Stealthy Multi-task Adversarial Attacks

Jiacheng Guo[1,2*], Tianyun Zhang[1*†], Lei Li[1], Haochen Yang[1], Hongkai Yu[1], and Minghai Qin[1,3†]

[1] Cleveland State University, Cleveland, OH 44115, USA
{j.guo58,l.li15,h.yang15}@vikes.csuohio.edu,
{t.zhang85,h.yu19}@csuohio.edu
[2] University of Wisconsin-Madison, Madison, WI 53706, USA
[3] Western Digital Research, San Jose, CA 95119, USA
{minghai.qin}@wdc.com
*Equal Contribution †Corresponding Authors

Abstract. Deep neural networks are highly vulnerable to adversarial perturbations, raising serious safety concerns in the real-world systems. While prior work mainly explores single-task attacks or jointly degrading all tasks in multi-task models, practical scenarios often demand more selective and stealthy attack strategies. To address this challenge, we propose **S**tealthy **M**ulti-**T**ask **A**dversarial **A**ttack (**SMTA**^{2}), a novel framework that selectively degrades a targeted task while strictly preserving the performance of non-targeted tasks. We formulate this objective as a constrained multi-objective optimization problem and design task-aware adversarial perturbations that maximize degradation on the targeted task without causing collateral damage on non-targeted tasks. To enhance practicality, we further introduce an automated loss-weight tuning strategy that dynamically balances attack and preservation objectives. Experiments on two multi-task benchmarks NYUv2 and Cityscapes demonstrate that SMTA^2 achieves strong attack performance on targeted tasks while maintaining non-targeted tasks intact on both undefended and adversarially trained models, establishing the first systematic framework for stealthy and selective multi-task attack framework.

Keywords: multi-task learning, adversarial attacks, stealthy attacks, deep learning

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable success across a wide range of tasks [1, 10, 19, 23], yet remain highly vulnerable to adversarial attacks [20, 29]. These attacks introduce subtle, often imperceptible perturbations that can drastically alter model predictions [2, 13, 32]. Despite their minimal visual impact, such perturbations can cause severe performance degradation, posing significant safety risks in critical applications, such as autonomous driving [5, 38]. Figure 1 illustrates typical imperceptible adversarial perturbations under different magnitudes, where even small perturbations (e.g., ϵ=2/255 or 4/255) can effectively mislead DNNs.

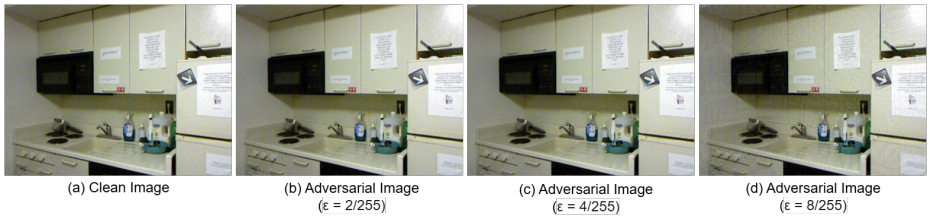

Fig. 1: Demonstration of a clean image (a) and adversarial images (b) and (c) from NYUv2 dataset.

Beyond single-task models, Multi-Task Learning (MTL) enables a shared network to jointly learn multiple related tasks, significantly improving model efficiency and generalization [4, 8, 33]. However, the shared representations also introduce coupled vulnerabilities: a perturbation crafted for one task can *unintentionally propagate to others* [12, 41]. Specifically, in safety-critical systems, attackers rarely aim for indiscriminate performance collapse. Instead, they may selectively target high-value tasks while preserving auxiliary functionalities to evade detection [35, 43]. For instance, manipulating traffic sign recognition while preserving lane detection may create subtle yet dangerous misbehavior in autonomous driving [28]. Similarly, degrading pedestrian segmentation while keeping depth estimation intact can cause catastrophic consequences without triggering obvious system-wide failure [27]. These scenarios highlight the need for *selective* and *stealthy* multi-task adversarial attacks that compromise targeted tasks while preserving the performance of others.

