks on 2D SS models has been neglected. Besides, although inference attacks have been extensively studied in classification tasks [35], limited studies research the robustness of SS models against inference attacks.

To address this gap, we are the first to conduct a comprehensive inference attack analysis on RGB image-based 2D SS models and provide more reliable solutions for SS technology in autonomous driving systems. This study aims to assess the vulnerabilities of *DeepLabV3* and *DeepLabV3+* under three different inference attacks [47], including **membership inference** [43], **attribute inference** [29], and **model inversion** attacks [59], across three representative backbones (MobileNetV2 [19], ResNet50, and ResNet101 [17]), using widely recognized segmentation datasets (VOC2012 [10], CityScapes [8], and ADE20K [58]).

Membership inference attacks attempt to determine whether a particular data record is in the model's training set, potentially exposing sensitive information about the training data. The recent research on inference attacks against the 2D SS task in autonomous driving explores the SS models against membership inference attacks [6, 18, 55], while our study analyzes the performance of holistic inference attacks and evaluates defense strategies. Attribute inference attacks allow adversaries to infer unexposed attributes of data by analyzing the model's predictions. Model inversion attacks enable adversaries to reconstruct input data from a model's outputs or internal information, thus compromising the privacy and security of autonomous driving.

The experimental results show that the three attacks are successful on all SS models. In member inference attacks, the average F1-score can reach 0.8693 and 0.8813 on *DeepLabV3* and *DeepLabV3+*, respectively, significantly outperforming random guesses (0.5). The *DeepLabV3+* model with ResNet101 as the backbone is more vulnerable to this attack, highlighting the increased risk of information leakage due to the model's greater capacity and complexity. In attribute inference attacks, the success attacks appear under both black-box (exposing inputs and outputs) and white-box (exposing the target model) settings, and the average accuracy close to 50% in the CityScapes dataset significantly surpasses random guess, which is 0.0556 as the inverse of the total number of cities. In the model inversion attacks, the average accuracy reaches 0.7201 and 0.7235 on *DeepLabV3* and *DeepLabV3+*, respectively. Additionally, we prove that the posterior distributions, masks, color histograms of the image, or the feature from the intermediate layer of the model can assist in inference attacks.

To mitigate the impact of these inference attacks, we employ two defense strategies, **D**ifferential **P**rivacy (DP) and model pruning. DP introduces noise into model outputs, thereby reducing the reliability of inference attacks while preserving model utility [2]. The effectiveness of DP in protecting model privacy while maintaining performance has been demonstrated in various works [21, 34]. Model pruning is initially utilized to cut down the number of parameters of the model, thereby lowering the computing complexity, and it is gradually developed for privacy protection [22, 41]. In SS tasks, it helps to weaken the ability to disclose training data details by alleviating overfitting, thus limiting the model's ability to memorize training data. We apply both techniques in this study, and the effective reduction of attack success rates is observed. Specifically, the average inference attack accuracy drops over 20%. While the model pruning also resists inference attacks, the defense efficacy is comparable to DP when pruning thresholds are appropriately chosen.

Contribution. In this study, we explore the vulnerability to three inference attacks of two commonly-used RGB image-based 2D SS models, *DeepLabV3* and *DeepLabV3+*. We systematically identify and summarize the factors and validate the effectiveness of the defensive mechanisms. Our primary contributions are as follows:

Just Accepted

- We conduct a comprehensive assessment of the *DeepLabV3* and *DeepLabV3+* under three inference attacks, revealing their weakness of the 2D image-based SS models. To the best of our knowledge, we are the first to explore 2D SS models' vulnerability in three inference attacks and assess feasible defense strategies.
- We find correlations between model complexity and the severity of inference attacks, especially membership inference and model inversion attacks, where higher model complexity generally leads to a greater risk of information exposure.
- We investigate DP and model pruning as potential defense strategies to prevent inference attacks, providing an empirical foundation for privacy protection in SS tasks. We are the first to introduce model pruning to mitigate data leakage in 2D image-based SS tasks.

2 RELATED WORK

2.1 2D Image-based Semantic Segmentation

SS assigns each pixel in an image a semantic label (e.g., road, vehicle, pedestrian), making it a core technique in autonomous driving by providing pixel-level scene understanding for safe and efficient navigation [30]. Early approaches built upon fully convolutional networks [27], which adapted classification networks for dense prediction. Subsequent architectures, such as U-Net [37] with its symmetric encoder-decoder design that improves accuracy, while Pyramid Scene Parsing Network [56] further boosted performance via multi-scale context pooling. More recently, *DeepLabV3* and *DeepLabV3+* [46] have emerged as foundational models in modern autonomous driving systems, offering robust multi-scale feature fusion and superior segmentation accuracy.

Historically, SS mainly centered on 2D image data, using RGB images for pixel-level classification. In contrast, 3D point cloud-based segmentation operates in three-dimensional space, capturing more precise shapes and spatial relationships of objects [48]. For autonomous driving, 3D segmentation typically relies on LiDAR sensors, which provide highly accurate depth information [31]. Nevertheless, 2D image-based segmentation remains widely used due to the low cost, easy deployment of cameras, and well-established training pipelines [44]. Besides, 2D and 3D data can be fused to achieve complementary advantages, which makes 2D SS indispensable for modern autonomous driving [23]. However, in autonomous driving, SS tasks identify lanes, vehicles, and pedestrians, supporting high-level decisions like path planning and obstacle avoidance, the detailed scene information in segmentation outputs may reveal personal or location-specific data if misused. Especially, SS models continuously obtain and process large-scale data from real-world scenarios, making them vulnerable to inference attacks [1].

2.2 Inference Attacks

Inference attacks are distinguished from adversarial attacks, the former is inferring training data or private information from the model, aiming to compromise privacy, and the latter is making the model output wrong results through small perturbations [45].

Membership inference attack involves determining whether specific data is part of the training set of a machine learning model. Successfully executing the attack can lead to unauthorized access to sensitive information, causing potential data leaks [20]. This is particularly concerning in autonomous driving SS tasks, where outputs are inherently rich in sensitive information, including user preferences, frequent destinations, and routes. In autonomous driving systems, an adversary could exploit a membership inference attack to infer which data points (e.g., specific drivers' data or vehicles' locations) were used in training, potentially allowing them to track individuals, predict driving habits, or even access private and sensitive user data. Such attacks can severely undermine privacy and trust in autonomous systems.

In 2018, Salem et al. [40] proved that adversaries could conduct the membership inference attack without prior knowledge of the victim model's parameters. In 2019, Sablayrolles et al. [39] expanded the understanding of the membership inference attack by categorizing them into black-box and white-box attacks. Black-box attacks

Just Accepted

restrict adversaries from only accessing the model's outputs, while white-box attacks grant access to the model's parameters and architecture. In this study, the black-box setting is employed in membership inference attacks to reflect realistic adversary scenarios.

Prior studies have demonstrated the efficacy of membership inference on 2D SS models [6, 18]. Zhang et al. [55] extended the attack by proposing a label-only method to improve attack performance. However, previous studies focused exclusively on the membership inference attack, neglecting other inference attack types that also pose significant privacy risks. Our study also concludes attribute inference and model inversion attacks to provide a more comprehensive assessment of privacy vulnerabilities of the SS model with different backbones.

Attribute inference attacks deduce sensitive attributes of data from model outputs, even when the model is not explicitly trained to predict these attributes, which can result in unintended privacy leaks. Previous studies have confirmed that attribute inference can cause data leakage in DL models on social networks by deducing the user's behavior records and could infer race based on facial datasets [28]. In autonomous driving systems, sensor-captured data, such as camera outputs, may expose sensitive attributes, allowing adversaries to infer location-based features or user-specific details from SS outputs. The adversary could potentially infer the vehicle's location based on road patterns or frequently visited areas captured by the vehicle's sensors, posing privacy risks. In a typical attack scenario, the adversary might analyze the segmented outputs of the SS model to infer attributes like the driver's routine destinations, travel behavior, or specific locations the vehicle has visited.

Xiong et al. [50, 51] proposed GAN-based methods to obfuscate sensitive information in camera images and prevent location inference. However, their approach focused on identifying unique objects, such as special buildings, rather than concentrating on SS tasks. In SS tasks, detailed pixel-level predictions inherently further increase the risk of attribute inference, amplifying privacy concerns in autonomous driving systems. Our study specifically addresses the vulnerabilities of 2D SS models in autonomous driving, analyzing the effectiveness of attribute inference attacks based on the street view and evaluating defense strategies.

Model inversion attacks aim to reconstruct input data, such as images or attributes, from a model's outputs or internal states, posing significant privacy risks. The concept of model inversion attacks was first introduced by Fredrikson et al. [11], demonstrating that recognizable images of individuals could be reconstructed from a facial recognition model's output probability. Yang et al. [52] showed that even with limited access to model outputs, it is possible to perform inversion attacks effectively. This highlights the potential for model inversion to exploit detailed outputs, particularly concerning pixel-wise tasks like SS tasks.

In autonomous driving systems, model inversion attacks have negative implications for semantic segmentation models due to their detailed pixel-level outputs. These models are particularly vulnerable because the high-resolution segmentation can be used to infer detailed and sensitive information from the input data. Recently, Shibly et al. [42] demonstrated that adversaries could reconstruct scenes from training data by leveraging gradient information and auto-encoders, but it aims to combine memory modules to reconstruct inputs destroyed by adversarial perturbations. In contrast, model inversion attacks, which focus on reconstructing training data from model output, present more serious privacy risks to 2D image-based SS tasks.

In summary, all inference attacks can threaten the 2D semantic segmentation models, but current research has limited studies on the vulnerability of holistic inference attacks. Therefore, we fill this gap in the comprehensive assessment of inference attacks and explore factors to improve the models' robustness and privacy-preserving ability.

2.3 Defenses

Differential privacy (DP) is a privacy-preserving strategy that introduces controlled noise into data to balance data utility and individual privacy [14]. It has been widely applied in various fields, such as enhancing privacy in natural language processing by masking sensitive linguistic patterns [52] and in the Internet of Things by securing

Just Accepted

device-generated data [57]. In autonomous driving, Han et al. [15] employed DP to anonymize trajectory data, protecting vehicular location information, while Li et al. [24] proposed a DP-based federated learning strategy to secure vehicular message privacy. However, these works focus on protecting user behavior and location data, neglecting the risks of image data leakage. Therefore, we introduce an adaptive mean-value method to integrate DP into image data protection to fill this gap.

Model Pruning is a technique to reduce the computational complexity and storage requirements of models by eliminating redundant neurons [26]. Han et al. [16] first introduced pruning methods that significantly reduced model parameters and computation by removing redundant weights and quantizing them. Over the years, pruning methods have evolved to include structured pruning, which removes entire filters or layers, and unstructured pruning, which targets individual weights [12]. Beyond efficiency, model pruning has emerged as a potential privacy protection technique. Yuan et al. [53] demonstrated that reasonable pruning strategies could mitigate the membership inference attack by reducing the information retained in model parameters. In 2023, Chu et al. [7] proposed the PriPrune algorithm, which applies pruning in federated learning to enhance privacy protection and reduce information leakage. In this study, we leverage model pruning to balance performance and privacy in SS tasks.

Overall, DP has demonstrated defense effectiveness in various fields, including autonomous driving, where it primarily protects user-related data such as identity, behavior, and location. However, its application to image data privacy remains underexplored, leaving gaps in protecting sensitive information in SS tasks. Similarly, while model pruning is increasingly recognized for its privacy-preserving potential, its use against inference attacks in 2D image-based SS tasks has not been studied. This work fills these gaps by evaluating the effectiveness of DP for securing image data and introducing model pruning as a novel defense mechanism to resist inference attacks.

3 METHODOLOGY

In this section, we first introduce the 2D image-based SS models utilized in this study and three inference attacks against these models on different datasets, followed by technologies of defense strategies.

3.1 Semantic Segmentation

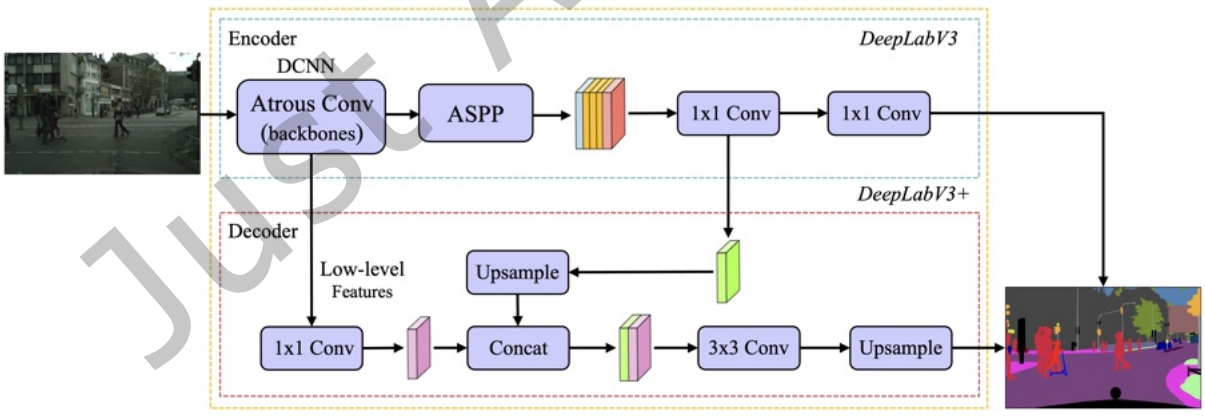


Fig. 1. The model structure of DeepLabV3 and DeepLabV3+. The blue box is the DeepLabV3, the red box is the decoder module, and the yellow box is the DeepLabV3+.

DeepLabV3 [4] is a widely used architecture for SS that effectively captures multi-scale contextual information. It integrates key components to preserve high-resolution feature maps while gathering contextual data at

Just Accepted

different scales. As shown in Figure 1, the typical structure of *DeepLabV3* is highlighted in the blue box, whereas *DeepLabV3+*, represented by the entire yellow box, includes both the encoder and decoder.

