Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(8): 13562–13580.
DOI: 10.3934/mbe.2023605
Received: 08 May 2023
Revised: 19 May 2023
Accepted: 07 June 2023
Published: 14 June 2023

Research article

Attention distraction with gradient sharpening for multi-task adversarial attack

Bingyu Liu, Jiani Hu and Weihong Deng*

Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China

* **Correspondence:** Email: whdeng@bupt.edu.cn.

Abstract: The advancement of deep learning has resulted in significant improvements on various visual tasks. However, deep neural networks (DNNs) have been found to be vulnerable to well-designed adversarial examples, which can easily deceive DNNs by adding visually imperceptible perturbations to original clean data. Prior research on adversarial attack methods mainly focused on single-task settings, i.e., generating adversarial examples to fool networks with a specific task. However, real-world artificial intelligence systems often require solving multiple tasks simultaneously. In such multi-task situations, the single-task adversarial attacks will have poor attack performance on the unrelated tasks. To address this issue, the generation of multi-task adversarial examples should leverage the generalization knowledge among multiple tasks and reduce the impact of task-specific information during the generation process. In this study, we propose a multi-task adversarial attack method to generate adversarial examples from a multi-task learning network by applying attention distraction with gradient sharpening. Specifically, we first attack the attention heat maps, which contain more generalization information than feature representations, by distracting the attention on the attack regions. Additionally, we use gradient-based adversarial example-generating schemes and propose to sharpen the gradients so that the gradients with multi-task information rather than only task-specific information can make a greater impact. Experimental results on the NYUD-V2 and PASCAL datasets demonstrate that the proposed method can improve the generalization ability of adversarial examples among multiple tasks and achieve better attack performance.

Keywords: deep learning; adversarial attack; multi-task; attention heat map; gradient editing

1. Introduction

Over the last decade, deep neural networks (DNNs) have achieved considerable success on a multitude of visual tasks [1–4], such as semantic segmentation [5], monocular depth estimation [6],

surface normal estimation [7], etc. A large number of studies have focused on single-task situations, that is, training a separate neural network for a single task. However, real-world artificial intelligence (AI) problems often require solving multiple learning tasks concurrently. For instance, an autonomous driving system should be able to detect all objects in the driving scene and estimate their distance and trajectory in order to keep the car safely operated in its surroundings. Similarly, a human-computer interaction system should be able to detect the presence of people, estimate the pose, track the hand, etc., in order to analyze human actions and make reactions. Therefore, these problems have motivated researchers to develop generalized deep learning models in which multiple learning tasks can be solved simultaneously.

Multi-task learning (MTL) [8–10] is a technique to train a multi-task network that can infer several desired task outputs with only one input given. MTL generally leverages a shared encoder for all the desired tasks and separate task-specific heads as decoders for different tasks. Such multi-task networks of MTL can reduce the storage cost, as well as the inference time, compared to the single-task case, where each individual task is solved separately by its own network. The reason is that the shared layers result in less memory cost than several single-task networks and avoid repeatedly calculating the features, while the single-task networks are required to calculate once for each task. Thus, developing MTL to learn shared representations from multi-task supervisory signals has become a popular research theme.

Despite the notable achievements of DNNs in visual perception, many recent works have demonstrated that DNNs are vulnerable to adversarial examples [11, 12], which are able to cheat DNNs to produce incorrect predictions in high confidence with imperceptible perturbations added to the clean data. Attack methods to craft adversarial examples have been widely researched for the single-task setting, where adversarial examples are crafted only for a single task. Obviously, these single-task adversarial examples will have poor attack performance on other tasks on which there has been no training. Therefore, existing single-task adversarial attack methods have their limitations in the multi-task setting, which has more practical significance.

There have been few works on adversarial attacks under the multi-task setting. Guo et al. [13] proposed a multi-task adversarial attack method by building a multi-task generator for adversarial perturbations. The generator has a similar architecture to an MTL network consisting of a shared encoder for all tasks and multiple task-specific decoders. That is, the adversarial examples for each task are generated separately through the task-specific decoders. However, using the same input to perform multiple tasks is more common in multi-task problems. Therefore, we concentrate on generating consistent adversarial examples for multiple tasks, i.e., generating one adversarial example for each image to attack an MTL network with multiple tasks. In addition, aiming at one specific class to attack is more rational in real-world applications because many real-world systems will collapse as long as a specific class is attacked. Thus, we focus on class-specific multi-task adversarial attacks which generate adversarial examples for each class contained in each image to make the multiple tasks fail on the class-specific regions.

To this end, we propose a multi-task adversarial attack method that uses attention distraction with gradient sharpening (ADGS). Specifically, in order to improve the generalization capability of the adversarial examples among multiple tasks, we propose an attention-distracting loss to attack the attention heat maps motivated by that attention can contain more generalization information; for instance, the authors of [14] used an attention-based method to fuse the rich semantic information of

heterogeneous nodes, and those of [15] used an attention mechanism to capture the correlation between the spatial traffic flow images' channels. By optimizing the loss, the generated adversarial examples can distract the attention of the multi-task network from the class-specific regions to other regions so that the multiple tasks can fail on the specific class. Further, on the basis of the gradient-based adversarial example-generating schemes, we propose a gradient sharpening method to amplify the influence of gradients that contain multi-task information, rather than solely task-specific information. The main contributions of our paper can be summarized as follows:

- We focus on a significant multi-task adversarial attack problem to generate consistent adversarial examples for a multi-task network. Such adversarial examples can attack multiple tasks with one consistent adversarial example for each image, while existing single-task adversarial examples can only attack one concerned task. Therefore, multi-task adversarial attacks can be more harmful to real-world visual systems, which are commonly multi-task systems.
- We propose an attack method using ADGS for the multi-task setting, which can improve the generalization capability of the consistent adversarial examples among multiple tasks, and therefore realize more effective multi-task adversarial attacks.
- The effectiveness of the proposed ADGS method is empirically demonstrated in experiments on NYUD-V2 [16] and PASCAL [17] datasets to attack a multi-task network with multiple tasks. Our ADGS method can not only obtain a larger average per-task performance drop than the baseline attack methods, but it also has better attack performance on all of the concerned tasks, which indicates the improved generalization capability of ADGS among multiple tasks.

The remainder of this paper is structured as follows. In the next section, we review the literature related to MTL and adversarial attacks. In Section 3, we introduce the proposed ADGS method for class-specific multi-task adversarial attacks. In Section 4, experimental results are shown and we validate the effectiveness of the proposed method. Finally, we conclude this paper and discuss future works in Section 5.

2. Related works

In this section, we briefly review the related literature on MTL and adversarial attacks.

2.1. Multi-task learning

Recently, MTL [8–10] has been widely studied as a result of the extensive application scenarios and the improvement of task performance by leveraging the complementary knowledge from multiple tasks. Before the deep learning era, MTL works aimed at obtaining better generalization performance by implementing joint task learning that models the common information among tasks [18–20]. Along with the development of deep learning, MTL has attracted more attention. In the context of deep learning, MTL is performed by learning shared representations from multi-task supervisory signals and outputting multi-task predictions by using task-specific heads [21–23].