However, achieving *selective degradation* is fundamentally challenging [41, 43]. Multi-task objectives are inherently coupled through shared parameters, and increasing the loss of one task typically propagates gradients that harm others. Enforcing strict preservation constraints while maximizing degradation on a targeted task leads to a highly constrained and non-trivial optimization problem. Naively reweighting losses often fails to maintain non-targeted task performance, especially under strong adversarial perturbations.

To address these challenges, we propose a novel concept termed **Stealthy Multi-Task Adversarial Attack (SMTA^2)**. Given an MTL framework with multiple tasks, SMTA^2 aims to significantly degrade a **targeted task** while strictly preserving or even improving the performance of other tasks which are not expected to be attacked, regarded as **non-targeted tasks**. By crafting task-aware adversarial perturbations through a weighted loss formulation, SMTA^2 enables fine-grained control over task-specific attack behaviors. Furthermore, we develop an automated weight searching approach to dynamically search for optimal task weights, enabling efficient and adaptive attack generation without manual tuning. This strategy significantly reduces computational overhead and improves generalization across diverse inputs. Compared to random system noise which typically causes uniform and mild degradation across all tasks, SMTA^2

introduces asymmetric and task-selective degradation, a property that is empirically validated in Section 4.

To the best of our knowledge, this is the first work that systematically formulates and solves the problem of stealthy and task-selective adversarial attacks in MTL, together with a principled evaluation criterion for preservation-aware attack assessment. The main contributions are summarized as follows:

- We propose "Stealthy Multi-Task Adversarial Attacks" (**SMTA**^{2}), a novel attack paradigm which enables selective degradation of a targeted task while strictly preserving the original performance of non-targeted tasks in MTL.
- The "Stealthy Multi-Task Adversarial Attacks" is formulated as a constrained multi-objective optimization problem and designed as a task-aware objective with explicit preservation constraints.
- We develop an automated loss-weight optimization strategy that dynamically searches for optimal task weights, improving attack efficiency and stability over manual tuning.
- Experiments on NYUv2 and Cityscapes datasets in different L_p norm attacks demonstrate strong targeted attack effectiveness while preserving non-targeted tasks on both undefended and adversarially trained models.

2 Related Work

2.1 Multi-task Learning

Multi-task learning (MTL) is a machine learning paradigm which enables the simultaneous training of multiple related tasks, providing advantages such as improved data efficiency, reduced overfitting, and enhanced generalization capabilities [4,8]. By leveraging shared representations among tasks, MTL has become increasingly popular across diverse fields, including natural language processing and computer vision [21, 39, 42]. Recent advancements have categorized MTL techniques into several domains, such as regularization strategies, task relationship modeling, feature propagation, optimization frameworks, and pre-training methods [14, 16, 19].

MTL approaches are broadly classified into joint training and multi-step training methods, depending on the nature of task relationships and the level of interaction between tasks [24, 42], which has evolved significantly with the emergence of deep learning and pretrained foundation models. These advancements have introduced innovations such as task-promotable, task-agnostic training, and capabilities for zero-shot and few-shot learning, making MTL applicable in scenarios with limited task-specific labeled data [18,39]. As MTL continues to mature, emerging research focuses on improving task balancing, addressing task interference, and enhancing robustness for its up-to-date application in complex domains [24, 25].

2.2 Adversarial Attacks

Adversarial attacks pose a fundamental challenge to DNNs, where carefully crafted perturbations can induce incorrect predictions despite being largely imperceptible to humans [3, 29, 32, 35, 37]. Such perturbations expose critical vulnerabilities in modern architectures.