One of the key innovations in *DeepLabV3* is the use of Atrous Convolutions, which apply different dilation rates to feature maps to capture multi-scale context while maintaining the resolution of the input. The backbone network, which includes MobileNetV2, ResNet50, and ResNet101, extracts features from the input image. The Atrous Spatial Pyramid Pooling (ASPP) module then aggregates multi-scale information using atrous convolutions with different dilation rates, where the formal representation of atrous convolution is:

$$y[i] = \sum_k x[i + r \cdot k] \cdot w[k], \tag{1}$$

where $x[i], y[i]$ represent the input and output feature map respectively, $w[k]$ is the filter weight at position k, and r is the atrous rate, determining the spacing between the filter weights. This architecture is particularly useful for SS tasks, as it enables the model to handle complex, high-resolution input data like images in autonomous driving scenarios.

Building upon *DeepLabV3*, *DeepLabV3+* [5] introduces a new decoder module that enhances the accuracy of object boundaries and restores the original image resolution. This is achieved by upsampling the feature maps and combining them with outputs from the ASPP and backbone networks. The decoder is shown in the red box of Figure 1, where the concatenated features are processed through convolution layers to generate the final segmentation map. The backbone network options, including MobileNetV2, ResNet50, and ResNet101, are consistent with *DeepLabV3*, ensuring that the model benefits from their robust feature extraction capabilities.

The models are trained using a pixel-wise cross-entropy loss function (L_{CE}), which calculates the discrepancy between the predicted segmentation map and the ground truth labels. The loss is defined as: $L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$, where y is the true label, $\hat{y}$ is the predicted probability, and N represents the number of classes. To evaluate segmentation performance, we use the mean intersection over union (mIoU) and pixel accuracy (PA) as metrics:

$$\text{PA} = \frac{\text{number of pixels correctly classified}}{\text{total number of pixels}} \tag{2}$$

$$\text{mIoU} = \frac{1}{N} \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{3}$$

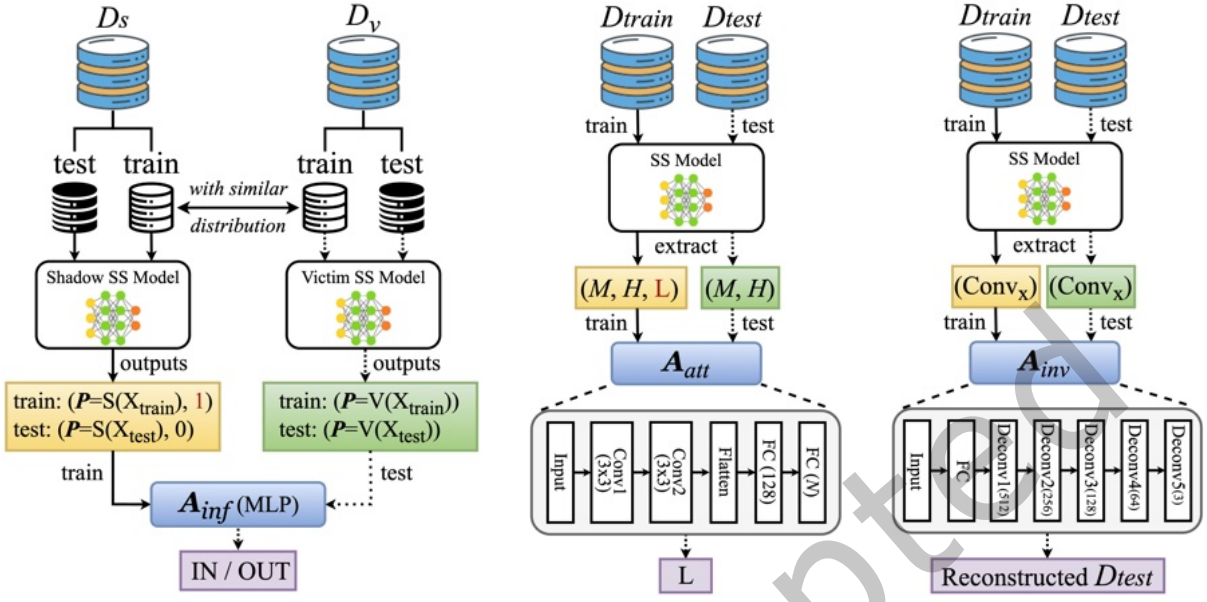
where TP is pixels correctly classified as the target class, FP represents pixels incorrectly classified as the target class, and FN means pixels that belong to the target class but are incorrectly classified as another class.

3.2 Inference Attacks Models

Figure 2 presents all the processes of inference attacks. During the training phase of all SS models, we follow the training protocol of [4, 5], applying data augmentation techniques, including random scaling of the input data (from 0.5 to 2.0), crop size to 513, and horizontal flips to prepare the data for training. In the testing phase, the data processing involves only basic normalization, ensuring that the model receives consistent input for a realistic assessment of real-world data.

Figure 2a illustrates the architecture of the **membership inference attack**, which can determine whether specific data is part of the training set of a victim model. A key component of membership inference is the shadow SS model ($\mathcal{S}$), which approximates the behavior of the victim SS model ($\mathcal{V}$) by training on a shadow dataset (D_s) with a similar data distribution to the training set of the victim dataset (D_v). The feasibility of this assumption depends on the adversary's capability. In this study, we adopt the black-box setting where the adversary has knowledge of $\mathcal{V}$'s training domain but not its exact parameters.

Just Accepted



(a) Membership inference. (b) Attribute inference. (c) Model inversion.

Fig. 2. The structure of membership inference, attribute inference, and model inversion attacks. In membership inference, $\mathcal{A}_{inf}$ is the membership inference attack model, $\boldsymbol{P}$ represents the combined mean and standard posterior probability from the model, and 0 and 1 are the corresponding membership labels. In attribute inference, $\mathcal{A}_{att}$ is the attribute inference attack model, $\boldsymbol{M}$ and $\boldsymbol{H}$ denote the image mask and color histograms, respectively, and the L is the city label. In model inversion, $\mathcal{A}_{inv}$ is the model inversion model, $Conv_x$ represents the features from the intermediate convolutional layer x.

In most membership inference attack settings, $\mathcal{S}$ is assumed to have the same architecture as $\mathcal{V}$ to ensure similar output distributions, as established in prior works [6, 39, 43]. This crucial assumption helps to maintain consistent feature extraction processes, making the outputs from both models more comparable, which is essential for the adversary to function effectively. While different architectures may affect attack performance, previous research has shown that the impact is limited when both models are trained on similar data distributions [3]. Moreover, in SS tasks, even when trained on different datasets, the attack performance tends to improve when $\mathcal{S}$ is simpler than $\mathcal{V}$, compared to when both the $\mathcal{S}$ and the $\mathcal{V}$ share the same architecture [18]. Based on these findings, we assume that the architectures of $\mathcal{S}$ and $\mathcal{V}$ are identical in our study.

However, prior research has not specifically examined the performance of the $\mathcal{S}$ under different backbones in SS models. Therefore, we evaluate the performance of the $\mathcal{S}$ by varying its backbone architecture while keeping the training data distribution constant. Specifically, both D_s and D_v are divided into training and testing sets: D_s^{train}, D_s^{test} in $\mathcal{S}$, and D_v^{train}, D_v^{test} in $\mathcal{V}$. During the $\mathcal{S}$ training, the model learns from the training set D_s^{train} of D_s and the outputs posterior probability ($\boldsymbol{P}$) derived from the model's *logits* (the raw, untransformed output values produced by a model). These *logits* are transformed into probabilities using the softmax function for multi-class classification, defined as: $softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$, where z_i is the *logit* for class i, j represents the index of the class, and K is the total number of classes. Then, the $\boldsymbol{P}$ is the combination of the mean and standard deviation of the posterior probability. During the training of $\mathcal{S}$, $\boldsymbol{P} = \mathcal{S}(D_s^{train})$, with the label 1 (*member*), indicating that the training data belongs to the model's training set. For the testing set D_s^{test} of $\mathcal{S}$, the model's outputs $\boldsymbol{P} = \mathcal{S}(D_s^{test})$

Just Accepted

with the label 0 (*non-members*). These labeled outputs create a dataset for training the membership inference attack model ($\mathcal{A}_{inf}$), which typically uses a 5-layer multilayer perception (MLP) structure, learns to differentiate between the *member* and the *non-member* based on the $\mathcal{S}$'s outputs.

Meanwhile, $\mathcal{V}$ is trained on D_v^{train}, with the model outputs $\boldsymbol{P} = \mathcal{V}(D_v^{train})$ and $\boldsymbol{P} = \mathcal{V}(D_v^{test})$. After training on the $\mathcal{S}$'s outputs, the $\mathcal{A}_{inf}$ takes the unlabeled outputs of $\mathcal{V}$ as input and infers whether each instance belongs to the $\mathcal{V}$'s training set or not. The formal expression is:

$$\textbf{Membership Inference}(D_s, \mathcal{A}_{inf}, \mathcal{V}, \theta) = \arg\max_{y \in \{0,1\}} P(y \mid f_{\mathcal{V}}(\boldsymbol{P}; \theta), \mathcal{A}_{inf}(\boldsymbol{P}, f_{\mathcal{V}}(\boldsymbol{P}; \theta))) \tag{4}$$

To evaluate the performance of the membership inference attack, we use accuracy, F1-score, and False Positive Rate (FPR) [3, 25]. Accuracy measures the proportion of correctly classified samples out of the total number of samples, F1-score reflects the balance trade-off between false positives (FP) and false negatives (FN), FPR measures the proportion of non-member samples that are incorrectly classified as members. The metric equations are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{F1 Score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{6}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{7}$$

True positive (TP) is the number of member data correctly predicted as members, while FP indicates the number of non-member data predicted as members. True negative (TN) refers to the number of non-member data that are correctly predicted as non-members. FN represents the number of member data that are incorrectly classified as non-members. The accuracy and F1-score above 50% indicate a successful attack since membership inference is a binary classification problem, where the probability of a random guess is 50% [43]. Additionally, a high FPR suggests that a significant number of non-member samples are incorrectly classified as members, which undermines the attack's effectiveness. Therefore, both the accuracy and FPR should be considered when evaluating the success of the attack. The success of this attack is influenced by the adversaries's knowledge about $\mathcal{V}$, including its architecture, backbones, and dataset size.

Attribute inference attack aims to infer the city where the vehicle is located from the input image that contains landmarks or unique infrastructure. The structure of this attack is shown in Figure 2b. In this study, this attack is under both black-box and white-box settings. The former reflects the reality of attack scenarios, and the latter provides an upper bound of attack performance and helps to evaluate the degree of disclosure of attribute information. Specifically, in the black-box setting, the SS model (C) is trained on the training set, and the final output image masks ($\boldsymbol{M}$) are trained on the attribute inference attack model ($\mathcal{A}_{att}$). Under the white-box setting, the feature includes $\boldsymbol{M}$ and color histograms ($\boldsymbol{H}$), which are extracted from the intermediate layer of the SS model. These features reflect key information about the image content, enabling $\mathcal{A}_{att}$ to effectively leverage the fine-grained details that are critical for predicting the city label (L).

During the training stage, the $\mathcal{A}_{att}$ training on the features and the corresponding L from the training set for learning. The $\mathcal{A}_{att}$ is a CNN model, consisting of two convolutional layers with the 3x3 kernel, followed by a flattening layer and two fully connected layers with 128 and N units, N is the total number of cities in the dataset. During the testing phase, the $\mathcal{A}_{att}$ is fed with the $\boldsymbol{M}$ and $\boldsymbol{H}$ extracted from the testing set to predict L for each data point. The integration of $\boldsymbol{M}$ and $\boldsymbol{H}$ enables the $\mathcal{A}_{att}$ to comprehensively analyze image content, allowing it to infer city-related attributes. The attribute inference attack principle is:

$$\textbf{Attribute Inference}(D, \mathcal{A}_{att}, C, \theta) = \arg\max_{L \in \mathcal{A}_{aia}} P(L \mid f_C(\boldsymbol{M}, \boldsymbol{H}; \theta), \mathcal{A}_{aia}(f_C(\boldsymbol{M}, \boldsymbol{H}; \theta))), \tag{8}$$

Just Accepted

The primary evaluation metric is attack accuracy, chosen for its relevance in classification tasks, with the success threshold set at the random classification rate. This rate is defined as the inverse of N, where if the accuracy exceeds this threshold, the attribute inference is considered successful. Unlike membership inference, attribute inference involves classification tasks with relatively balanced data distributions, making accuracy a reliable metric for evaluating attack performance.

Model inversion attacks aim to infer raw input data by reconstructing it from the model's intermediate layer outputs, including features and activation values. These attacks are considered white-box by accessing the SS model's internal structure and intermediate outputs. Figure 2c explains the attack structure, including its components. The SS model ($\mathcal{I}$) is trained on the training set and extracts features ($Conv_x$) from the intermediate layer x. These features serve as inputs to train the inversion attack model ($\mathcal{A}_{inv}$), which attempts to reconstruct the testing images.

In this study, we extracted the model features from the intermediate convolutional layer. Because its output contains a wealth of local and global features, it contributes to an inversion attack to reconstruct the input image. The feature map of the convolutional layer is hierarchical and can provide feature information from low level to high level, which is helpful to restore image details. Moreover, the features of the convolutional layer can be easily extracted through forward hooks, which improves the operability of the attack.