Misra et al. [24] proposed a "cross-stitch" unit in the encoder, which combined the activations from multiple single-task networks and can be trained end-to-end. The cross-stitch networks perform soft parameter sharing in deep MTL architectures, where each task is assigned its own set of parameters and a feature-sharing mechanism handles the cross-task talk. Gao et al. [25] proposed a

similar architecture as a cross-stitch network, named the neural discriminative dimensionality reduction cnn, which incorporated a dimensionality reduction mechanism into the feature fusion layers. Liu et al. [26] proposed a multi-task attention network, which consisted of a single shared backbone network to extract a global feature pool, together with a soft-attention module for each task to select task-specific features from the global pool. Wallingford et al. [27] proposed a task-adaptive parameter sharing method to adaptively select a minimal subset of the existing layers in a pre-trained multi-task network for each task and replace them with task-specific parameters. Soft parameter sharing approaches need to assign quite a few parameters for each task so that the size of the multi-task network tends to grow linearly with the number of tasks. On the other hand, hard parameter sharing is a kind of technique that shares all of the parameters in the multi-task encoder and utilizes unshared parameters in the task-specific heads. Kokkinos [28] proposed the UberNet by constructing an image pyramid to process the multi-resolution versions of the images through shared encoders, which branched out into additional task-specific layers. Chen et al. [22] proposed a gradient normalization (GradNorm) algorithm to control the training of multi-task networks by dynamically tuning gradient magnitudes. Sener and Koltun [23] proposed a multiple-gradient descent algorithm to update the shared multi-task network weights by finding a common descent direction among the task-specific gradients. Ott et al. [29] proposed several strategies to weight the losses for multivariate time series classification and trajectory regression in a multi-task network.

2.2. Adversarial attacks

Szegedy et al. [11] first demonstrated that DNNs can be easily fooled by adversarial examples, which can be generated by adding elaborate and visually imperceptible perturbations to clean images. They used a box-constrained L-BFGS method to calculate adversarial examples. These carefully crafted adversarial examples can fool the DNNs with high probability while appearing indistinguishable from the clean images to the human visual system. Such adversarial attacks on DNNs on image classification tasks have been extensively studied. Compared with the time-consuming L-BFGS attack in [11], Goodfellow et al. [12] proposed a faster method, named the fast gradient sign method (FGSM), which generated adversarial examples by performing one-step updates along the direction of the sign of the gradient at each pixel. Madry et al. [30] further developed the FGSM by using a projected gradient descent (PGD) method which generated adversarial examples by iteratively updating multiple small steps while adjusting the direction after each step. Moosavi-Dezfooli et al. [31] proposed the DeepFool algorithm to generate minimal adversarial perturbations in an iterative manner by moving the data points toward the classification boundary. In contrast to these adversarial input example attacks, another type of adversarial attack pays attention to attacking network weight parameters. Rakin et al. [32] proposed the targeted bit-flip attack method to make a classification network predict some selected images as a target class by flipping a few vulnerable weight bits, which were selected by a searching algorithm.

In addition, adversarial attacks have also been actively investigated beyond the image classification task in computer vision. Xie et al. [33] proposed the dense adversary generation (DAG) algorithm to compute adversarial examples for semantic segmentation or object detection tasks. To generate adversarial examples, the DAG algorithm iteratively optimizes a loss function that targets all pixels for semantic segmentation, or object proposals for object detection, instead of the entire image in image classification. Zhao et al. [34] proposed an AP-GAN attack method for the image-retrieval task, which

uses a generative adversarial network to generate adversarial patches instead of modifying the entire image. As for the face attribute recognition task, Mirjalili and Ross [35] proposed a technique to generate adversarial perturbations for face images so that the adversarial face images can fool a gender classifier while retaining the biometric utility for a face matching system. For the monocular depth estimation task, Yamanaka et al. [36] proposed to generate adversarial patches attached to the clean images that can fool the target models into estimating an incorrect depth for the regions of the patches. Besides, there are other types of attack and defense methods focusing on data security. Niu et al. [37] proposed to model the dynamic process of the advanced persistent threat attack. Chen et al. [38] proposed a privacy-preserving deep learning model for vehicular ad-hoc networks by encrypting the transportation data.

However, there have been few works on adversarial attacks focusing on the significant multi-task situation. Guo et al. [13] proposed a unified framework that can craft adversarial examples for multiple tasks. The framework consists of a shared encoder for all tasks and multiple task-specific decoders, which is similar to the architecture of MTL networks. Their work is committed to leveraging shared knowledge among multiple tasks to generate multi-task adversarial examples efficiently. The adversarial examples for different tasks are not the same due to their task-specific decoders. In contrast, we pay attention to generating consistent adversarial examples for different tasks since using the same input to perform multiple tasks is more common in multi-task problems.

3. Class-specific multi-task adversarial attack

The framework of our proposed method is depicted in Figure 1. We propose to generate adversarial examples to attack a multi-task network. First, we design an attention-distracting loss function to attack the attention heat maps, which contain more generalization information than feature representations. Then, during the gradient-based adversarial example-generating process, we propose to sharpen the gradients so that the gradients with multi-task generalization information, rather than only task-specific information, can make a greater impact.

3.1. Problem formulation

The objective of the multi-task adversarial attack is to craft perturbations that can fool a deep multi-task model $f(\cdot)$ for each image X_i in the dataset so that each adversarial image can conceal all task labels $y_i = \left\{y_i^1, y_i^2, \ldots, y_i^T\right\}$ from the deep multi-task model, where T represents the number of the tasks. Considering that attacking a specific class is more rational in reality, we focus on the class-specific setting. That is, for each class k contained in each image, we find a perturbation ΔX_i^k to make all of the tasks fail on the specific regions $R_i^k = \{(u, v) | c_i(u, v) = k\}$ as much as possible, which can be formulated as follows:

$$\begin{aligned} \max_{\Delta X_i^k} & \frac{1}{T \cdot \left|R_i^k\right|} \sum_{\substack{t=1,2,\ldots T \\ (u,v)\in R_i^k}} d\left(f\left(X_i + \Delta X_i^k\right)^t (u, v), y_i^t(u, v)\right) \\ & \text{s.t.} \left\|\Delta X_i^k\right\|_p \leq \varepsilon \end{aligned} \tag{3.1}$$

where c_i denotes the pixel-level class label of X_i. $d(x_1, x_2)$ denotes a distance function to measure the difference between x_1 and x_2. $\|\cdot\|_p$ is the L_p norm, and ε limits the maximum deviation of the