Gradient-based optimization remains a dominant paradigm for attack generation. Universal adversarial perturbations [31] demonstrated that input-agnostic, gradient-derived perturbations can reliably fool classifiers, laying the foundation for subsequent gradient-driven methods and extensions beyond classification. These approaches enabled adversarial attacks to be studied in structured and real-world settings, including aerial object detection [11, 36], semantically stealthy segmentation [6], and audio tasks [21] where perturbations preserve visual plausibility while inducing substantial degradation. In MTL, Guo et al. [17] introduced a Multi-Task Adversarial Attack (MTA) framework that exploits shared representations to craft joint attacks, outperforming single-task baselines. Zhang et al. [41] later standardized MTA methodologies and broadened evaluations. Nevertheless, unlike those non-stealthy architectures [17, 41], our objective is stricter: selectively attacking a targeted task and strictly preserving non-targeted tasks.

2.3 Adversarial Robustness in MTL

To improve adversarial robustness, Madry et al. [29] formulated defense as a min-max optimization problem, while Mao et al. [30] introduced natural supervision to recover correct predictions under attack.

Adversarial vulnerabilities in MTL have also been investigated. Zhang et al. [40] analyzed the robustness–accuracy trade-off theoretically. Zhang et al. [41] proposed Gradient Balancing Multi-Task Attack to jointly optimize attacks across tasks, improving task balancing through shared representations. Guo et al. [15] incorporated min–max optimization to enhance robustness in multi-task adversarial learning. These methods mainly focus on jointly attacking all tasks through improved optimization. Moreover, Zhe et al. [43] predefined "hidden tasks" which can be stealthily compromised via shared representations without explicitly modeling task relationships, but they used different task definitions and settings. Comparatively, we propose a unique SMTA^2 framework which attacks multi-task models without changing architectures or task relationships.

Overall, prior works either design general optimization-based attacks or indiscriminately degrade all tasks, leaving selective and stealthy mechanisms underexplored. Crucially, the structural dependency induced by shared representations has not been systematically exploited for controlled adversarial manipulation. We address this gap by proposing a principled framework for selective and stealthy adversarial attacks in multi-task scenarios that explicitly models inter-task relationships, enabling fine-grained degradation while preserving all non-targeted tasks.

3 Methodology

3.1 Problem Formulation

Consider a multi-task learning setting where each input sample x is associated with multiple task-specific ground-truth labels $y = (y_1, y_2, ..., y_m)$. Let $L_i(x, y_i)$ denote the loss corresponding to the i-th task in a pretrained multi-task model with shared representations.

Definition. The **Stealthy Multi-Task Adversarial Attack (SMTA^2)** is a kind of adversarial attack which aims to degrade the performance of a designated *targeted task* while strictly preserving the performance of all remaining *non-targeted tasks*.

Let i_t be the targeted task. The SMTA^2 can be formulated as

$$\begin{aligned} \underset{\delta}{\text{maximize}} \quad & L_{i_t}(x+\delta, y_{i_t}), \\ \text{subject to} \quad & L_i(x+\delta, y_i) \leq L_i(x, y_i) \text{ for } i \neq i_t, \\ & ||\delta||_p \leq \epsilon, x+\delta \in \mathcal{B}, \end{aligned} \tag{1}$$

where δ is the adversarial noise, ϵ is a value to constrain the strength of the adversarial noise, and $\mathcal{B}$ is the box constraint to ensure that a valid image is derived after adding the adversarial noise.

The attack without the first constraint, $L_i(x+\delta, y_i) \leq L_i(x, y_i)$ for $i \neq i_t$, is to be called a **non-stealthy** multi-task attack.

Optimization Challenge. Problem (1) is substantially more challenging than standard adversarial attacks. In multi-task DNNs, tasks share a common backbone representation, leading to strong inter-task coupling. Perturbations that increase the loss of one task naturally propagate to others. Therefore, the feasible region satisfying all preservation constraints is often narrow under strong perturbations, making direct optimization intractable.