During training, $\mathcal{A}_{inv}$ learns from the convolutional features of the training set processed by $\mathcal{I}$, and intermediate parameters are extracted from a selected convolution layer using forward hooks, which capture feature maps during the forward pass, ensuring that key structural and semantic information is retained for reconstruction. Then $\mathcal{A}_{inv}$ uses the corresponding features to perform the reconstruction through a series of deconvolution layers. The formula expression is:

$$\textbf{Model Inversion}(D, \mathcal{A}_{inv}, \mathcal{I}, x, \theta) = \arg\min_{\hat{x}} L(\mathcal{A}_{inv}(\mathcal{I}(\mathcal{F}(x)); \theta), x), \tag{9}$$

where $\hat{x}$ is the reconstructed image, $\mathcal{F}(x)$ represents the features extracted from the input image x, and $\mathcal{I}$ refers to the SS model. The reconstruction process employs a weighted combination of Mean Squared Error (MSE) loss (L_{MSE}) and perceptual loss (L_{Per}) as the loss function. The weights are designed to balance pixel-level fidelity (L_{MSE}) and high-level semantic coherence (L_{Per}), ensuring accurate pixel reconstruction and the preservation of higher-order image features. The L_{Per}, computed using a pre-trained Visual Geometry Group (VGG) network, captures high-level semantic features by comparing feature activations in deep layers of the network. This ensures that the reconstructed images are visually and semantically similar to the originals, improving the perceptual quality of the reconstruction. The formal expression of L_{MSE} and L_{Per} are:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \tag{10}$$

$$L_{Per} = \frac{1}{N}\sum_{i=1}^{N}\|\phi(x_i) - \phi(\hat{x}_i)\|_2^2, \tag{11}$$

where x_i is the input image, $\hat{x}_i$ is the inference result, $\phi(x_i)$ is the extracted feature map, and N is the number of data.

The deconvolution layers in $\mathcal{A}_{inv}$ are 5 layers ($Deconv_1$ to $Deconv_5$), which are essential for recovering the spatial resolution of the image and reintroducing fine details lost during the feature extraction process [33]. These layers progressively reduce the number of units in each deconvolution layer from 512 channels to 3 channels in the output layer. The decreasing number of channels at each layer reflects the process of refining and reconstructing the image from abstracted features, where the goal is to progressively recover the spatial resolution of the image and reintroduce the fine details lost during the feature extraction process. Specifically, earlier deconvolution

Just Accepted

layers may capture larger, more abstract features, and later layers focus on reconstructing finer details, with the final layer returning the image to its original color channels (3 for RGB images). Each deconvolution operation can be expressed as: $x_{i+1} = ReLU(InstanceNorm(ConvTrans2d(x_i))$, where x_i is the output of the layer $i+1$, $InstanceNorm$ used to standardize the output of each sample, and $ConvTrans2d$ represents the deconvolution layer for image upsampling.

During the evaluation, pixel accuracy and canny edge loss (L_{Canny}) are used as metrics. Pixel accuracy measures the proportion of correctly reconstructed pixels. The canny function refers to the canny edge detection algorithm, where any pixel with a gradient magnitude above the upper threshold is immediately classified as an edge, whereas pixels with a gradient magnitude below the lower threshold are classified as non-edges. Pixels with gradient values between the lower and upper thresholds are classified based on their connectivity to known edges. If a pixel is connected to an edge pixel (i.e., part of a continuous edge), it is also classified as an edge. Otherwise, it is classified as a non-edge. By comparing these edge maps, the L_{Canny} measures how closely the edges in the generated image align with those in the original image.

L_{Canny} relies on using the CV2 library to evaluate how closely the edge maps of the generated image match the original image. The squared differences between edge values at corresponding pixel positions are averaged to form the loss. The L_{Canny} is computed as:

$$L_{Canny} = \frac{1}{N}\sum_{i=1}^{N}(\text{Canny}(\hat{x}_i) - \text{Canny}(x_i))^2, \tag{12}$$

where N is the total number of pixels, $\hat{x}_i$ and x_i represent the i-th pixel of the generated image and the original image, respectively. Canny($\hat{x}_i$) and Canny(x_i) represent the output of the canny edge detection applied to the generated image and the original image at the i-th pixel.

In summary, for a clear comparison of the three inference attacks, we summarize the knowledge levels and target consequence of them in Table 1.

Table 1. Comparison of knowledge levels and target consequence in different inference attacks.

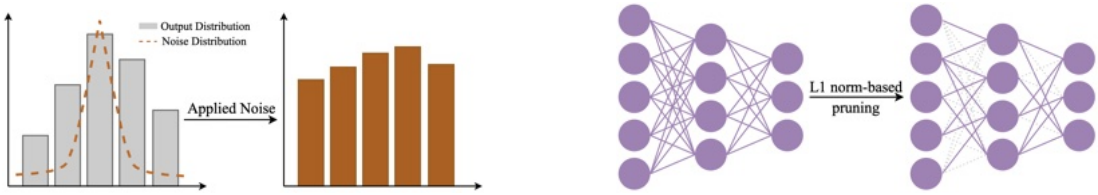
Attack Type	**Knowledge Level**	**Target Consequence**
Membership Inference	Black-box: Only query access to model's outputs Train $\mathcal{S}$ on D_s with similar distribution to D_v	Determine whether given samples are used for training $\mathcal{V}$
Attribute Inference	Black-box: Only access to output masks $\boldsymbol{M}$ White-box: Access to internal features $\boldsymbol{H}$ and $\boldsymbol{M}$	Predict missing attributes from model outputs (i.e., city label)
Model Inversion	White-box: Full access to model's structure and intermediate layer parameters	Reconstruct input images from intermediate feature representations

3.3 Defense Strategies

DP is an advanced technique designed to protect sensitive information by strategically introducing controlled noise into data, thereby balancing utility and confidentiality. In inference attacks, models' outputs are the primary target, as adversaries can exploit them to extract private information. To mitigate this risk, DP injects noise to interfere with the outputs, making it more challenging for adversaries to infer sensitive details.

In this study, we apply noise to the victim model's training phases during the attack model training process. By introducing DP, the models' output can become less sensitive to individual data points, ensuring that the privacy of the data is preserved. Specifically, in the membership inference attack, DP is applied to $\boldsymbol{P}$, in attribute

Just Accepted



(a) Differential Privacy. (b) Model Pruning.

Fig. 3. Defense mechanism principles. In differential privacy, by introducing the Laplace noise distribution into the original distribution, the output distribution is smoothed, making it difficult to infer the presence of any individual data point. In model pruning, the less important weights are removed to reduce the model's complexity and mitigate overfitting, which helps to privacy-preserving.

inference attacks, DP is employed on to the $\boldsymbol{M}$ and $\boldsymbol{H}$, and in model inversion attacks, DP is applied to the features extracted from intermediate layers during SS model training.

To generate noise in a manner consistent with DP, we select the Laplace distribution, which is characterized by its sharp peak around the mean and long tails. This distribution is selected for noise generation due to its ability to provide calibrated perturbations that align with DP requirements [9]. The probability density function of the Laplace distribution is defined as:

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b}), \tag{13}$$

where μ represents the mean, and b is the scale parameter controlling the spread of the distribution. The effect of noise in DP is expressed in Figure 3a, where the introduction of Laplace noise smoothens the output distribution, making it more challenging to infer the individual data.

Another crucial parameter of DP is the privacy budget (ϵ), which inversely correlates with data availability and governs the level of privacy protection. A higher value of ϵ results in better model utility while offering moderate privacy guarantees. However, setting ϵ too low can severely degrade model performance, whereas excessively high values weaken privacy protection. Based on prior research, an ϵ value between 10 and 100 is typically chosen to balance privacy and utility [2]. For ϵ-differential privacy, the Laplace distribution can be modeled as $D \sim Lap(\mu, \frac{\Delta f}{\epsilon})$, where Δf is the sensitivity, typically assigned a value of 1, which measures the maximum variation in the output of a function f due to a unit change in its input.

Traditionally, the Laplace distribution uses μ as 0 for noise injection. However, this approach is often less effective in obscuring the models' outputs, especially when the outputs are skewed or have significant values. To address this issue, we propose an adaptive mechanism for the Laplace distribution that adjusts the noise parameters dynamically. Instead of a fixed mean, the mean is dynamically set to the average *logits* of the SS models during the injection process. This ensures that the injected noise aligns with the magnitude of the model output, providing more effective perturbation, particularly when the posterior values are high and require more substantial noise.

The specific pseudo-code of DP against the membership inference attack model is present in Algorithm 1. Initially, computes the mean value of the $\mathcal{S}$'s prediction outputs on D_s across all categories, which serves as the central value for noise generation. Subsequently, random noise is sampled from a Laplace distribution, where the scale parameter controls the noise intensity, thereby determining the level of privacy protection. The generated noise is then added to outputs, increasing the defense against $\mathcal{A}_{inf}$ without significantly compromising model performance. The core idea behind this approach is to perturb the models' outputs, making it more challenging for

Just Accepted

adversaries to infer membership information. By introducing this noise perturbation, the attack model struggles to accurately capture the output patterns of the target model, thereby reducing the effectiveness of $\mathcal{A}_{inf}$.

Model Pruning is a widely used technique in deep learning for model compression, aimed at reducing computational complexity, memory usage, and inference latency by removing redundant parameters [26]. By eliminating less important weights and connections, pruning reduces memory and computational costs and mitigates overfitting by limiting the model's ability to memorize training data, thereby enhancing privacy protection. This reduction in overfitting makes the model less capable of retaining sensitive information, further strengthening privacy defenses. The impact of pruning on improving attack resistance is particularly significant in inference attacks, as it can prevent adversaries from exploiting specific features accessible in the unpruned model. While the primary motivation for pruning is to optimize computational resources, its importance in SS tasks becomes particularly apparent when integrated into the real-time requirements of autonomous driving systems. Specifically, pruning contributes to faster inference and reduced latency, both crucial for real-time decision-making in autonomous driving.

Algorithm 1 Introducing Differential Privacy as Defense Strategy against Model Outputs

Require: Trained shadow SS model $\mathcal{S}$, input shadow dataset D_s, loss function $\mathcal{L}$, noise scale σ
Ensure: Differentially private perturbed model outputs

Initialize: $O \leftarrow [], L \leftarrow []$
for each $(X, Y) \in D_s$ **do**
  $O_X \leftarrow \mathcal{S}(X)$ ▷ Obtain model outputs
  $\boldsymbol{P} \leftarrow softmax(O_X)$ ▷ Compute posterior probabilities using softmax
  $\mu \leftarrow \text{mean}(O_X)$ ▷ Compute the mean of model outputs
  $\eta \sim \text{Laplace}(\mu, \sigma)$ ▷ Sample Laplace noise with scale σ
  $\boldsymbol{P} \leftarrow \boldsymbol{P} + \eta$ ▷ Add noise to model outputs in-place
  $\ell \leftarrow \mathcal{L}(O'_X, Y)$ ▷ Compute loss using the perturbed outputs
end for
return O, L

In this study, we apply L1 norm-based unstructured pruning after training the SS model. This technique removes a certain percentage of the smallest magnitude weights in the model, as illustrated in Figure 3b. Compared to structured pruning, which removes the network neurons, unstructured pruning is more appropriate for SS tasks that are sensitive to accuracy. Specifically, given a weight matrix $\mathcal{W}$, the pruning function is defined as:

$$\mathcal{W}' = Prune(\mathcal{W}, \tau), \tag{14}$$

where τ is the pruning threshold that determines which weights to remove, and $\mathcal{W}'$ represents the pruned weight matrix. The optimization objective for pruning is given by:

$$\min_{\mathcal{W}'} L(\mathcal{W}') + \lambda \|\mathcal{W}'\|_1, \tag{15}$$

where $L(\mathcal{W}')$ is the loss function of the pruned model, $|\mathcal{W}'|_1$ is the L1 norm regularization term promotes sparsity, and λ controls the trade-off between model accuracy and sparsity.

Pruning is applied to all convolutional layers in the network, targeting small-magnitude weights in each layer. By comparing different pruning rates, we can analyze the trade-off between model efficiency, segmentation accuracy, and resilience against inference attacks. After pruning, fine-tuning is performed to restore performance lost due to weight removal. The fine-tuning process uses the L_{CE} loss to guide the model in adapting to the reduced parameter set. This ensures that the pruned model recovers as much accuracy as possible while maintaining the pruning benefits.

Just Accepted

4 EXPERIMENTAL SETTINGS

This section describes the experimental setup used in this study. We first introduce the datasets employed for SS tasks, followed by a summary of the hyperparameters and the configuration of inference attacks. Next, we provide a detailed explanation of the inference attacks and the defense mechanisms implemented to mitigate inference attacks. Our experiments are conducted on a Linux server equipped with an Intel Xeon Silver 4210 CPU and four Nvidia GeForce RTX 3090-Ti GPUs.

4.1 Datasets

This study utilizes three commonly used datasets for SS: VOC2012 [10], CityScapes [8], and ADE20K [58]. These datasets are briefly described as follows:

(1) V0C2012: This dataset contains approximately 11,000 images and is widely used for SS tasks. It includes annotated images across a range of object categories.
(2) CityScapes: A dataset focused on urban scenes, specifically captured for autonomous driving applications. It contains high-resolution images with pixel-level annotations.
(3) ADE20K: A diverse dataset with over 20,000 images covering various complex scenes and object categories. It is commonly used for SS tasks.

In our experiments, we divide each dataset into two subsets with similar distribution for the membership inference attack: one part of the data is D_s for training and testing $\mathcal{S}$, while the remaining part is used for $\mathcal{V}$. For attribute inference, only the training set of CityScapes is employed. In model inversion attacks, the training and testing sets are directly utilized for models C and $\mathcal{I}$.

4.2 HyperParameters.

The hyperparameters fine-tuned on the validation set in this work are outlined in Table 2. These hyperparameters are categorized according to the SS tasks, three inference attacks, and two defense strategies, respectively.

SS Models. In this study, we use *DeepLabV3* and *DeepLabV3+* as the base models for 2D image-based SS tasks, and the backbones of these models include ResNet50, ResNet100, and MobileNetV2. The output stride (*OS*) in *DeepLabV3* and *DeepLabV3+* is the ratio of the spatial resolution of the input image to the final output resolution before the fully connected layer, following the [5] setting as 16.