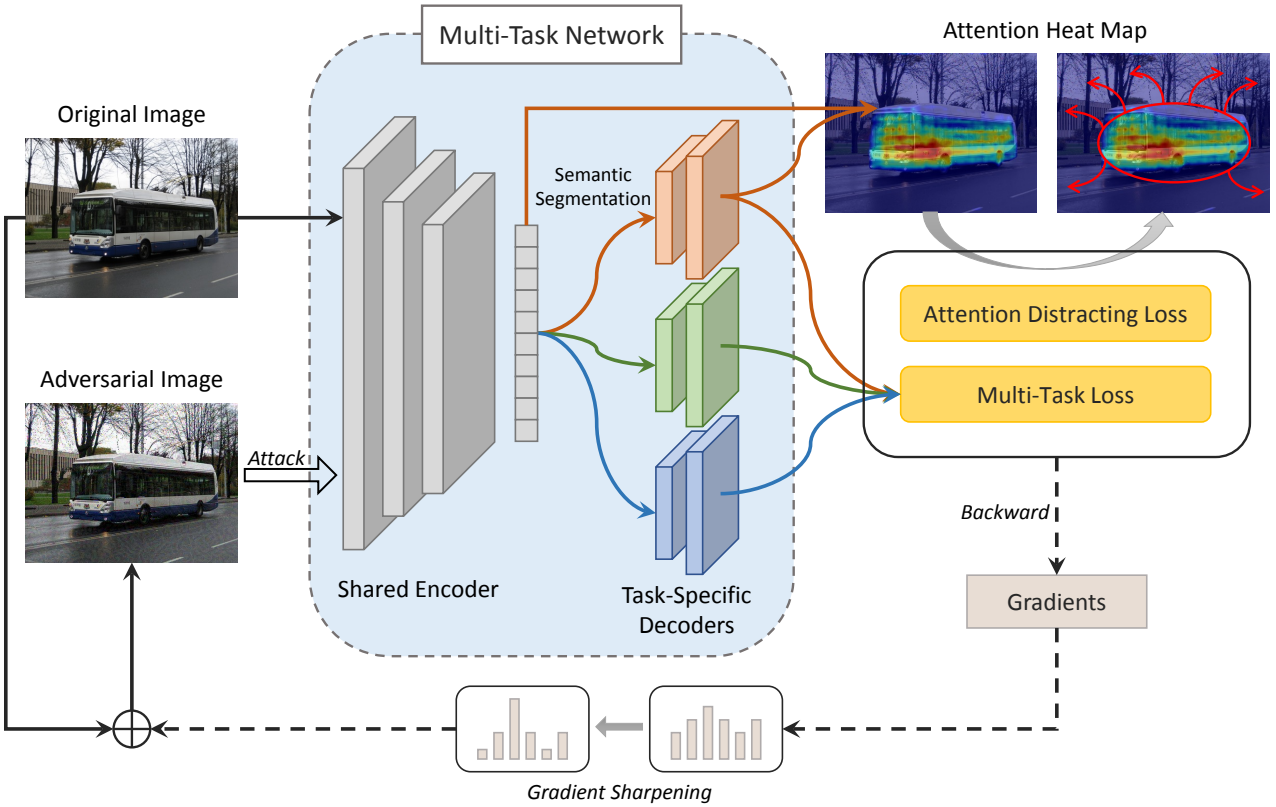

Figure 1. Overview of the proposed method. We first propose to attack the attention heat maps by using attention-distracting loss to improve the generalization ability of the adversarial examples among multiple tasks. Then, we use gradient-based schemes to generate adversarial examples and propose a gradient sharpening operation to reduce the impact of the gradients with only task-specific information. The generated adversarial examples are used to attack a multi-task network to make the multiple tasks fail as much as possible.

perturbation.

3.2. Attention-distracting loss

Semantic segmentation is a basic pixel-level task in MTL. Most MTL models contain the semantic segmentation branch because semantic segmentation has correlations with many other tasks, and it can make the feature representations keep abundant information so that combining the semantic segmentation task can boost the performance under an MTL setup. Therefore, we concentrate on taking advantage of the semantic segmentation branch to generate adversarial examples with high generalization capability among the multiple tasks. Considering that attention heat maps contain more generalization information than feature representations, we propose to attack attention heat maps in order to make the adversarial examples able to attack multiple tasks effectively.

The basic idea is to distract the focus of the attention heat maps. Let $H(X, k)$ stand for the attention heat map of the input X and a specified class k. $H(X, k)$ is a tensor with a dimension consistent with that of X. In this paper, we utilize Grad-CAM [39] to calculate the attention heat map $H(X, k)$. Grad-CAM

is a well-known method to calculate activation maps for classification tasks, where the output from the network is typically a global label and no spatial information is involved. However, in our case, the output from the network is a pixel-wise semantic segmentation map that depends on the location of a pixel. Therefore, we extend the idea of Grad-CAM to our problem. First, we focus on one pixel and use Grad-CAM to calculate the attention heat map on the basis of the output on the pixel (u, v), as follows:

$$G_k^{u,v}(i,j) = \sum_m \left(A_m(i,j) \cdot \frac{1}{|R_A|} \sum_{(i,j)\in R_A} \frac{\partial F_s^k(u,v)}{\partial A_m(i,j)} \right) \tag{3.2}$$

where A_m is the m-th feature map of the multi-task model and (i, j) denotes the spatial location on the feature map. R_A is the location set of the feature map. $F_s^k(u, v)$ denotes the semantic segmentation output of class k on pixel (u, v). We then resize the attention map $G_k^{u,v}(i, j)$ to the same size as the input image by using bilinear interpolation and take the average over the regions $(u, v) \in R^k$ to obtain the final attention heat map, as follows:

$$H(X,k) = \frac{1}{\left|R^k\right|} \sum_{(u,v)\in R^k} \text{Resize}\left(G_k^{u,v}\right) \tag{3.3}$$

This attention heat map can reflect the importance of the pixels for the predictions on class k's specific regions R^k.

Based on the calculated attention heat map, our goal is to lead the network to concentrate on irrelevant regions of class k so that the network can make incorrect predictions for multiple tasks. Therefore, we propose an attention-distracting loss function to distract the attention from the real regions of class k to other regions, which can be formulated as follows:

$$\begin{aligned} \mathcal{L}_{ad}(X,k) = & \frac{1}{\left|R^k\right|} \sum_{(i,j)\in R^k} H(X,k)(i,j) \\ & - \frac{1}{\left|R^{k^*}\right|} \sum_{(i,j)\in R^{k^*}} H(X,k)(i,j) \end{aligned} \tag{3.4}$$

where $R^{k^*} = \{(u, v) | c(u, v) \neq k\}$ represents the irrelevant regions of class k. For the predictions on class k's relevant regions R^k, optimizing this attention-distracting loss can decrease the attention on the regions themselves and, meanwhile, increase the attention on other irrelevant regions. That is, the predictions on R^k for multiple tasks can be seriously disturbed by irrelevant regions.

Moreover, the attention-distracting loss could be readily combined with the existing task-attacking loss schemes, such as cross-entropy loss for semantic segmentation, resulting in the following overall attacking loss:

$$\mathcal{L}_{att}(X,k) = \lambda_{ad}\mathcal{L}_{ad}(X,k) - \sum_{t=1}^{T} \lambda_{mt}^t \mathcal{L}_{mt}^t(X,k) \tag{3.5}$$

where λ_{ad} and λ_{mt}^t represent the trade-off parameters for the attention-distracting loss $\mathcal{L}_{ad}$ and the multi-task loss on the t-th task $\mathcal{L}_{mt}^t$. Minimizing the overall attacking loss $\mathcal{L}_{att}$ can decrease the attention-distracting loss and simultaneously increase the multi-task losses. Therefore, we can distract the attention and, meanwhile, directly reduce the multi-task performance by optimizing loss $\mathcal{L}_{att}$.

Here, in order to make the attack meet the demand for the class-specific setting, we modify the original multi-task losses by focusing on the k-th class. In detail, a class-specific region mask is used during the calculation of the losses, as follows:

$$\mathcal{L}_{mt}^{t}(X,k) = \frac{1}{\left|R^{k}\right|} \sum_{(i,j)\in R^{k}} \mathcal{L}_{mt,px}^{t}(X)(i,j) \tag{3.6}$$

where $\mathcal{L}_{mt,px}^{t}(X)$ denotes the original pixel-level loss for task t. Especially, for the human part segmentation task, we use the region mask of class $person$.