Membership Inference Attack. This attack aims to determine whether a specific data point is part of the training set. We apply this attack to all three datasets. For this attack, $\mathcal{S}$ is used to mimic the behavior of $\mathcal{V}$. The features from the training dataset in D_s are labeled as *members* (label 1), while features from the validation dataset are labeled as *non-members* (label 0). A fine-tuned 5-layer MLP is used as the attack model $\mathcal{A}_{inf}$, consisting of fully connected layers with 256, 128, 64, 32, and 1 units, respectively, utilizing ReLU activations, dropout as 0.5 for regularization, and a sigmoid output for binary classification to identify the member and non-member data. The layers of the MLP model and regularization strength (dropout rate) are fine-tuned to optimize attack performance by performing on the validation set using a grid search approach.

Attribute inference attack. This attack focuses on inferring hidden attributes of the data. Due to limited attribute information in the VOC2012 and ADE20K datasets, we apply this attack only to the CityScapes dataset, where each image contains hidden city attributes. The dataset is the training set of CityScapes that contains 18 cities, in which 80% data is used to train the C and 20% data for evaluation. The labels correspond to the encoded city name, which can be extracted from the input data path.

Model inversion attack. The goal of this attack is to reconstruct the training data. In this study, all the datasets are utilized, all split into 80% training and 20% testing sets. $\mathcal{A}_{inv}$ trained on extracted features from the intermediate layer, *module.backbone.low_level_features.1.conv.1* in MobileNetV2, and *module.backbone.layer2.0.conv2* in the ResNet-based backbones. These layer strikes a reasonable balance between capturing high-level features of the

Just Accepted

No.	Tasks	Hyperparams	Value	No.	Tasks	Hyperparams	Value
1	SS	Epochs	300	20	$\mathcal{A}_{att}$	Scheduler	StepLR
2		*BS*	[10], [58]: 8	21			Step Size: 50
3			[8]: 16	22			γ: 0.05
4		Optimizer	SGD	23		Loss	L_{CE}
5			lr: 1e-3	24	$\mathcal{A}_{inv}$	Epochs	50
6			wd: 1e-4	25		*BS*	16
7		Loss	L_{CE}	26		Optimizer	Adam
8		*OS*	16	27			lr: 1e-4
9	$\mathcal{A}_{inf}$	Epochs	500	28			wd: 1e-6
10		*BS*	16	29		Loss Weights	L_{MSE}: 0.9
11		Optimizer	Adam	30			L_{Per}: 0.1
12			lr: 1e-2	31		L_{Canny}	$\mathcal{T}$: [100, 200]
13			wd: 1e-3	32	DP	ϵ	20, 60, 100
14		Loss	L_{CE}	33	Pruning	τ	0.3, 0.5, 0.7
15	$\mathcal{A}_{att}$	Epochs	100	34		Fine-tuning	Epochs: 3
16		*BS*	8	35			lr: 1e-4
17		Optimizer	Adam	-	-	-	-
18			lr: 1e-4				
19			wd: 1e-6				

Table 2. Fine-tuned hyperparameters for different tasks. *BS*: batch size. SGD: stochastic gradient descent. lr: learning rate. wd: weight decay. L_{CE}: cross-entropy loss. *OS*: output stride. Adam: adaptive moment estimation. γ: decay rate of the learning rate. L_{MSE}: MSE loss. L_{Per}: Perceptual loss. L_{Canny}: canny edge loss. $\mathcal{T}$: lower and upper threshold range. τ: model pruning threshold.

Just Accepted

image and preserving essential spatial information, enabling it to effectively assist model inversion attacks in recovering input data while mitigating the risk of excessive information loss that may result from earlier or deeper layers of networks. The extracted data then applied adaptive average pooling to reduce the spatial dimensions to 8×8, then flattening and concatenating them into a single feature vector. For reconstruction, $\mathcal{A}_{inv}$ contains a fully connected layer and 5 deconvolution layers, with the decreased output channels 512, 256, 128, 64, and 3.

DP. The defense mechanism that reduces the model's sensitivity to individual data points by injecting noise, thereby smoothing the model's outputs. DP is used to obfuscate the $\boldsymbol{P}$ in $\mathcal{A}_{inf}$, $\boldsymbol{M}$, $\boldsymbol{H}$ in $\mathcal{A}_{att}$, and intermediate layer parameters in $\mathcal{A}_{inv}$, reducing the risk of information leakage, respectively. Besides, the Laplace distribution injects controlled noise into model outputs, where the μ is the average of *logits* of the models' outputs, adaptively adjusted according to the data being obfuscated.

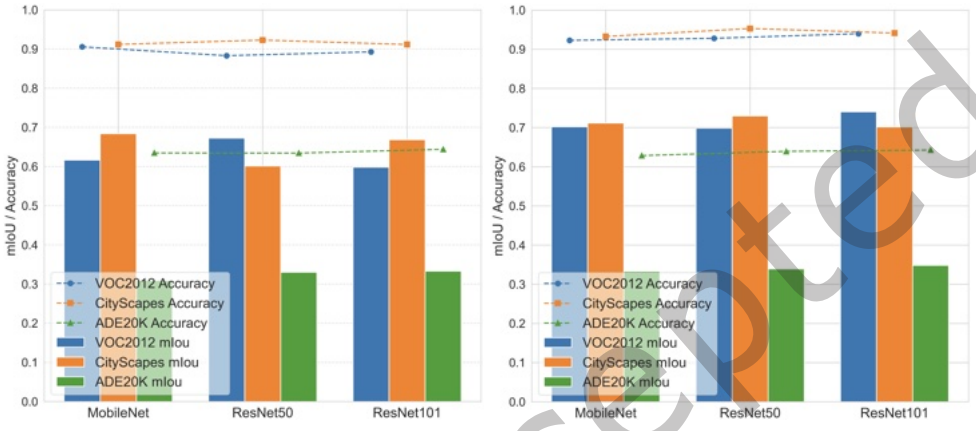
Model Pruning. It is conducted after training the SS models. This technique reduces the size of the model by eliminating unimportant weights and neurons, thereby decreasing the model's capacity to memorize training data. After pruning, the model is fine-tuned to recover performance lost, with the epoch set to 3 to balance model recovery and computational efficiency, considering the time-sensitive nature of autonomous driving applications.

5 EXPERIMENTAL EVALUATION

This section first evaluates the effectiveness of three types of inference attacks respectively, followed by the efficacy of different defense strategies.

5.1 Membership Inference Attack

5.1.1 Attack. In this study, the membership inference attack is conducted as a black-box attack, leveraging the $\mathcal{S}$ to infer membership status. We first conduct an ablation experiment to evaluate the performance of $\mathcal{S}$ trained with different backbone networks (MobileNet, ResNet50, and ResNet101) across three datasets: VOC2012, CityScapes, and ADE20K. Then we evaluate how the diversity of the training set impacts the effectiveness of the membership inference attack. This is followed by an exploration of how the size of the training set affects the generalization ability and overfitting degree of $\mathcal{A}_{inf}$. For the performance of $\mathcal{S}$, the result is shown in Figure 4.



(a) DeepLabV3. (b) DeepLabV3+.

Fig. 4. The mIoU and pixel accuracy of shadow models using three backbones (MobileNetV2, ResNet50, and ResNet101) across different datasets for both DeepLabV3 and DeepLabV3+.

The overall performance of $\mathcal{S}$, evaluated in mIoU and accuracy, differs across datasets, with good performance on VOC2012 and CityScapes, while ADE20K presents the most significant challenge. Specifically, on *DeepLabV3*, the average mIoU for $\mathcal{S}$ on these datasets is 0.6290 for VOC2012, 0.6514 for CityScapes, and 0.3241 for ADE20K. Although CityScapes exhibits a slightly lower mIoU than VOC2012, it maintains a high classification accuracy of 0.9155. In contrast, due to the large number of categories and complex scenes in the ADE20K dataset, the mIoU is much lower than VOC2012 and CityScapes.

With the *DeepLabV3+* architecture, the overall performance of $\mathcal{S}$ improves, especially on VOC2012 and CityScapes, demonstrating that the *DeepLabV3+* architecture handles SS tasks in these datasets more effectively. However, on ADE20K, despite a slight increase in mIoU, the overall performance remains unsatisfactory, with accuracy just slightly over 0.64. This indicates that the improvements made by *DeepLabV3+* are still limited when dealing with more complex scenarios. The overall results prove that the performance of $\mathcal{S}$ is close to $\mathcal{V}$ and can reflect the SS model performance in the perception module of autonomous driving.

Next, we conduct an ablation experiment on how different training data impact the $\mathcal{A}_{inf}$ by comparing features including the mean of posterior probability (*MP*), standard deviation of posterior probability (*SP*), loss of $\mathcal{S}$ (L_{CE}) [18], and maximum confidence score (*MC*) [40]. Besides, we also compare with the previous membership inference attack methods [18] and [6] by strictly reproducing following the official open-source implementations and using the hyperparameters they report with the best performance. Both studies follow the standard membership inference assumption that $\mathcal{S}$ shares the same architecture as $\mathcal{V}$. [18] proposed structured loss maps (L_{CE} values at all the locations) can conduct successful attacks, and the patches of pixels that have strong or small confidence

Just Accepted

scores are rejected to emphasize the important parts. [6] directly using the *logits*. We follow the data split method in [18] and evaluate it with *DeepLabV3+* architecture for both $\mathcal{S}$ and $\mathcal{V}$. The evaluated dataset is CityScapes, and the result is illustrated in Table 3.

	MobileNetV2			**ResNet50**			**ResNet101**		
Methods	F1-score	FPR	Time	F1-score	FPR	Time	F1-score	FPR	Time
[18] (L_{CE})	0.9436	0.1178	4.89	0.9525	0.0917	5.05	0.9618	0.1125	4.92
[6] (*logits*)	0.8824	0.1775	2.06	0.9007	0.2568	2.11	0.9031	0.1254	2.11
Ours (*MP*+*SP*+*C*)	0.9169	0.1602	2.47	**0.9577**	0.2053	2.54	0.9567	0.0701	2.58
Ours (*MP*+*SP*+L_{CE})	0.9028	0.1003	2.52	0.8857	0.1380	2.46	0.9422	0.0831	2.45
Ours (*MP*+*SP*)	**0.9685**	0.1217	2.44	0.9146	0.3801	2.47	**0.9776**	0.0701	2.43

Table 3. The ablation experiment of membership inference attacks performance against different methods. The best is marked bold. The shadow model is *DeepLabV3+*, the dataset is CityScapes, and the backbones include MobileNetV2, ResNet50, and ResNet101. Time (s): Inference time for each batch (size=16). *MP*: mean of the posterior probability. *SP*: standard deviation of posterior probability. *C*: confidence score. *Loss*: loss of the shadow model.

According to the results, our method of combining *MP* and *SP* provides the most effective indicators for distinguishing between members and non-members, reaching 0.9776 in the ResNet101 backbone. Including additional features does not significantly improve performance and even leads to a slight decline in effectiveness, especially when combined with *Loss*. This is likely because the additional features may introduce noise or irrelevant information that detracts from the ability to infer membership, particularly when the model is already leveraging the informative *MP* and *SP* features accurately. The performance differences observed across the backbones (MobileNetV2, ResNet50, and ResNet101) further highlight that simpler models like MobileNetV2 benefit more from the *MP* and *SP* combination, while more complex models such as ResNet101 show a more balanced performance across different feature combinations. Regarding the inference time, the methodology of [6] receives the best result, while the complex loss computation for each class in [18] raises the time cost, which grows as the number of classes increases. Overall, these findings emphasize that carefully selected features are crucial for optimizing $\mathcal{A}_{inf}$, with unnecessary features potentially hindering attack performance. Therefore, for the following $\mathcal{A}_{inf}$, *MP*, and *SP* are selected as the training data.



(a) VOC2012. (b) CityScapes. (c) ADE20K.

Fig. 5. The F1-score comparison of membership inference attacks across three backbones in DeepLabV3, exploring the impact of different ratios of shadow model training sets and victim model training sets on the attack performance.

To investigate the impact of $\mathcal{S}$ training data size on attack performance, we also conduct an ablation study by varying the proportion between D_s and D_v. Specifically, we evaluate the attack effectiveness when the training

Just Accepted

set of $\mathcal{S}$ and $\mathcal{V}$ is 30%:70%, 50%:50%, and 70%:30% of the available training data while keeping all other settings constant. The 50%:50% ratio serves as the baseline, representing the standard assumption in $\mathcal{A}_{inf}$ where $\mathcal{S}$ training data is comparable to $\mathcal{V}$. The 30%:70% and 70%:30% ratios represent extreme cases where the $\mathcal{S}$ has significantly fewer or more training samples than the $\mathcal{V}$.

The F1-score of $\mathcal{A}_{inf}$ under different $\mathcal{S}$ training ratios in *DeepLabV3* with the results shown in Figure 5. The highest F1-score is 0.9741, achieved by ResNet101 on the CityScapes dataset with a 70%:30% training ratio between $\mathcal{S}$ and $\mathcal{V}$, which is attributed to the structured nature of CityScapes and aligns well with SS tasks, leading to higher attack effectiveness. Additionally, ResNet101 has stronger feature extraction and memorization capabilities as a deeper network, making it more susceptible to $\mathcal{A}_{inf}$.

In the CityScapes dataset, at the 30%:70% training ratio using ResNet101, the F1-score (0.9564) slightly surpasses that of ResNet50 trained with 50% of the dataset (0.9435). Possible explanations include the stronger memorization capability of ResNet101, which may retain more training-specific features even with only 30% shadow training data, leading to higher attack success rates. Overall, ResNet101 consistently achieves the highest F1-scores on CityScapes, indicating that deeper networks tend to overfit structured datasets, making them more vulnerable to $\mathcal{A}_{inf}$. Ultimately, the 30%:70% training ratio offers a more realistic attack scenario, which maintains a relatively low attack effectiveness by preventing excessive overfitting, making it a suitable choice for evaluating the privacy risks in autonomous driving perception systems. Therefore, we adopt the 30% training ratio for $\mathcal{S}$ in our experiments.