3.3. Gradient sharpening

Basically, we generate the adversarial examples on two types of schemes by calculating the gradient g. The first is a one-step scheme to compute the adversarial examples, which can be formulated as follows:

$$X_{adv}^{k} = Clip_{X,\varepsilon}\left(X - \varepsilon \frac{g(X,k)}{\|g(X,k)\|_1/N}\right) \tag{3.7}$$

where the $Clip_{X,\varepsilon}(\cdot)$ function performs per-pixel clipping to make the result limited in the L_∞ ε-neighborhood of the source image X. The gradient g is normalized by its average L_1-norm, i.e., $\|g(X,k)\|_1/N$, where N is the size of the image. The second scheme is computing the adversarial examples in a multi-step iterative process, which can be described as follows:

$$\begin{aligned} X_{adv}^{k,0} &= X \\ X_{adv}^{k,j+1} &= Clip_{X,\varepsilon}\left(X_{adv}^{k,j} - \alpha \frac{g\left(X_{adv}^{k,j},k\right)}{\left\|g\left(X_{adv}^{k,j},k\right)\right\|_1 \Big/ N}\right) \end{aligned} \tag{3.8}$$

where ε limits the maximum deviation of the overall perturbation, while α represents the step deviation in the multi-step iterative process. Based on the overall attacking loss, the gradient g can be calculated by

$$g(X,k) = \frac{\partial \mathcal{L}_{att}(X,k)}{\partial X} \tag{3.9}$$

However, due to the integration of multi-task branches, the overall loss involves much task-specific information. Meanwhile, the gradients of regions which are correlated to multiple tasks can have greater absolute values than those of regions which are only correlated to a single task. Therefore, in order to make the adversarial examples pay more attention to the regions which are correlated to multiple tasks, we propose a gradient sharpening method. Specifically, we use an exponential function to sharpen the calculated gradients so that the gradients with higher absolute values can make a greater impact. The sharpened gradients can be calculated by

$$g_s(X,k) = \operatorname{sign}(g(X,k)) \exp\left(\frac{|g(X,k)|}{\gamma}\right) \tag{3.10}$$

where $0 < \gamma < 1$ is a parameter to control the sharpening degree. A lower value of γ can make the gradients sharper, and vice versa. Then, we replace the gradient g in Eq (3.7) or (3.8) with the sharpened gradient g_s and generate the adversarial examples. In addition, the final perturbation added to the original image can be calculated by

$$\Delta X^{k} = X_{adv}^{k} - X \tag{3.11}$$

4. Experiments

We evaluated the proposed method by performing extensive experiments on two common datasets. We also designed several validation experiments to show the effectiveness of the proposed method from different aspects. The details of the experimental settings and results are reported in this section.

4.1. Experimental settings

4.1.1. Datasets

We performed our experimental evaluation on the NYUD-V2 [16] and PASCAL [17] datasets. The NYUD-V2 dataset is composed of indoor scene images recorded by both RGB and depth cameras. Each image with a label density map and depth map was used for semantic segmentation and depth estimation tasks in this study. We used the original 795 training and 654 test images for our experiments. The PASCAL VOC dataset is a widely used dataset for semantic segmentation. As an extension, PASCAL-Context [40] provides additional annotations for PASCAL VOC 2010. We used the split from PASCAL-Context, which has annotations for both semantic segmentation and human part segmentation. In addition, we obtained the surface normal labels from [41], which distilled them from a pre-trained state-of-the-art model [42]. Based on this PASCAL dataset, we performed three tasks: semantic segmentation, human part segmentation and surface normal estimation.

4.1.2. Implementation details

We generated adversarial examples to fool a deep MTL network pre-trained on the datasets with a shared encoder for all of the tasks and separate task-specific heads, which is one of the most commonly used backbones in deep MTL [9, 10]. As for the architecture, we used a DeepLab-v3+ [43] backbone network, which is based on a ResNet [3] encoder with dilated convolutions and powerful decoders with atrous spatial pyramid pooling modules to preserve reasonable spatial dimensions for dense predictions. ResNet-50 and ResNet-18 were used as encoders for the NYUD-V2 and PASCAL datasets, respectively. All networks in our experiments were implemented by using PyTorch [44] on a single GeForce RTX 3090 GPU. In order to train well-performed multi-task networks on the two datasets, we used the effective training setup from [41], which selects the optimal loss weights by grid search. As for the trade-off parameters in our attacking loss function, we use the same loss weights as the training strategy of the multi-task network for the multi-task losses and set $\lambda_{ad} = 1$ and $\lambda_{ad} = 0.7$ for the proposed attention-distracting loss on the NYUD-V2 and PASCAL datasets, respectively. In addition, the adversarial perturbation is bounded as $\varepsilon = 0.12 \times 255$, and γ is set to 0.1 for the appropriate sharpening level.

4.1.3. Evaluation metrics

We first evaluate the performance of original images and adversarial images on different tasks by using several different metrics. We use the mean intersection over union ($mIoU$) to evaluate the semantic segmentation and human part segmentation tasks. The depth estimation task is evaluated by using the root mean square error. The surface normals are evaluated by using the mean error in the predicted angles. Then, we use the drop of the generated adversarial images on the task performance to evaluate the success rate of the adversarial attacks. A larger drop value means a more powerful

adversarial attack. The class-specific multi-task adversarial attack performance of attack method a can be defined as the average per-task drop on performance with respect to the original image *ori*:

$$D_a^k = \frac{1}{T}\sum_{t=1}^{T}(-1)^{l_t}\frac{M_{ori,t}^k - M_{a,t}^k}{M_{ori,t}^k} \tag{4.1}$$

where $l_t = 1$ if a lower value is better for performance measure M_t of task t, and it is 0 otherwise. In addition, the average D_a^k among all classes can be calculated by

$$D_a = \frac{1}{K}\sum_{k=1}^{K} D_a^k \tag{4.2}$$

where K represents the number of classes.

4.1.4. Baselines

In order to show the effectiveness of the proposed method, we compare our proposed method on the two types of schemes with two popular attack strategies: FGSM [12] and PGD [30]. The FGSM is a one-step attack strategy, and PGD is a multi-step variant. We use the multi-task attack losses without the proposed attention-distracting loss to perform FGSM and PGD attacks. The gradient-sharpening operation is not used during the calculation of the gradients. In addition, the FGSM is compared with our method on the one-step scheme, while PGD is compared with our method on the multi-step iterative scheme for fair comparison.

4.2. Experimental results

We evaluated the proposed ADGS method, including the one-step scheme *ADGS_os* and the multi-step iterative scheme *ADGS_ms*, on the NYUD-V2 and PASCAL datasets. As a comparison, we also implemented the baseline methods, FGSM and PGD. Considering the efficiency, the calculation of the attention heat maps and the backpropagation of the attention-distracting loss will increase computational time cost, while the gradient-sharpening operation has little impact. But, the increase is quite reasonable and can hardly affect efficiency since it only takes 0.141 s and 0.053 s for one generation step of the proposed ADGS method on the NYUD-V2 and PASCAL datasets, respectively, while the baseline methods take 0.053 s and 0.025 s.

The results on the NYUD-V2 dataset are reported in Table 1. We can see that the proposed ADGS method can obtain a larger average per-task drop D_a^k than the baseline attack methods on all 40 classes. In fact, the adversarial examples generated by ADGS had better attack performance on both semantic segmentation and depth estimation tasks, which indicates the effectiveness of the proposed ADGS method. The reason can be that ADGS neglects much task-specific information and pays attention to the multi-task common information during the generation of the adversarial examples so that the generated adversarial examples can have higher generalization capability among the multiple tasks.

As for the PASCAL dataset, the results are shown in Tables 2 and 3. While the number of tasks we performed on the PASCAL dataset increased to three, the proposed ADGS could still realize better attack performance on all of the semantic segmentation, surface normal estimation and human part segmentation tasks. This can further demonstrate that the proposed attention-distracting loss and

Table 1. Comparison results on NYUD-V2 dataset for semantic segmentation and depth estimation tasks. "Avg." represents average results among the classes. The symbols ↓ and ↑ denote that lower and higher values mean better attack performance, respectively. The specific values of the proposed method's increase or decrease from the baseline methods are shown in parentheses.