Regarding the further comprehensive study of $\mathcal{A}_{inf}$, we analyze the performance of both *DeepLabV3* and *DeepLabV3+* models across the three backbones (MobileNetV2, ResNet50, and ResNet101), with the results shown in Table 4 and Table 5. The performance is evaluated using accuracy, F1-score, and FPR across VOC2012, CityScapes, and ADE20K datasets. We highlight the best performance values among the three backbones.

Datasets	**MobileNetV2**			**ResNet50**			**ResNet101**		
	Accuracy	F1-score	FPR	Accuracy	F1-score	FPR	Accuracy	F1-score	FPR
VOC2012	0.7037	0.6641	**0.1513**	**0.7403**	0.7326	0.2057	0.7231	**0.7541**	0.3830
CityScapes	**0.9599**	**0.9707**	**0.1195**	0.9205	0.9435	0.2367	0.9569	0.9685	0.1217
ADE20K	0.8930	0.8947	0.1208	0.9415	0.9416	**0.0570**	**0.9532**	**0.9541**	0.0638

Table 4. Membership inference attack performance against **DeepLabV3** comparison of different backbones on three datasets. The best is marked bold.

Datasets	**MobileNetV2**			**ResNet50**			**ResNet101**		
	Accuracy	F1-score	FPR	Accuracy	F1-score	FPR	Accuracy	F1-score	FPR
VOC2012	**0.8696**	**0.8712**	**0.1253**	0.8051	0.5920	0.0921	0.8066	0.8137	0.2057
CityScapes	0.9569	0.9685	0.1217	0.8746	0.9146	0.3801	**0.9696**	**0.9776**	**0.0701**
ADE20K	0.9017	0.8932	0.1090	0.9416	0.9446	0.1107	**0.9550**	**0.9562**	**0.0705**

Table 5. Membership inference attack performance against **DeepLabV3+** comparison of different backbones on three datasets. The best is marked bold.

According to the result, for the *DeepLabV3* model, the ResNet101 backbone excels not only in SS tasks but also in $\mathcal{A}_{inf}$ performance. On the VOC2012 dataset, ResNet101 achieves an F1-score of 0.7541, higher than ResNet50. However, it also has a significantly higher FPR of 0.3830 compared to ResNet50 and MobileNetV2. This indicates

Just Accepted

that ResNet101, while capturing more features, may also capture more irrelevant or noisy features that lead to more misclassifications. In contrast, ResNet50, being a less complex model, has fewer parameters and may generalize better in $\mathcal{A}_{inf}$, leading to relatively lower FPR and more accurate inference on simpler datasets like VOC2012. On the CityScapes dataset, MobileNetV2 shows a slightly better performance than ResNet-based models in both accuracy and F1-score, with the lowest FPR of 0.1195, which indicates that MobileNetV2 can extract features effectively without leading to too many false positives in the more complex CityScapes dataset. The accuracy and F1-score in ResNet101 are close to MobileNetV2 in CityScapes and are 0.9569 and 0.9685, respectively. On ADE20K, $\mathcal{A}_{inf}$ maintains relatively high effectiveness, especially on the ResNet101, indicating that ResNet101 contributes to membership exploitation on complex datasets, as ResNet101 tends to extract richer patterns from the data. These findings highlight the trade-off between feature richness and false positive rates, with ResNet101 excelling in capturing complex patterns but more prone to misclassifications and MobileNetV2 achieving a better balance in CityScapes.

For *DeepLabV3+*, as shown in Table 5, $\mathcal{A}_{inf}$ demonstrates improved performance over *DeepLabV3*, suggesting that the enhanced segmentation capabilities of *DeepLabV3+* also contribute to increased vulnerability to $\mathcal{A}_{inf}$. In VOC2012, MobileNetV2 achieves the highest accuracy and F1-score, with the lowest FPR of 0.1253, while both ResNet50 and ResNet101 attain similar attack accuracy. However, ResNet101 has a much higher FPR compared to ResNet50, indicating an effective attack but with a higher likelihood of misclassification. On CityScapes, *DeepLabV3+* with ResNet101 achieves the highest accuracy (0.9696), and F1-score (0.9776), and the FPR is reduced to 0.0701, indicating ResNet101 is more prone to overfitting the training data, and exhibits greater robustness in distinguishing training samples from non-training ones compared to *DeepLabV3*. Besides, compared to the previous work, which used the UPerNet [49] as the $\mathcal{S}$ and *DeepLabV3+* as the $\mathcal{V}$, our F1-score is slightly higher than their best performance (0.9620). For ADE20K, the ResNet-based models again demonstrate superior attack success compared to MobileNetV2, where the ResNet101 achieves an accuracy of 0.9532 with an FPR of 0.0638, which is slightly above ResNet50, proving that both backbones are effective in extracting membership-related patterns from this complex dataset.

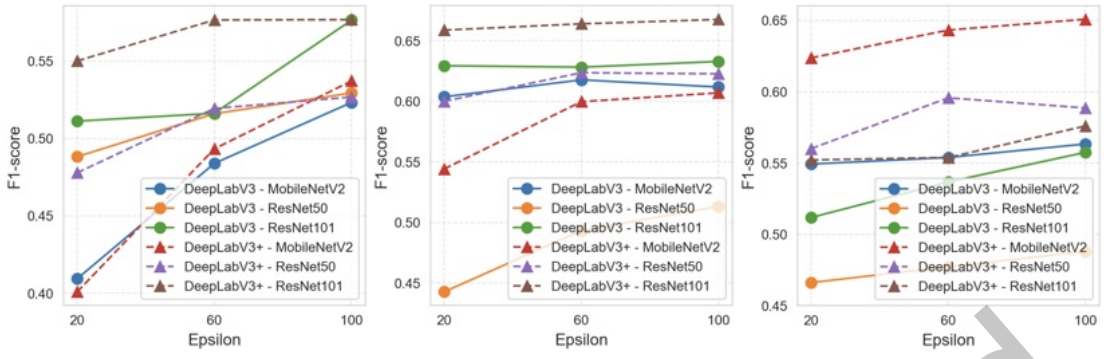
In summary, the results suggest that deeper networks, like ResNet101, are more susceptible to $\mathcal{A}_{inf}$ in real-world autonomous driving's perception module that performs SS tasks. While these models excel at capturing fine-grained features like road markings and object boundaries, this capability can lead to overfitting, making it easier for adversaries to distinguish between training and non-training samples, thus increasing vulnerability to attacks. Datasets with well-defined structures, like CityScapes and ADE20K, which represent urban driving environments, further amplify attack success, as these datasets allow the model to memorize specific details during training. Furthermore, *DeepLabV3+*, as a more complex model, shows a higher susceptibility to $\mathcal{A}_{inf}$ compared to *DeepLabV3*, demonstrating how model complexity and dataset characteristics interact to heighten privacy risks. These findings underscore the potential privacy threats in autonomous driving systems, where adversaries could exploit these vulnerabilities to infer sensitive information, ultimately compromising user privacy and data security.

5.1.2 Defense. Next, we evaluate the effect of DP and model pruning against $\mathcal{A}_{inf}$.

DP. To evaluate the effectiveness of different privacy budgets, we conduct an ablation study by varying the value of ϵ in the DP mechanism. Specifically, we test $\epsilon = 20$, 60, and 100 to analyze its impact on attack performance, and the result is present in Figure 6. Our results indicate that increasing ϵ generally improves attack performance, as the added DP noise decreases the model's ability to obscure membership-related information. However, the extent of improvement varies across datasets and model architectures.

On the VOC2012 dataset, when $\epsilon = 20$, ResNet101 in *DeepLabV3* achieves the highest accuracy of 0.5112, suggesting that lower privacy budgets (i.e., stronger DP noise) significantly degrade attack effectiveness. As ϵ increases to 60 and 100, attack accuracy steadily improves, indicating that reduced DP noise allows the adversary

Just Accepted



(a) VOC2012. (b) CityScapes. (c) ADE20K.

Fig. 6. F1-score of membership inference attacks under different ϵ in DP. The attack performance improves with the increase of ϵ.

to extract more reliable information from the model outputs. A similar trend is observed for *DeepLabV3+*, where the attack success rate increases as ϵ grows, reinforcing the relationship between noise intensity and attack effectiveness. For the CityScapes dataset, the impact of ϵ follows a similar pattern but is less pronounced compared to VOC2012. Notably, *DeepLabV3+* demonstrates weaker resistance to attacks at higher ϵ values, particularly for ResNet50 and ResNet101, which achieve 0.6641 and 0.6677 at $\epsilon = 100$. This suggests that for CityScapes, DP noise has a more consistent effect across models. Still, its diminishing impact at higher ϵ values indicates that excessive noise removal does not necessarily benefit all datasets equally. For the ADE20K dataset, the effect of DP is most significant. However, at $\epsilon = 100$, the ResNet101's F1-score reaches 0.5760, indicating that excessive DP noise removal increases vulnerability. MobileNetV2 and ResNet50 in *DeepLabV3+* outperform those in *DeepLabV3*, further supporting that the more complex model architectures are more sensitive to $\mathcal{A}_{inf}$. Therefore, considering the trade-off between attack effectiveness and computational efficiency, we select $\epsilon = 60$ as the most balanced and representative setting for subsequent experiments. This choice ensures that the model retains sufficient privacy protection while allowing for meaningful attack analysis, making it an optimal setting for further evaluation.

Table 6 presents the standard deviations of F1-score for applying DP ($\epsilon = 60$) to $\mathcal{A}_{inf}$ among 5 times of trials. The results display significant decreases in attack performance across all models, demonstrating the effectiveness of DP in mitigating $\mathcal{A}_{inf}$.

Datasets	DeepLabV3			DeepLabV3+		
	MobileNetV2	ResNet50	ResNet101	MobileNetV2	ResNet50	ResNet101
VOC2012	0.4788±0.02	0.4992±0.04	0.6732±0.02	0.4210±0.01	0.4924±0.01	0.4430±0.03
CityScapes	0.3557±0.06	0.4110±0.05	0.5144±0.01	0.4383±0.03	0.5205±0.04	0.5000±0.00
ADE20K	0.5166±0.03	0.6495±0.04	0.4401±0.05	0.4891±0.03	0.5001±0.05	0.6604±0.03

Table 6. F1-score of membership inference attack under DP with $\epsilon = 60$.

For the *DeepLabV3* model, the application of DP leads to a noticeable reduction in F1-score across all datasets, particularly in CityScapes. On the VOC2012 dataset, the average decrease is 0.1665 in the F1-score, in which the F1-score of MobileNetV2 drops 0.4992. Similarly, the F1-score for ResNet50 and ResNet101 also show a decline, which indicates that DP has effectively obfuscated the model's predictions, making it more challenging for the

Just Accepted

$\mathcal{A}_{inf}$ to distinguish between training and non-training data. On the CityScapes dataset, the effect of DP is more pronounced, and the average reduction is 0.5339 in the F1-score. Compared to the F1-score in ResNet50, ResNet101 exhibits higher performance, but still greatly mitigates the $\mathcal{A}_{inf}$. For the ADE20K dataset, MobileNetV2 and ResNet101 yield relatively low F1-score (0.5166 and 0.4401, respectively), but ResNet50 stands out with the highest F1-score (0.6495), which might suggest that ResNet50 retains clearer features under DP, making it more susceptible to $\mathcal{A}_{inf}$. These results indicate that applying DP significantly reduces the effectiveness of $\mathcal{A}_{inf}$ across all datasets.

For the *DeepLabV3+* model, the impact of DP follows a similar trend. On VOC2012, MobileNetV2 and ResNet50 exhibit more variation compared to *DeepLabV3*, and the average decline in F1-score is 0.3068. However, ResNet101 exhibits a higher accuracy compared to *DeepLabV3*, despite a drop in F1-score to 0.4430, which suggests that even under DP, *DeepLabV3+* with ResNet101 still allows $\mathcal{A}_{inf}$'s impact, though it also introduces greater classification uncertainty. For ADE20K, *DeepLabV3+* produces the lowest F1-scores for MobileNetV2 (0.4891) but higher than the lowest F1-score compared to *DeepLabV3*. Besides, the F1-score in ResNet50 and ResNet101 attains 0.5001 and 0.6604, respectively, where the upward trend in the F1-score suggests that in ADE20K, as the complexity of the backbone increases, the model remains more vulnerable to $\mathcal{A}_{inf}$ under DP. Overall, compared to $\mathcal{A}_{inf}$ without defense strategies, both *DeepLabV3* and *DeepLabV3+* exhibit a noticeable decline in attack performance.

However, the extent of this decline varies across datasets. Specifically, in VOC2012, *DeepLabV3+* experiences a more significant drop than *DeepLabV3*, but exhibits a smaller decline in $\mathcal{A}_{inf}$ performance after DP is introduced compared to CityScapes and ADE20K. This indicated that simpler datasets with less fine-grained detail are less susceptible to the obfuscation effects of DP. Conversely, on the more detailed CityScapes and ADE20K datasets, *DeepLabV3* suffers greater performance degradation, implying that its simpler architecture, while generally more robust, is more sensitive to DP's perturbation effects when fine-grained details are abundant. Besides, the most significant reduction in performance occurs on CityScapes, suggesting the fine-grained details such as urban structures and human figures, which are central to membership inference, are more directly impacted by the noise introduced by DP. The resulting obfuscation of these specific features leads to a sharp reduction in the attack's ability to distinguish between members and non-members. ADE20K exhibits a relatively slightly smaller reduction than CityScapes in attack performance, likely due to its broader scene diversity, which dilutes the effectiveness of both $\mathcal{A}_{inf}$ and DP's perturbation.