Task	Method	wall	floor	cabinet	bed	chair	sofa	table	door	window	book-shelf	picture	counter	blinds	desk
Semantic Segmentation ↓	Clean	76.9901	81.3799	56.1077	61.2220	57.6939	58.1987	38.1248	37.0479	43.4251	43.2707	58.2955	55.4290	55.4129	19.1879
	FGSM	47.3930	45.4958	16.0723	11.0596	10.1264	22.7287	0.8679	2.3653	6.1089	12.4503	20.0025	5.5507	13.3367	0.3029
	ADGS_os	44.4819	41.5778	14.6603	11.0465	8.1305	22.2645	0.8673	2.3409	6.0079	10.4823	17.6635	5.3165	13.3357	0.2710
	PGD	33.5250	22.6349	5.9242	1.9502	1.8691	1.3900	0.0000	0.4543	0.0102	0.2488	5.6603	0.1923	1.9070	0.0000
	ADGS_ms	26.3597	17.5876	3.9453	1.3440	1.2667	1.0780	0.0000	0.3825	0.0058	0.0140	4.0807	0.0981	1.8266	0.0000
Depth Estimation ↑	Clean	0.5854	0.3197	0.4420	0.3137	0.4842	0.4418	0.5393	0.7262	0.8865	0.7056	0.6494	0.5338	0.4717	0.4129
	FGSM	1.3057	0.8148	1.1072	0.6519	0.8720	0.8259	0.8516	1.2113	1.6467	1.3053	1.4624	1.1134	1.0825	0.7108
	ADGS_os	1.3107	0.8194	1.1683	0.6521	0.8846	0.8380	0.8517	1.2581	1.6785	1.3151	1.4795	1.1260	1.0833	0.7287
	PGD	2.2207	1.4189	1.6785	0.8806	1.2030	1.0348	1.1471	1.5570	2.2553	1.6317	2.2755	1.4921	1.6873	1.1452
	ADGS_ms	2.2917	1.5711	2.0371	0.8870	1.2304	1.0753	1.1583	1.7007	2.2573	1.6538	2.3448	1.5169	1.7535	1.2406
D_a^k (%) ↑	FGSM	80.74	99.48	110.93	94.87	81.27	73.94	77.82	80.21	85.84	78.11	95.44	99.28	102.71	85.28
	ADGS_os	83.06	102.61	119.10	94.92	84.30	75.71	77.83	83.46	87.75	81.08	98.76	100.67	102.80	87.54
	PGD	167.90	208.00	184.60	138.76	122.61	115.92	106.35	106.59	127.19	115.34	170.35	139.59	177.13	138.68
	ADGS_ms	178.62	234.91	226.93	140.28	125.96	120.77	107.39	116.58	127.31	117.17	177.04	142.00	184.22	150.23

Task	Method	shelves	curtain	dresser	pillow	mirror	floor mat	clothes	ceiling	books	refrig-erator	tele-vision	paper	towel	shower curtain
Semantic Segmentation ↓	Clean	13.6158	48.5403	40.3091	38.4484	36.2413	31.1598	20.6583	66.4646	30.9703	47.7018	54.1486	30.9493	34.5802	30.2603
	FGSM	0.0001	9.2770	0.5528	4.9420	0.0000	2.8643	0.5000	16.9892	0.6651	0.9573	2.6580	2.3315	1.2761	0.0000
	ADGS_os	0.0000	9.1310	0.5222	4.6005	0.0000	1.9793	0.4822	15.0138	0.4619	0.6029	2.6105	2.3084	1.0660	0.0000
	PGD	0.0000	1.4859	0.0000	0.4213	0.0000	0.0000	0.0000	8.9575	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	ADGS_ms	0.0000	1.3438	0.0000	0.3258	0.0000	0.0000	0.0000	8.2651	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Depth Estimation ↑	Clean	0.5850	0.5018	0.3416	0.4258	0.9438	0.3605	0.4899	0.9541	0.4919	0.4945	0.7112	0.3806	0.4965	0.2747
	FGSM	1.1218	1.0815	0.7797	0.8332	1.2392	0.6517	0.8076	1.8005	0.8682	0.9965	1.4918	0.7544	1.0696	0.5495
	ADGS_os	1.1270	1.1000	0.7904	0.8558	1.3438	0.6527	0.8148	1.8191	0.9066	1.0062	1.4921	0.7546	1.0826	0.5933
	PGD	1.6581	1.8227	0.9551	1.1466	1.7877	0.8236	1.1145	2.7418	1.5463	1.6348	2.2144	1.3101	1.8185	0.7554
	ADGS_ms	1.7015	1.8552	1.1677	1.1961	2.2983	0.8486	1.1393	2.9048	1.7052	1.6634	2.2531	1.3386	1.8766	1.1058
D_a^k (%) ↑	FGSM	95.88	98.21	113.44	91.41	65.65	85.79	81.21	81.58	87.18	99.75	102.42	95.34	105.87	100.02
	ADGS_os	96.32	100.20	115.04	94.51	71.19	87.35	81.99	84.04	91.41	101.11	102.49	95.40	107.48	107.99
	PGD	141.72	180.09	139.80	134.09	94.71	114.23	113.75	136.95	157.18	165.30	155.68	172.11	183.13	137.50
	ADGS_ms	145.43	183.47	170.92	140.03	121.76	117.70	116.28	146.01	173.33	168.19	158.40	175.85	188.98	201.27

Task	Method	box	white-board	person	night stand	toilet	sink	lamp	bathtub	bag	other structure	other furniture	other prop	Avg.
Semantic Segmentation ↓	Clean	13.1307	51.9528	73.3729	33.4379	71.6365	52.0400	43.7487	38.2039	10.5524	26.1339	14.5337	35.5264	43.2526
	FGSM	0.0000	0.0000	9.9659	0.0008	10.4161	4.3659	8.8168	0.0000	0.0000	3.1974	0.2881	10.2078	7.6033
	ADGS_os	0.0000	0.0000	9.5471	0.0007	10.0458	1.7914	8.1467	0.0000	0.0000	2.7937	0.1635	9.2670	6.9745 (*-0.6288*)
	PGD	0.0000	0.0000	0.4877	0.0000	0.0000	0.3154	0.3073	0.0000	0.0000	0.3802	0.0371	1.7777	2.2484
	ADGS_ms	0.0000	0.0000	0.4803	0.0000	0.0000	0.0023	0.0580	0.0000	0.0000	0.0240	0.0000	1.4320	1.7480 (*-0.5004*)
Depth Estimation ↑	Clean	0.6017	0.5873	0.4252	0.2881	0.2229	0.2832	0.5839	0.4470	0.5608	0.8809	0.5875	0.6082	0.5260
	FGSM	0.8273	0.9164	1.1222	0.5749	0.5938	0.8830	1.2003	0.4457	0.8587	1.3775	0.8919	1.0720	0.9943
	ADGS_os	0.9314	1.0417	1.1475	0.5765	0.6125	0.9291	1.2181	0.4732	0.8661	1.4067	0.9662	1.1085	1.0203 (*+0.0260*)
	PGD	1.1754	1.4978	1.8442	0.7790	0.9077	1.4209	1.9631	0.6213	1.3496	1.7602	1.1948	1.6370	1.4777
	ADGS_ms	1.6311	1.9918	1.8450	0.8002	1.1363	1.5940	2.0962	0.8368	1.3861	1.7838	1.6074	1.7023	1.6046 (*+0.1269*)
D_a^k (%) ↑	FGSM	68.75	78.02	125.17	99.77	125.93	151.70	92.71	49.85	76.56	72.07	74.92	73.76	90.97
	ADGS_os	77.40	88.69	128.43	100.05	130.38	162.31	95.00	52.93	77.22	74.50	81.67	78.09	94.07 (*+3.10*)
	PGD	97.67	127.52	216.53	135.20	203.61	250.56	167.75	69.50	120.33	99.18	101.56	132.08	144.17
	ADGS_ms	135.54	169.57	216.63	138.88	254.89	281.42	179.43	93.60	123.58	101.20	136.80	137.93	158.16 (*+13.99*)

Table 2. Comparison results on PASCAL dataset for semantic segmentation and surface normal estimation tasks. "Avg." represents average results among the classes. The symbols ↓ and ↑ denote that lower and higher values mean better attack performance, respectively. The specific values of the proposed method's increase or decrease from the baseline methods are shown in parentheses.

Task	Method	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table
Semantic Segmentation ↓	Clean	69.7485	55.3695	65.2569	45.2368	54.5278	80.0767	69.7038	77.9084	23.2084	62.9036	35.6100
	FGSM	6.3797	13.0935	4.8273	2.1559	15.9784	23.6383	27.0001	28.2298	1.5824	0.0819	2.3187
	ADGS_os	5.4093	12.9291	2.0414	1.7624	10.6560	8.1874	23.0062	27.3757	1.3500	0.0814	2.2334
	PGD	0.0355	1.3790	1.0742	1.3339	5.0285	7.0247	9.2926	5.3788	0.0000	0.0000	0.0053
	ADGS_ms	0.0000	1.1483	0.0076	1.3154	2.0298	1.8503	7.9092	4.9685	0.0000	0.0000	0.0016
Surface Normal Estimation ↑	Clean	15.6729	17.5815	13.7144	17.5100	15.7998	12.5370	16.2255	14.1990	16.0765	16.0318	14.9989
	FGSM	24.5548	30.4238	25.6373	27.7976	28.5948	24.7454	31.6472	30.6735	26.8303	49.0884	24.8347
	ADGS_os	25.8617	30.4932	26.5835	29.1060	29.4265	30.4890	32.3229	31.0406	27.0509	49.1167	25.6008
	PGD	27.1021	30.8113	30.0788	27.3372	32.1825	25.4998	36.1721	34.2953	28.8159	51.8043	28.4371
	ADGS_ms	29.1838	31.2109	34.4765	30.6392	36.4599	43.7306	41.3351	37.7924	29.5512	61.6953	28.4942
D_a^k (%) ↑	FGSM	73.76	74.70	89.77	76.99	75.84	83.93	78.16	89.90	80.04	153.03	79.53
	ADGS_os	78.63	75.04	95.35	81.16	83.35	116.48	83.10	91.74	81.22	153.12	82.21
	PGD	86.44	86.38	108.84	76.59	97.23	97.31	104.80	117.31	89.62	161.57	94.79
	ADGS_ms	93.10	87.72	125.69	86.04	113.52	173.25	121.70	129.89	91.91	192.42	94.99

Task	Method	dog	horse	motorbike	person	potted plant	sheep	sofa	train	TV/ monitor	Avg.
Semantic Segmentation ↓	Clean	71.3308	68.2018	65.5548	77.5841	41.8748	71.2738	35.7286	68.4359	55.2319	59.7383
	FGSM	24.0875	2.8101	7.2248	43.8461	6.7924	1.9667	9.7871	5.4348	1.5692	11.4402
	ADGS_os	22.6547	2.8099	5.1636	39.7630	3.9179	0.7290	9.2771	1.8319	1.2599	9.1220 (*-2.3182*)
	PGD	6.4676	0.0653	0.1937	32.5648	2.0136	0.0000	3.7443	0.8408	0.0759	3.8259
	ADGS_ms	6.3779	0.0575	0.1372	26.6130	1.8882	0.0000	3.5932	0.6464	0.0243	2.9284 (*-0.8975*)
Surface Normal Estimation ↑	Clean	14.6964	16.9342	15.8940	14.8435	16.6265	15.7815	14.0807	16.3957	13.0599	15.4330
	FGSM	30.1769	41.7528	31.6862	25.1007	32.9519	31.5115	26.9563	28.1213	24.5354	29.8810
	ADGS_os	30.8173	41.7639	32.4610	25.9008	35.1169	31.8405	27.1240	32.5282	26.0290	31.0337 (*+1.1527*)
	PGD	32.5360	43.8609	34.8493	27.3991	34.2904	34.7990	29.6597	28.3556	26.9978	32.2642
	ADGS_ms	36.4277	43.8949	43.4086	28.0833	46.9028	38.4817	29.6697	40.7102	30.5723	37.1360 (*+4.8718*)
D_a^k (%) ↑	FGSM	85.78	121.22	94.17	56.87	90.98	98.46	82.02	81.79	92.51	87.97
	ADGS_os	88.97	121.25	98.18	61.14	100.93	100.37	83.33	97.86	98.51	93.60 (*+5.63*)
	PGD	106.16	129.46	109.48	72.62	100.72	110.25	100.08	85.86	103.29	101.94
	ADGS_ms	119.46	129.56	136.45	77.63	138.79	121.92	100.33	123.68	117.02	118.75 (*+16.81*)

Table 3. Comparison results on PASCAL dataset for human part segmentation task. The symbol ↓ denotes that lower values mean better attack performance. The specific values of the proposed method's decrease from the baseline methods are shown in parentheses.

Task	Method	head	torso	upper arm	lower arm	upper leg	lower leg	mIoU
Human Part Segmentation ↓	Clean	86.1157	68.0468	49.7283	49.9712	44.5645	40.6091	56.5059
	FGSM	56.9747	35.2089	17.4860	16.0012	10.1198	6.4798	23.7117
	ADGS_os	54.9319	33.7157	16.9755	15.5809	8.7527	5.0452	22.5003 (*-1.2114*)
	PGD	39.0677	21.1128	8.4473	8.5785	4.0701	2.6900	13.9944
	ADGS_ms	34.6721	18.3947	7.9043	8.1781	3.5166	1.9604	12.4377 (*-1.5567*)

gradient-sharpening operation can leverage the multi-task generalization information to generate more powerful adversarial examples for multiple tasks.

In addition, we present some qualitative examples in Figure 2. From the figure, we can see that the proposed ADGS method exactly distracts the attention from the class-specific regions and can do more damage to the multi-task network on multiple tasks compared to the baseline attack methods.