Besides, applying DP to the SS models introduces an additional computational overhead. The time complexity of DP is $O(N)$, and the average extra time for generating and applying noise increases across different backbones for each data in VOC2012, CityScapes, and ADE20K are 3.1ms, 4.8ms, and 5.0ms, respectively. However, this trade-off is expected, as DP enhances privacy protection while maintaining acceptable computational efficiency.

In summary, while the overall reductions in attack performance under DP for *DeepLabV3* and *DeepLabV3+* are comparable, the dataset characteristics are also an influential factor. Specifically, structured and detail-rich datasets like CityScapes experience a more pronounced decline in $\mathcal{A}_{inf}$ effectiveness, whereas simpler datasets such as VOC2012 exhibit relatively smaller reductions, indicating that the inherent characteristics of the dataset play a critical role in attack performance. Furthermore, applying DP introduces an additional computational overhead, where the average extra time for generating and applying noise increases with the volume of datasets. This overhead can affect real-time processing in autonomous vehicles, where low-latency decision-making is essential for safe driving. Therefore, while DP offers privacy benefits, its computational costs and effectiveness vary depending on the dataset complexity and the specific requirements of autonomous driving systems.

Model Pruning. We experiment with different pruning thresholds (τ) set to 0.3, 0.5, and 0.7 to evaluate the defense efficacy. The selection of the thresholds is motivated by the need to balance model performance and privacy protection. Specifically, the lower pruning rate ($\tau = 0.3$) can preserve more parameters, maintaining higher segmentation accuracy, the higher pruning rates ($\tau = 0.7$) further reduce the risk of privacy leakage by

Just Accepted

DeepLabV3												
	0.3				0.5				0.7			
Datasets	MN mIoU	R50 mIoU	R101 mIoU	time	MN mIoU	R50 mIoU	R101 mIoU	time	MN mIoU	R50 mIoU	R101 mIoU	time
VOC2012	0.5413	0.6075	0.6345	0.35	0.4081	0.6218	0.6441	0.60	0.1535	0.5931	0.5964	0
CityScapes	0.4531	0.5453	0.5492	2.43	0.3907	0.5660	0.5355	3.65	0.2796	0.5026	0.5384	2.30
ADE20K	0.3103	0.3100	0.3139	3.39	0.2918	0.3118	0.3357	3.33	0.2509	0.3111	0.3357	2.88
Param(M)	1.52	11.87	17.56	-	2.54	19.79	29.26	-	3.55	27.71	40.97	-
DeepLabV3+												
	0.3				0.5				0.7			
Datasets	MN mIoU	R50 mIoU	R101 mIoU	time	MN mIoU	R50 mIoU	R101 mIoU	time	MN mIoU	R50 mIoU	R101 mIoU	time
VOC2012	0.5500	0.6153	0.6594	0.18	0.4256	0.6157	0.6594	0.52	0.4399	0.6137	0.6415	0
CityScapes	0.4994	0.5739	0.6035	2.68	0.4100	0.5863	0.6069	3.64	0.2921	0.5629	0.5939	1.72
ADE20K	0.2967	0.3325	0.3249	3.98	0.2535	0.3245	0.3235	2.79	0.2930	0.3263	0.3235	3.30
Param(M)	1.56	11.91	17.59	-	2.59	19.85	29.32	-	3.63	27.79	41.05	-

Table 7. The ablation study of mIoU performance of pruned *DeepLabV3* and *DeepLabV3+* at three pruning thresholds (0.3, 0.5, and 0.7). **Time**: The average extra time (ms) cost per data sample during training after model pruning. **Param**: The change in the number of activated model parameters before and after pruning. **MN**: MobileNetV2, **R50**: ResNet50, **R101**: ResNet101.

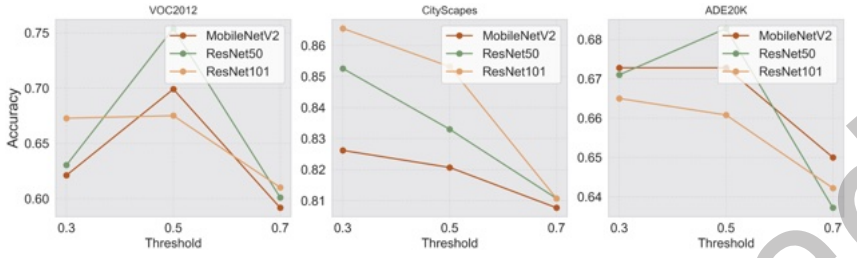
Just Accepted

aggressively eliminating redundant weights, and the intermediate threshold ($\tau = 0.5$) is chosen as a trade-off point.

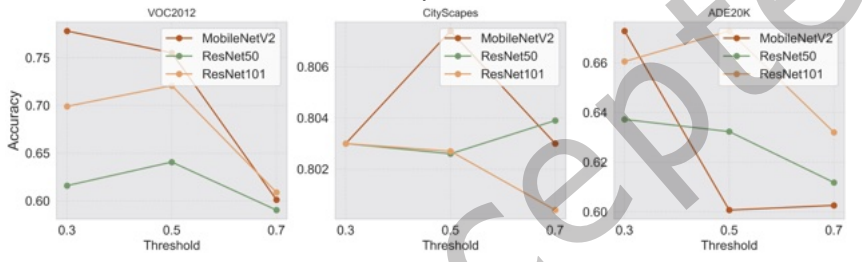
Table 7 presents an ablation study of mIoU performance, extra time cost, and the number of activated parameter changes of pruned *DeepLabV3* and *DeepLabV3+* across backbones (MobileNetV2, ResNet50, and ResNet101) at three pruning thresholds τ (0.3, 0.5, and 0.7). As the pruning threshold increases from 0.3 to 0.7, there is a general decline in the mIoU for both models across all datasets. However, some models reach the best performance when $\tau = 0.5$, especially for the ResNet50 backbone, as under this threshold, it may remove redundant weights and retain important connections. Besides, *DeepLabV3+* generally maintains a higher mIoU than *DeepLabV3* due to its complexity, though pruning still reduces its performance, particularly for MobileNetV2. For the CityScapes dataset, the mIoU drops from 0.4994 at $\tau = 0.3$ to 0.2921 at $\tau = 0.7$, highlighting the significant negative effect of pruning at higher thresholds. This indicates that while pruning reduces model size and complexity, it comes at the cost of segmentation accuracy, especially with higher pruning thresholds.

Regarding the extra time cost, it is slightly higher when $\tau = 0.5$, because pruning at this threshold removes some connections but still retains enough complexity to require more training iterations to compensate for the information loss. The increase in time suggests that pruning may lead to slightly less efficient training due to the fine-tuning stage for recovering the model performance and the additional computational overhead required to handle sparse structures. Furthermore, the number of activated parameters decreases as the pruning threshold increases. For both *DeepLabV3* and *DeepLabV3+*, the number of changed parameters for MobileNetV2 is significantly lower compared to ResNet50 and ResNet101, reflecting the different levels of model complexity. The decrease in activated parameters in *DeepLabV3+* is slightly higher than in *DeepLabV3*, indicating that the degree of reduction is also influenced by the model's architecture.

In summary, while pruning generally reduces the number of activated parameters, it gradually decreases mIoU performance with higher pruning thresholds and slightly increases the time cost. This highlights the trade-off between reducing model size and maintaining performance in segmentation tasks. *DeepLabV3+* consistently presents less robust than *DeepLabV3* in terms of mIoU, but the gap narrows as pruning thresholds increase, reflecting the greater sensitivity of more complex models to pruning.



(a) DeepLabV3.



(b) DeepLabV3+.

Fig. 7. The ablation study of the membership inference attack accuracy of the pruned models among the three pruning thresholds.

Figure 7 displays the $\mathcal{A}_{inf}$ accuracy for pruned models under different τ, with results for *DeepLabV3* and *DeepLabV3+* models across three datasets: VOC2012, CityScapes, and ADE20K. For the *DeepLabV3* model, pruning results in varying effects on $\mathcal{A}_{inf}$ accuracy depending on the backbone used and the dataset. On VOC2012, the accuracy decreases with increasing τ. Notably, ResNet50's accuracy increases at higher pruning thresholds, peaking at 0.7540 at a $\tau = 0.5$, suggesting a more resilient model for pruning. For CityScapes, the trend remains similar, with MobileNetV2 showing the lowest accuracy and ResNet101 the highest at 0.8655 at $\tau = 0.3$. Interestingly, ResNet-based models again show significant increases in performance as pruning increases, reaching the highest accuracy at $\tau = 0.7$.

For the *DeepLabV3+* model, pruning leads to similar trends, but the overall defense effect is slightly stronger across all datasets compared to *DeepLabV3*. On VOC2012, MobileNetV2 shows the largest drop. ResNet50 and ResNet101 display more stable and similar performance, whereas ResNet50 and ResNet101 achieve the highest attack accuracy, 0.6407 and 0.7208, at $\tau = 0.5$ before dropping, suggesting they retain useful features. On CityScapes, the pruned *DeepLabV3+* models exhibit more robust against $\mathcal{A}_{inf}$, reflecting in the lower attack performance than *DeepLabV3*. ResNet101 shows steady declines, reaching peak accuracy values at $\tau = 0.3$. A different result is displayed in ResNet50, where the highest performance achieves 0.8039 when $\tau = 0.7$, slightly higher than $\tau = 0.5$. It may be attributed to a balance between pruning away noisy or overfitted features at $\tau = 0.5$ and forcing the model to generalize better by removing less relevant details at $\tau = 0.7$, which likely results in a more generalized model that focuses on key features, improving its performance compared to the slightly more

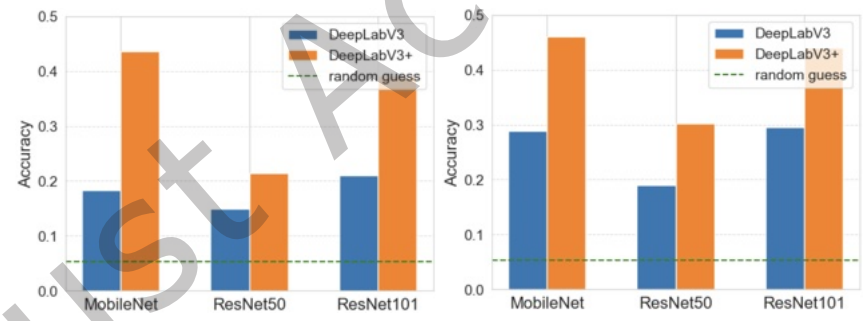
Just Accepted

overfitted state at $\tau = 0.5$. On ADE20K, the significant drop in $\mathcal{A}_{inf}$ performance is in MobileNetV2 at $\tau = 0.5$. *DeepLabV3+* has a more complex encoder-decoder architecture, which is more sensitive to pruning, especially for lightweight models like MobileNetV2, which may be more susceptible to pruning in complex architectures, thus weakening the ability to attack models. Additionally, more complex backbones like ResNet50 and ResNet101 exhibit less sensitivity to pruning because of their greater capacity to learn and generalize, which results in more stable performance across pruning thresholds.

Overall, with the ascend of the pruning threshold, the accuracy of $\mathcal{A}_{inf}$ shows a significant decline, which indicates that a higher pruning rate can effectively weaken the inference ability. At $\tau = 0.5$, the accuracy of some $\mathcal{A}_{inf}$ peaks but declines significantly at $\tau = 0.7$, proving that pruning may make the model parameters more concentrated, temporarily exposing inferring patterns but weakening them with further pruning. In summary, a higher pruning rate can significantly improve the defense against $\mathcal{A}_{inf}$, but due to its redundancy reduction nature, it needs to prevent the selected threshold from being beneficial to attacks. Additionally, VOC2012 shows the most obvious decline in $\mathcal{A}_{inf}$, which may be attributed to its small dataset size and fewer categories, resulting in the models are more likely to lose discriminative features after pruning, thus reducing the attack performance. In autonomous driving, this highlights the need for balancing pruning rates and dataset complexity to achieve the best trade-off between privacy protection and model performance, ensuring that the model can perform real-time, accurate semantic segmentation tasks while minimizing privacy risks.

5.2 Attribute Inference Attacks

5.2.1 Attack. As the nature of $\mathcal{A}_{att}$ needs the datasets to contain the hidden features, therefore we only evaluate $\mathcal{A}_{att}$ on the CityScapes, in which the city names as the hidden attribute. Figure 8 provides a comparison of $\mathcal{A}_{att}$ accuracy under black-box and white-box settings for *DeepLabV3* and *DeepLabV3+*, across three different backbones: MobileNetV2, ResNet50, and ResNet101. The criteria for a successful attack is the attack accuracy surpasses the random guessing rate, which is 0.0556, as CityScapes included 18 cities.



(a) Black-box. (b) White-box.

Fig. 8. The attribute inference attack accuracy in black-box and white-box settings. The random guessing is the inverse of the total number of cities.

The accuracy in all cases is well above the threshold of 0.0556 for random prediction, suggesting that the $\mathcal{A}_{att}$ can learn valid patterns from the data rather than purely random guessing. In the black-box setting, where only the $\boldsymbol{M}$ are accessible, the performance of *DeepLabV3+* significantly outperforms *DeepLabV3* across all backbones. Specifically, for MobileNetV2, *DeepLabV3+* achieves the highest accuracy at 0.4358, which is over double that of *DeepLabV3*. Similarly, for ResNet50 and ResNet101, *DeepLabV3+* again shows higher attack accuracy compared to the lower values of *DeepLabV3*. The sharp increase in accuracy for *DeepLabV3+* attributes to its more complex

Just Accepted

architecture, with an encoder-decoder structure to optimize multi-scale feature fusion further, which potentially captures more features that $\mathcal{A}_{att}$ can exploit.