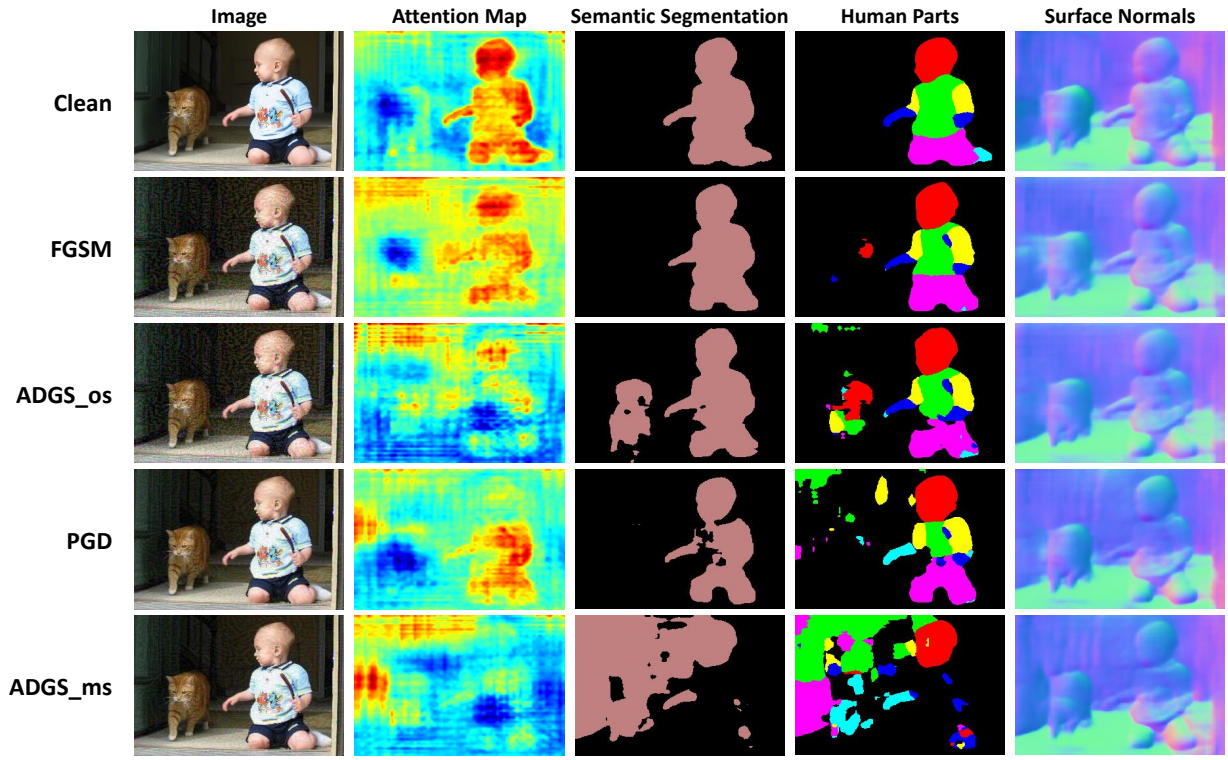

Figure 2. Qualitative results on class *person* of PASCAL dataset. The first column shows the original image for the clean baseline (first row) and adversarial images for the attack methods. Differences can be seen between different methods in attention map and the three tasks on PASCAL dataset.

4.3. Validation experiments

We also conducted several validation experiments to demonstrate the effectiveness of the proposed method from various perspectives.

4.3.1. Attack on single-task networks

In order to further validate the generalization capacity of the proposed ADGS method for multiple tasks, we aimed to attack single-task networks using the adversarial examples generated from the multi-task network. Each single-task network comprises an encoder and a task-specific decoder with the same architecture as the corresponding branch in the multi-task network. We trained the multiple single-task networks independently and then used the adversarial examples generated from the multi-task network to attack them. Note that the black-box setting has been applied in this attack scenario, as knowledge of the single-task networks is not accessible during the generation of the adversarial examples. The attack results on the NYUD-V2 dataset (two tasks) and the PASCAL dataset (three tasks) are shown in Tables 4 and 5, respectively. Here, the average per-task drop D_a is calculated based on the performance of multiple single-task networks rather than a multi-task network. Our proposed method can obtain competitive attack results and outperform the baseline methods on this black-box multi-task to single-task transferring attack scenario, which demonstrates that the proposed ADGS

method can improve the generalization ability of the adversarial examples to perform more powerful attacks on unseen single-task networks.

Table 4. Average attack results for single-task networks on the NYUD-V2 dataset for all of the classes. Here, D_a is calculated based on the performance of multiple single-task networks rather than a multi-task network. The symbols ↓ and ↑ denote that lower and higher values mean better attack performance, respectively. The specific values of the proposed method's increase or decrease from the baseline methods are shown in parentheses.

Method	Semantic Segmentation ↓	Depth Estimation ↑	D_a (%) ↑
Clean	43.6960	0.5339	-
FGSM	10.2176	0.9294	79.93
ADGS_os	9.7074 (*-0.5102*)	0.9537 (*+0.0243*)	82.85 (*+2.92*)
PGD	4.1700	1.2500	117.75
ADGS_ms	3.5925 (*-0.5775*)	1.3211 (*+0.0711*)	125.57 (*+7.82*)

Table 5. Attack results for single-task networks on class *person* of PASCAL dataset. Here, D_a is calculated based on the performance of multiple single-task networks rather than a multi-task network. The symbols ↓ and ↑ denote that lower and higher values mean better attack performance, respectively. The specific values of the proposed method's increase or decrease from the baseline methods are shown in parentheses.

Method	Semantic Segmentation ↓	Surface Normal Estimation ↑	Human Part Segmentation ↓	D_a (%) ↑
Clean	78.6754	13.7414	56.8839	-
FGSM	48.1401	22.6802	24.8303	53.40
ADGS_os	45.7296 (*-2.4105*)	23.0849 (*+0.4047*)	23.7961 (*-1.0342*)	56.01 (*+2.61*)
PGD	38.1858	21.5924	16.9978	59.57
ADGS_ms	34.0767 (*-4.1091*)	21.9311 (*+0.3387*)	15.9169 (*-1.0809*)	62.77 (*+3.20*)

4.3.2. Attack on a multi-task network without semantic segmentation branch

The proposed attention-distracting loss function is based on the commonly used semantic segmentation branch in multi-task networks. In order to investigate the attack performance of the proposed method on the multi-task networks without a semantic segmentation branch, we have designed an attack strategy. We have incorporated an auxiliary semantic segmentation decoder into a trained multi-task network without the semantic segmentation branch and fine-tuned the decoder to perform the semantic segmentation task. The modified multi-task network is then used to generate adversarial examples by using the proposed ADGS method, which are employed to attack the original multi-task network. The experiments were conducted on the class *person* of the PASCAL dataset, with surface normal estimation and human part segmentation tasks. As a comparison, we performed the baseline FGSM and PGD methods to directly attack the original multi-task network without the auxiliary semantic segmentation branch. The results are reported in Table 6. As can be observed, based on the weak auxiliary semantic segmentation branch, the proposed ADGS method can still realize competitive attack performance on the two original tasks, which indicates that ADGS can

improve the generalization ability of the adversarial examples on multiple tasks regardless of the semantic segmentation performance. Thus, the proposed method can be generalized to attack most multi-task networks by adding an auxiliary semantic segmentation branch when the semantic segmentation task is not performed.

Table 6. Attack results for a multi-task network without a semantic segmentation branch on class *person* of PASCAL dataset. The symbols ↓ and ↑ denote that lower and higher values mean better attack performance, respectively. The specific values of the proposed method's increase or decrease from the baseline methods are shown in parentheses.

Method	Surface Normal Estimation ↑	Human Part Segmentation ↓							D_a^k (%) ↑
		head	torso	upper arm	lower arm	upper leg	lower leg	mIoU	
Clean	14.7008	85.8677	67.6339	50.1127	49.7789	43.6892	39.6205	56.1172	-
FGSM	25.2011	58.7565	35.9006	20.1132	17.6379	9.9755	7.6565	25.0067	63.43
ADGS_os	25.9299 (*+0.7288*)	56.2437	34.5907	19.9024	17.3155	9.0294	7.0439	24.0209 (*-0.9858*)	66.79 (*+3.36*)
PGD	27.8284	44.4897	23.9749	11.4169	10.5115	5.6239	3.7747	16.6319	79.83
ADGS_ms	28.8143 (*+0.9859*)	40.3611	21.9501	11.1502	10.4040	4.9918	3.4026	15.3766 (*-1.2553*)	84.30 (*+4.47*)

5. Conclusions

In this paper, we focus on the multi-task setting of adversarial attacks, where the objective is to generate a consistent adversarial example for each image to attack multiple tasks in an MTL network. This scenario holds practical significance due to the widespread use of multi-task real-world AI systems. To enhance the generalization capability of the consistent adversarial examples across multiple tasks, we have proposed a multi-task attack method with ADGS. First, we leverage the generalization knowledge contained in attention heat maps to distract attention from the attack regions. Second, to reduce the impact of task-specific information during the generation of adversarial examples, we propose to sharpen the gradients for the gradient-based adversarial example-generating schemes. In this way, the gradients with multi-task information will have a greater impact than those with only task-specific information. Our experimental results on two test benchmarks demonstrate the efficacy and superiority of the proposed method. In addition, improving the robustness of multi-task networks against generalized multi-task adversarial attacks can be a future trend.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62276030 and 62236003.