In the white-box setting, the gap between *DeepLabV3* and *DeepLabV3+* is less pronounced compared to the black-box setting, indicating that when the adversary has access to extensive knowledge of the model, the attack performance depends more on the leaked information rather than the architecture of the model. Specifically, the accuracy for MobileNetV2 is 0.2881 for *DeepLabV3* and 0.4610 for *DeepLabV3+*, and ResNet50 and ResNet101 exhibit more consistent results for both models, with *DeepLabV3+* achieving 0.3025 for ResNet50 and 0.4406 for ResNet101. In this scenario, both models seem to perform similarly as they are more directly accessible for $\mathcal{A}_{att}$ to interpret and leverage $\boldsymbol{M}$ and $\boldsymbol{H}$ (internal features), which also prove that except for output probability, other features can also provide information for inference attacks.

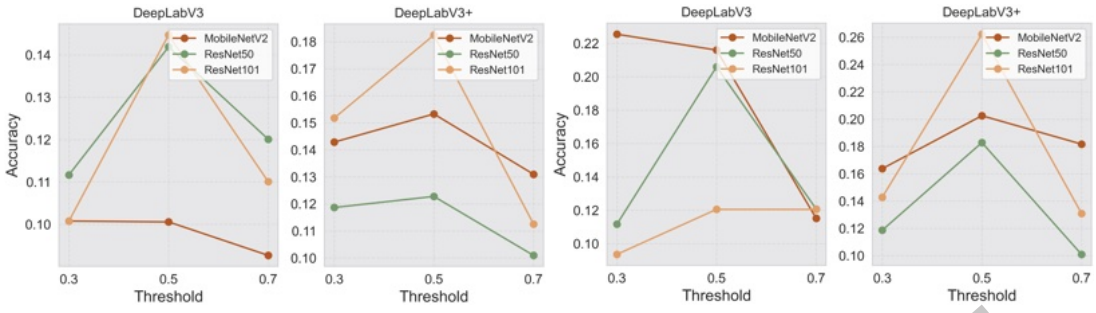
Although space improvement exists in overall accuracy, it is quite a challenging task to infer the city where the picture is based on the city's architecture, especially in the SS task of autonomous driving. Even advanced DL models may struggle to accurately distinguish 18 different cities from architectural features, especially since they may have similarities in architectural style. Besides, the images in the dataset can contain a lot of background information and noise, making it difficult for the model to focus on the architectural features associated with the city. However, this challenge highlights a significant privacy threat in autonomous driving systems: even with complex models, there is a potential for inferring sensitive information. Therefore, while inferring the city from architectural features is challenging, our attack successfully demonstrates the feasibility, underscoring the importance of addressing privacy concerns in autonomous driving systems.

5.2.2 Defense. **DP.** The ablation study presented in Table 8 analyzes the accuracy of $\mathcal{A}_{att}$ under applied DP efficacy for three different ϵ values (20, 60, and 100) in both black-box and white-box settings. In general, as ϵ increases, we observe a trend where the performance of the models tends to improve, indicating that with lower ϵ values, the privacy protection is stronger, making it more challenging to infer L. The best performance is mainly exhibited in ResNet101, while the worst cases appear in MobileNetV2 and ResNet50 due to the difference in model capacity and memorization ability. ResNet101 is a deeper and more powerful model that is more prone to overfitting the training data. Thus, DP noise has a more noticeable effect in mitigating this overfitting, improving its robustness. Besides, ResNet50 exhibits a less dramatic change. For *DeepLabV3*, the accuracy fluctuates slightly across ϵ values (0.1205 at $\epsilon = 20$ to 0.1279 at $\epsilon = 100$), indicating that ResNet50 is more resilient to changes in privacy budget. Additionally, in the white-box setting, the overall performance is higher than the black-box setting, suggesting that the feature $\boldsymbol{H}$ as the extra knowledge used in the white-box setting is also sensitive to DP and that having access to intermediate features (like color histograms) provides more information for the attack, making it more susceptible to inference when DP is applied to the $\boldsymbol{M}$ only.

Backbones	Black-box						White-box					
	DeepLabV3			DeepLabV3+			DeepLabV3			DeepLabV3+		
	20	60	100	20	60	100	20	60	100	20	60	100
MobileNetV2	0.1122	0.1179	0.1650	0.1348	0.1608	0.2781	0.1004	0.1257	0.2367	0.1662	0.1867	0.3389
ResNet50	0.1205	0.1206	0.1279	0.1109	0.1263	0.1745	0.1206	0.1204	0.1308	0.2113	0.2425	0.2508
ResNet101	0.1225	0.1232	0.1237	0.2525	0.2893	0.3007	0.1991	0.2069	0.2573	0.2877	0.2953	0.3710

Table 8. The ablation study on the accuracy of attribute inference attacks under the applied DP efficacy at three different ϵ values (20, 60, and 100). The gray background color emphasizes the best performance within the same model but with different backbones and ϵ, and the worst performance values are boxed.

Just Accepted



(a) Black-box. (b) White-box.

Fig. 9. The attribute inference attack accuracy of the pruned models among the three pruning thresholds (0.3, 0.5, and 0.7).

Model Pruning. In this attack, the pruned model is utilized to infer the city attribute according to the buildings of the CityScapes datasets, whose performance is displayed in Figure 9. Compared with original models, the accuracy of pruned models in $\mathcal{A}_{att}$ decreases substantially and generally declines with ascending thresholds. In black-box and white-box attacks, the accuracy of $\mathcal{A}_{att}$ peaks when $\tau = 0.5$ and rapidly declines when $\tau = 0.7$. This might be because pruning typically removes weights that have less impact on the final output, making the remaining weights more focused on expressing the model's core features. Specifically, ResNet101 reaches the highest in both *DeepLabV3* (0.1447) and *DeepLabV3+* (0.1825) at $\tau = 0.5$, in which the core information helps $\mathcal{A}_{att}$ to capture useful features easily, thereby causing the climbing accuracy. However, in the white-box setting, MobileNetV2 and ResNet50 reach a higher accuracy than ResNet101 in *DeepLabV3* when $\tau = 0.5$.

In the black box attack, as only the $\boldsymbol{M}$ is trained, ResNet-based models retain more semantic information, making the attack more successful, while MobileNetV2 results in less output information and the lowest attack performance due to the lightweight model. While in white-box attacks, due to the introduction of $\boldsymbol{H}$, the lighter network (MobileNetV2, ResNet50) is more susceptible to $\mathcal{A}_{att}$, while ResNet101 is more global in feature extraction, which reduces the performance of attack. ResNet50 is the most sensitive to model pruning in *DeepLabV3+* in both black-box and white-box settings, as ResNet50 is the mid-depth model compared to MobileNetV2 and ResNet101, with the most loss of key features after pruning, therefore the inference attack accuracy drops most significantly. In summary, these findings suggest that attack performance varies with model complexity, where ResNet-based models retain more semantic information in black-box attacks, while lighter networks are more susceptible in white-box settings, and ResNet50 is the most affected by pruning due to its mid-depth architecture. This underscores the importance of carefully considering model architecture and pruning strategies in designing secure perception modules for autonomous driving systems, where privacy and performance should be balanced effectively.

Datasets	DeepLabV3						DeepLabV3+					
	MobileNetV2		ResNet50		ResNet101		MobileNetV2		ResNet50		ResNet101	
	Acc.	loss	Acc.	loss	Acc.	loss	Acc.	loss	Acc.	loss	Acc.	loss
VOC2012	0.6637	0.2885	**0.6852**	**0.2597**	0.6831	0.2916	0.6687	0.2741	**0.6876**	**0.2628**	0.6833	0.2671
CityScapes	0.7961	0.2231	0.8442	**0.1824**	**0.8465**	0.1878	0.8280	0.1919	0.8422	**0.1886**	**0.8503**	0.2146
ADE20K	0.6426	0.2443	0.6580	0.2359	**0.6606**	**0.2288**	0.6356	0.2410	**0.6593**	**0.2321**	0.6564	0.2327

Table 9. Accuracy and Canny-loss of model inversion attack in three datasets.

Just Accepted

5.3 Model Inversion Attack

5.3.1 Attack. The accuracy and L_{Canny} of $\mathcal{A}_{inv}$ is present in Table 9, which displays an upward trend of accuracy with the increased complexity of backbones. For the *DeepLabV3* model, MobileNetV2 performs relatively worse than ResNet-based models across the datasets, indicating that ResNet-based SS models extract features more effectively for $\mathcal{A}_{inv}$. Specifically, the average accuracy and L_{Canny} of ResNet101 are 0.7301 and 0.2361, while in MobileNetV2 are 0.7008 and 0.2510, respectively. In the *DeepLabV3+* model, ResNet101 achieves the highest accuracy (0.8503) on CityScapes despite a relatively high L_{Canny}, emphasizing that while more complex models offer improved segmentation accuracy, also carry greater risks for privacy breaches, particularly in urban environments common in autonomous driving.

In addition to model architectures, the performance across different datasets also reveals significant differences. On the VOC2012 dataset, which is simpler with fewer categories, ResNet50 achieves relatively high accuracy. On the more complex CityScapes dataset, ResNet-based models both obtain good performance in accuracy (0.8442 for *DeepLabV3* and 0.8503 for *DeepLabV3+*), suggesting that the deeper networks are better equipped to handle complex urban scenes, leading to more effective model inversion. However, on ADE20K, which presents the most challenging and diverse scenes, the performance differences between backbones become less pronounced, with ResNet101's accuracy being 0.6606 for *DeepLabV3* and 0.6564 for *DeepLabV3+*, which remains relatively lower across all datasets, indicating that the complexity of ADE20K poses a challenge to achieve high inversion accuracy.

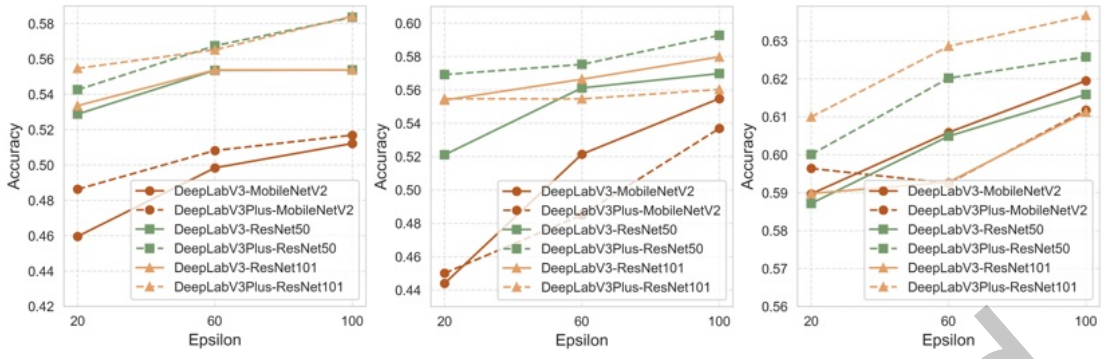
5.3.2 Defense. **DP.** In the $\mathcal{A}_{inv}$, DP is applied to the intermediate layer outputs of SS models to mitigate the performance of $\mathcal{A}_{inv}$. We conduct an ablation study on the accuracy of $\mathcal{A}_{inv}$ under different ϵ, as presented in Figure 10. The results demonstrate that $\mathcal{A}_{inv}$ is sensitive to the noise injected by DP. Specifically, it can be observed that as ϵ increases, the noise scale decreases, and then the accuracy of the attack generally improves. Besides, the increase is gentler because lower noise levels (higher ϵ) cause the attack model to better distinguish between the real features and the noise, which enhances its ability to recover the original data. However, as ϵ continues to increase, the sensitivity of the attack model reaches a bottleneck, and further improvements in accuracy become less noticeable. Overall, the results suggest an inverse relationship between the noise intensity of DP and the attack accuracy, with the attack performance improving as ϵ increases. This trend implies that when applying DP, the noise strength should be balanced between providing adequate privacy protection and defending against attacks. It highlights the importance of carefully selecting ϵ to ensure an optimal trade-off between privacy and attack resistance.

Model Pruning. In the results presented in Figure 11, we observe the performance of $\mathcal{A}_{inv}$ with different pruning thresholds across three datasets for both *DeepLabV3* and *DeepLabV3+* models. Despite the increase when $\tau = 0.5$, the accuracy of $\mathcal{A}_{inv}$ decreases as the τ increases. For *DeepLabV3*, MobileNetV2 shows the most substantial decrease in accuracy, dropping from approximately 0.6637 to 0.5039 on VOC2012 when $\tau = 0.3$, suggesting that pruning significantly impacts the lightweight model. For *DeepLabV3+*, MobileNetV2 also displayed the most significant reduction on VOC2012 and ADE20K when $\tau = 0.3$, the decline rate are 0.1559 and 0.0760, respectively. While in both *DeepLabV3* and *DeepLabV3+*, ResNet50 pruning models can mitigate $\mathcal{A}_{inv}$ against CityScapes most effectively when $tau = 0.5$. ResNet50 retains sufficient detail, unlike MobileNetV2, while avoiding the overfitting seen in ResNet101, which allows ResNet50 to generalize better and resist $\mathcal{A}_{inv}$ in CityScapes.