Conflict of interest

We declare that there is no conflict of interest.

References

1. A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Commun. ACM*, **6** (2017), 84-–90. https://doi.org/10.1145/3065386
2. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv: 1409.1556.
3. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
4. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 7132–7141. https://doi.org/10.1109/CVPR.2018.00745
5. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965
6. D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, preprint, arXiv: 1406.2283.
7. X. Wang, D. Fouhey, A. Gupta, Designing deep networks for surface normal estimation, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 539–547. https://doi.org/10.1109/CVPR.2015.7298652
8. R. Caruana, Multitask learning, *Mach. Learn.*, **28** (1997), 41–75. https://doi.org/10.1023/A:1007379606734
9. S. Ruder, An overview of multi-task learning in deep neural networks, preprint, arXiv:1706.05098.
10. S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, L. Van Gool, Multi-task learning for dense prediction tasks: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2022), 3614–3633. https://doi.org/10.1109/TPAMI.2021.3054719
11. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, et al., Intriguing properties of neural networks, *Int. Conf. Learn. Represent.*, 2014.
12. I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, *Int. Conf. Learn. Represent.*, 2015.
13. P. Guo, Y. Xu, B. Lin, Y. Zhang, Multi-task adversarial attack, preprint, arXiv:2011.09824.
14. Y. Li, C. Chen, M. Duan, Z. Zeng, K. Li, Attention-aware encoder–decoder neural networks for heterogeneous graphs of things, *IEEE Trans. Ind. Inform.*, **17** (2020), 2890–2898. https://doi.org/10.1109/TII.2020.3025592
15. B. Pu, Y. Liu, N. Zhu, K. Li, K. Li, Ed-acnn: Novel attention convolutional neural network based on encoder–decoder framework for human traffic prediction, *Appl. Soft. Comput.*, **97** (2020), 106688. https://doi.org/10.1016/j.asoc.2020.106688
16. N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in *Computer Vision – ECCV 2012*, (2012), 746–760. https://doi.org/10.1007/978-3-642-33715-4_54

17. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, *Int. J. Comput. Vis.*, **88** (2010), 303–338. https://doi.org/10.1007/s11263-009-0275-4

18. T. Evgeniou, M. Pontil, Regularized multi-task learning, in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (2004), 109–117. https://doi.org/10.1145/1014052.1014067

19. B. Bakker, T. Heskes, Task clustering and gating for bayesian multitask learning, *J. Mach. Learn. Res.*, **4** (2003), 83–99. https://doi.org/10.1162/153244304322765658

20. A. Argyriou, T. Evgeniou, M. Pontil, Convex multi-task feature learning, *Mach. Learn.*, **73** (2008), 243–272. https://doi.org/10.1007/s10994-007-5040-8

21. A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 7482–7491. https://doi.org/10.1109/CVPR.2018.00781

22. Z. Chen, V. Badrinarayanan, C. Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, preprint, arXiv:1711.02257.

23. O. Sener, V. Koltun, Multi-task learning as multi-objective optimization, *Adv. Neural Inf. Process. Syst.*, (2018), 525–536.

24. I. Misra, A. Shrivastava, A. Gupta, M. Hebert, Cross-stitch networks for multi-task learning, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 3994–4003. https://doi.org/10.1109/CVPR.2016.433

25. Y. Gao, J. Ma, M. Zhao, W. Liu, A. L. Yuille, Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 3205–3214. https://doi.org/10.1109/CVPR.2019.00332

26. S. Liu, E. Johns, A. J. Davison, End-to-end multi-task learning with attention, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 1871–1880. https://doi.org/10.1109/CVPR.2019.00197

27. M. Wallingford, H. Li, A. Achille, A. Ravichandran, C. Fowlkes, R. Bhotika, et al., Task adaptive parameter sharing for multi-task learning, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 7561–7570. https://doi.org/10.1109/CVPR52688.2022.00741

28. I. Kokkinos, Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6129–6138. https://doi.org/10.1109/CVPR.2017.579

29. F. Ott, D. Rügamer, L. Heublein, B. Bischl, C. Mutschler, Joint classification and trajectory regression of online handwriting using a multi-task learning approach, in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, (2022), 266–276. https://doi.org/10.1109/WACV51458.2022.00131

30. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, preprint, arXiv: 1706.06083.

31. S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 2574–2582, https://doi.org/10.1109/CVPR.2016.282

32. A. S. Rakin, Z. He, J. Li, F. Yao, C. Chakrabarti, D. Fan, T-bfa: Targeted bit-flip adversarial weight attack, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2022), 7928–7939. https://doi.org/10.1109/TPAMI.2021.3112932

33. C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, A. Yuille, Adversarial examples for semantic segmentation and object detection, in *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 1369–1378. https://doi.org/10.1109/ICCV.2017.153

34. G. Zhao, M. Zhang, J. Liu, Y. Li, J. R. Wen, Ap-gan: Adversarial patch attack on content-based image retrieval systems, *Geoinformatica*, **26** (2022), 347–377. https://doi.org/10.1007/s10707-020-00418-7

35. V. Mirjalili, A. Ross, Soft biometric privacy: Retaining biometric utility of face images while perturbing gender, in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, (2017), 564–573. https://doi.org/10.1109/BTAS.2017.8272743

36. K. Yamanaka, R. Matsumoto, K. Takahashi, T. Fujii, Adversarial patch attacks on monocular depth estimation networks, *IEEE Access*, **8** (2020), 179094–179104. https://doi.org/10.1109/ACCESS.2020.3027372

37. W. Niu, X. Zhan, K. Li, G. Yang, R. Chen, Modeling attack process of advanced persistent threat, in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, **10066** (2016), 383–391. https://doi.org/10.1007/978-3-319-49148-6_32

38. J. Chen, K. Li, S. Y. Philip, Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain, *IEEE Trans. Intell. Transp. Syst.*, **23** (2021), 11633–11642. https://doi.org/10.1109/TITS.2021.3105682

39. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 618–626. https://doi.org/10.1109/ICCV.2017.74

40. R. Mottaghi, X. Chen, X. Liu, N. G. Cho, S. W. Lee, S. Fidler, et al., The role of context for object detection and semantic segmentation in the wild, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 891–898. https://doi.org/10.1109/CVPR.2014.119

41. K. K. Maninis, I. Radosavovic, I. Kokkinos, Attentive single-tasking of multiple tasks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2019), 1851–1860. https://doi.org/10.1109/CVPR.2019.00195

42. A. Bansal, X. Chen, B. Russell, A. Gupta, D. Ramanan, Pixelnet: Representation of the pixels, by the pixels, and for the pixels, preprint, arXiv:1702.06506.

43. L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in *Computer Vision – ECCV 2018*, (2018), 801–818. https://doi.org/10.1007/978-3-030-01234-2_49

44. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, (2019), 8026–8037. https://doi.org/10.5555/3454287.3455008

AIMS Press