6 DISCUSSION

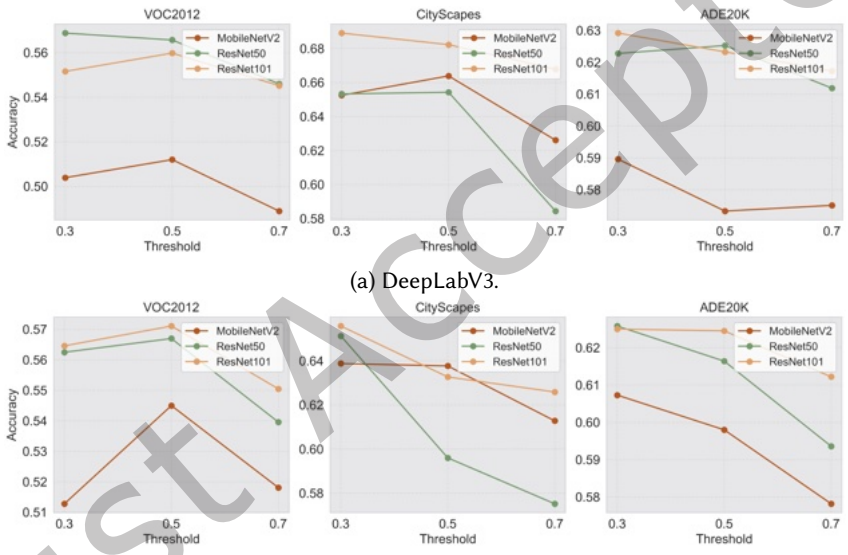
In the context of 2D RGB-based SS tasks for autonomous driving perception modules, membership inference attacks, attribute inference attacks, and model inversion attacks present certain strengths and limitations that should be considered for practical deployment.

Just Accepted



(a) VOC2012. (b) CityScapes. (c) ADE20K.

Fig. 10. The accuracy of model inversion attacks under different ϵ in DP. The attack performance improves with the increase of ϵ.



(a) DeepLabV3.

(b) DeepLabV3+.

Fig. 11. The model inversion attack accuracy of the pruned models.

First, membership inference attacks, determine whether specific data points belong to the training set of the victim model and are valuable in identifying privacy vulnerabilities in autonomous driving systems. These attacks can reveal sensitive information about private data and its underlying distribution. Successfully executing these attacks in autonomous driving systems can expose critical information about the vehicle's operating environment, potentially revealing patterns that could compromise user privacy or security. However, it faces challenges in real-world autonomous driving environments. While this attack can be effective, its general applicability is constrained by the need for a shadow model, which must closely mimic the victim model. The success of the attack is also influenced by the data distribution, particularly when there is a large sample bias in the training

Just Accepted

data. In practice, the training data for autonomous driving systems can be diverse, and such biases may limit the attack's effectiveness across different scenarios.

Attribute inference attacks in autonomous driving systems hold significant potential for inferring sensitive, hidden attributes, such as city labels, specific traffic conditions, or environmental factors, from publicly available model outputs. These attacks provide an opportunity to gain deeper insights into the underlying data, which could potentially expose critical information related to the vehicle's operating environment or user behavior. Such information, if exploited, could lead to significant privacy concerns, especially if the attacker gains insights into a vehicle's location, travel patterns, or even the behavior of pedestrians and other road users. However, these attacks face challenges in deployment due to the limited availability of certain attributes in real-world data. For example, not all attributes may be explicitly labeled, and certain cities or specific driving conditions may not have sufficient data to infer the target attribute reliably. In cases where there is limited attribute information, the difficulty of executing this attack increases, limiting its effectiveness in certain autonomous driving contexts.

Model inversion attacks have the potential to significantly reconstruct sensitive input data, such as vehicle sensor inputs or environmental details, from a model's internal features or outputs. This can be particularly impactful for malicious actors attempting to reverse-engineer proprietary driving data or sensitive information related to specific driving conditions, road layouts, or private journeys. The ability to extract such private information poses significant privacy risks, especially in real-time autonomous driving applications where the system continuously interacts with the surrounding environment. However, this attack is constrained by the complexity of real-world autonomous driving systems. While these attacks can be effective under ideal conditions, the high dimensionality of the feature space in deep learning models can hinder accurate data reconstruction, especially when the data is noisy or highly variable, as is often the case with real-time sensor inputs in autonomous driving.

Regarding defense mechanisms, DP and model pruning offer certain advantages but also present practical limitations. In 2D SS tasks for autonomous driving systems, DP is an effective defense technique that allows the model to maintain a high level of confidentiality while still performing reasonably, but it comes with limitations in terms of performance degradation and computational overhead. A higher privacy budget may result in insufficient privacy protection, while a lower privacy budget can significantly reduce model performance, especially in high-precision tasks such as SS, where the model's inference ability and prediction accuracy are compromised. Furthermore, the computational cost of applying DP can cause linear increases in inference time, which may slightly impact real-time processing requirements in autonomous driving systems.

Model pruning, which reduces model complexity by removing unimportant weights and neurons, improves inference efficiency and risks accuracy degradation. This reduction helps in improving inference efficiency and reducing computational load, which is particularly useful in real-time autonomous driving tasks where computational efficiency is crucial. However excessive pruning could lead to losing important features necessary for high-precision tasks like SS in autonomous driving applications, potentially affecting the model's ability to detect and segment critical objects in real-time. Careful pruning is required to avoid compromising model performance, especially in safety-critical driving tasks.

In summary, while membership inference, attribute inference, and model inversion attacks represent significant privacy threats to autonomous driving systems, their practical deployment is subject to limitations related to data diversity, attribute availability, and model complexity. Similarly, while DP and model pruning can offer privacy and efficiency benefits, their real-world application must strike a careful balance between privacy, performance, and computational efficiency, especially for real-time tasks in autonomous driving.

Just Accepted

7 CONCLUSION

In conclusion, we investigate the holistic privacy risks associated with RGB image-based 2D semantic segmentation (SS) models, focusing on different backbones by assessing their susceptibility to membership inference attacks, attribute inference attacks, and model inversion attacks. This study addresses two key gaps in the existing literature: the limited evaluation of attribute inference and model inversion attacks against 2D SS models and the insufficient defense strategies against these types of inference attacks. To fill these gaps, we *(1)* provide comprehensive evaluations of three inference attacks across commonly-used SS models on different datasets, and *(2)* assess general privacy-preserving techniques, including differential privacy (DP) and model pruning, against these three attacks in 2D SS models. Our findings demonstrate that both DeepLabV3 and DeepLabV3+ are vulnerable to various inference attacks, particularly when trained with different types of information. More complex models, such as DeepLabV3+ with the ResNet101 backbone, are especially prone to inference attacks, which pose significant privacy risks for autonomous driving systems.

In real-world autonomous driving applications, these vulnerabilities are concerning because the high accuracy of membership inference attacks, over 30% higher than the threshold, with values of 0.8693 for DeepLabV3 and 0.8813 for DeepLabV3+ across datasets—indicates a substantial risk of privacy breaches. The attack performance in attribute inference attacks peaks at 0.4610 on the VOC2012 dataset, and model inversion attacks achieve over 80% accuracy on the CityScapes dataset, further highlighting the potential for adversaries to infer sensitive information, such as specific urban environments or driving scenarios. These privacy risks are particularly critical in autonomous driving, where the models constantly process real-world data from vehicles in urban settings.

To mitigate these threats, we introduce DP and model pruning as defense strategies. These techniques show promising results, reducing the performance of the attacks by over 20% across all types of inference attacks. However, while DP effectively reduces privacy risks, it also introduces computational overhead, which must be carefully managed in real-time autonomous driving systems that require low-latency processing. Similarly, model pruning offers a more lightweight solution. Still, it should be applied to avoid compromising model accuracy, especially for more complex environments like urban scenes.

In future work, we will explore how to balance the trade-off between DP and data availability, developing more adaptive DP technologies that better address privacy concerns in autonomous driving systems. Moreover, model pruning could be optimized by dynamically adjusting the pruning threshold based on the model's training progress or performance, helping to find a balance between model efficiency, accuracy, and privacy protection.

Just Accepted

REFERENCES

[1] Andreas Aakerberg, Anders S Johansen, Kamal Nasrollahi, and Thomas B Moeslund. 2022. Semantic segmentation guided real-world super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 449–458.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1897–1914.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.

[6] Tomas Chobola, Dmitrii Usynin, and Georgios Kaissis. 2023. Membership inference attacks against semantic segmentation models. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. 43–53.

[7] Tianyue Chu, Mengwei Yang, Nikolaos Laoutaris, and Athina Markopoulou. 2023. PriPrune: Quantifying and Preserving Privacy in Pruned Federated Learning. *arXiv preprint arXiv:2310.19958* (2023).

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, et al. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

[9] Torbjørn Eltoft, Taesu Kim, and Te-Won Lee. 2006. On the multivariate Laplace distribution. *IEEE Signal Processing Letters* 13, 5 (2006), 300–303.
[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2012. The pascal visual object classes challenge 2012 (voc2012). In *Results*.
[11] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.
[12] Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574* (2019).
[13] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. 2017. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857* (2017).
[14] Trung Ha, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, and Manh Tuan Nguyen. 2019. Differential privacy in deep learning: an overview. In *2019 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 97–102.
[15] Qilong Han, Zuobin Xiong, and Kejia Zhang. 2018. Research on trajectory data releasing method via differential privacy based on spatial partition. *Security and Communication Networks* 2018, 1 (2018), 4248092.
[16] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems* 28 (2015).
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[18] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. 2020. Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 519–535.
[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
[20] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.
[21] Rui Hu, Yuanxiong Guo, and Yanmin Gong. 2023. Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy. *IEEE Transactions on Mobile Computing* (2023).
[22] Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. 2022. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2022), 10374–10386.
[23] Hyeokjun Kweon and Kuk-Jin Yoon. 2022. Joint learning of 2D-3D weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems* 35 (2022), 30499–30511.
[24] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. 2021. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* 23, 7 (2021), 8423–8434.
[25] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2022. {ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium (USENIX Security 22)*. 4525–4542.
[26] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270* (2018).
[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.
[28] Shagufta Mehnaz, Sayanton V Dibbo, Roberta De Viti, Ehsanul Kabir, Björn B Brandenburg, Stefan Mangard, Ninghui Li, Elisa Bertino, Michael Backes, Emiliano De Cristofaro, et al. 2022. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*. 4579–4596.
[29] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 691–706.
[30] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* 493 (2022), 626–646.
[31] Khan Muhammad, Tanveer Hussain, Hayat Ullah, Javier Del Ser, Mahdi Rezaei, Neeraj Kumar, Mohammad Hijji, Paolo Bellavista, and Victor Hugo C de Albuquerque. 2022. Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 22694–22715.
[32] Anh Nguyen, Tuong Do, Minh Tran, Binh X Nguyen, Chien Duong, Tu Phan, Erman Tjiputra, and Quang D Tran. 2022. Deep federated learning for autonomous driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1824–1830.
[33] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*. 1520–1528.
[34] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. 2023. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal*

Just Accepted

of Artificial Intelligence Research 77 (2023), 1113–1201.
[35] Maria Rigaki and Sebastian Garcia. 2023. A survey of privacy attacks in machine learning. *Comput. Surveys* 56, 4 (2023), 1–34.
[36] Giulia Rizzoli, Francesco Barbato, and Pietro Zanuttigh. 2022. Multimodal semantic segmentation in autonomous driving: A review of current approaches and future perspectives. *Technologies* 10, 4 (2022), 90.
[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.
[38] Giulio Rossolini, Federico Nesti, Gianluca D'Amico, Saasha Nair, Alessandro Biondi, and Giorgio Buttazzo. 2023. On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
[39] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*. PMLR, 5558–5567.
[40] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).
[41] Rulin Shao, Hui Liu, and Dianbo Liu. 2019. Privacy preserving stochastic channel-based federated learning with neural network pruning. *arXiv preprint arXiv:1910.02115* (2019).
[42] Kabid Hassan Shibly, Md Delwar Hossain, Hiroyuki Inoue, Yuzo Taenaka, and Youki Kadobayashi. 2023. Towards autonomous driving model resistant to adversarial attack. *Applied Artificial Intelligence* 37, 1 (2023), 2193461.
[43] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.
[44] Irem Ulku and Erdem Akagündüz. 2022. A survey on deep learning-based architectures for semantic segmentation on 2d images. *Applied Artificial Intelligence* 36, 1 (2022), 2032924.
[45] Jia Wang, Chengyu Wang, Qiuzhen Lin, Chengwen Luo, Chao Wu, and Jianqiang Li. 2022. Adversarial attacks and defenses in deep learning for image recognition: A survey. *Neurocomputing* 514 (2022), 162–181.
[46] Yunyan Wang, Chongyang Wang, Huaxuan Wu, and Peng Chen. 2022. An improved Deeplabv3+ semantic segmentation algorithm with multiple loss constraints. *Plos one* 17, 1 (2022), e0261582.
[47] Feng Wu, Lei Cui, Shaowen Yao, and Shui Yu. 2024. Inference Attacks in Machine Learning as a Service: A Taxonomy, Review, and Promising Directions. *arXiv preprint arXiv:2406.02027* (2024).
[48] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. 2021. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7515–7525.
[49] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*. 418–434.
[50] Zuobin Xiong, Zhipeng Cai, Qilong Han, Arwa Alrawais, and Wei Li. 2020. ADGAN: Protect your location privacy in camera data of auto-driving vehicles. *IEEE Transactions on Industrial Informatics* 17, 9 (2020), 6200–6210.
[51] Zuobin Xiong, Wei Li, Qilong Han, and Zhipeng Cai. 2019. Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 668–677.
[52] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 225–240.
[53] Xiaoyong Yuan and Lan Zhang. 2022. Membership inference attacks and defenses in neural network pruning. In *31st USENIX Security Symposium (USENIX Security 22)*. 4561–4578.
[54] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. 2020. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access* 8 (2020), 58443–58469.
[55] Guangsheng Zhang, Bo Liu, Tianqing Zhu, Ming Ding, and Wanlei Zhou. 2022. Label-only membership inference attacks and defenses in semantic segmentation models. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2022), 1435–1449.
[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2881–2890.
[57] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. 2020. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal* 8, 11 (2020), 8836–8853.
[58] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 633–641.
[59] Tianqing Zhu, Dayong Ye, Shuai Zhou, Bo Liu, and Wanlei Zhou. 2022. Label-only model inversion attacks: Attack with the least information. *IEEE Transactions on Information Forensics and Security* 18 (2022), 991–1005.

Received 19 August 2024; revised 28 February 2025; accepted 21 April 2025

Just Accepted