To obtain a tractable surrogate, we relax problem (1) to the following weighted objective:

$$\underset{||\delta||_p \leq \epsilon, x+\delta \in \mathcal{B}}{\text{maximize}} \quad L(x+\delta, y) = \sum_{i=1}^{m} w_i L_i(x+\delta, y_i), \tag{2}$$

where w_i is the weighting factor corresponding to the i-th task. If $i = i_t$, w_i is positive; otherwise w_i is negative. The constraint in Eq. (1) serves as an effective optimization proxy so that an appropriate combination of weighting factors corresponding to different tasks could be searched each time for metric preservation when solving problem (2). It is guaranteed that an optimal solution feasible to problem (1) could be found out by adjusting the weighting factors. Specifically, when the loss of a non-targeted task increases, its corresponding negative weight effectively penalizes further degradation. Crucially, success hinges on carefully adapting $\{w_i\}$ to balance attack strength and strict preservation, which is an inherently non-trivial task due to inter-task gradient interference.

Unlike conventional adversarial attacks which optimize only a single objective, SMTA^2 must simultaneously maximize one task loss while enforcing multiple preservation constraints. Due to the shared representations and gradient interference across tasks, satisfying these constraints while maintaining strong attack effectiveness is fundamentally challenging. In this case, it is crucial to find an optimal combination to balance different tasks. Therefore, effective weight adaptation strategies are essential for navigating this coupled optimization landscape. The details of how to satisfy the constraints in the stealthy attack is discussed in the subsection 3.2.

3.2 The Stealthy Multi-Task Adversarial Attack Framework

In a general multi-task adversarial attack framework, attacking one task could negatively affect all other tasks facing severe degradation as well. As a result, the performance of non-targeted tasks could also be significantly affected. To maintain the performance of non-targeted tasks, a positive weighting factor is set for the targeted task and negative weighting factors for the non-targeted tasks when solving problem (2). The positive weighting factor can degrade the performance of the targeted task, while the negative ones could compensate for the performance degradation of non-targeted tasks caused by attacking the targeted task. If the magnitudes of negative weighting factors are too small, the performance of non-targeted tasks will still be degraded. In this case, the first constraint in problem (1) is not satisfied. If the negative weights are too large, the performance of non-targeted tasks may even surpass the clean baseline, satisfying the preservation constraint but weakening the attack on the targeted task. and the solution of problem (2) will satisfy the constraints of problem (1). However, the strength of attacking the targeted task will be compromised.

In SMTA^2, the core challenge lies in enforcing strict performance constraints on non-targeted tasks while maximizing the degradation of the targeted task. This naturally leads to a constrained multi-objective optimization problem, where the loss of the targeted task is maximized subject to preservation constraints on the remaining tasks. To address this challenge, we design two strategies for searching appropriate task weights in the joint loss function: a *manual* search strategy and a more efficient *automated* optimization strategy.

Manual Weight Searching. In the manual approach, the weight of the targeted task is fixed to emphasize its degradation, while the weights of non-targeted tasks are manually adjusted to enforce preservation constraints. The procedure begins by assigning small-magnitude weights to non-targeted tasks. After performing the adversarial attack, if the performance of any non-targeted task violates the preservation constraint (i.e., performance drops below the baseline), its corresponding weight magnitude is increased in the next trial. This iterative process continues until all non-targeted tasks satisfy the constraint.

Although this approach guarantees the existence of a feasible solution to Problem (1), it relies heavily on trial-and-error hyperparameter tuning. Partic-

ularly, as the number of tasks increases, the search space grows rapidly, making manual tuning computationally expensive and time-consuming.

Algorithm 1 An automated approach for searching the weighting factors in the loss function

Input: input data x, the ground-truth label of the i-th task y_i, the initialized weighting factor w_i of the i-th task, model parameters and the corresponding loss $L(x, y) = \sum_{i=1}^{m} w_i L_i(x, y_i)$, total steps S in the adversarial attack, the targeted task i_t, the step size λ to adjust weighting factors, and the attenuation coefficient α.
Initialize $\lambda_i = \lambda$ for all $i \neq i_t$.
for $s = 1, 2, \ldots, S$ **do**
  perform one step PGD (or APGD or IFGSM) attack on loss $L(x, y)$ to generate the updated adversarial noise δ.
  for $i = 1, 2, \ldots, m$ **do**
    if $i \neq i_t$ **then**
      $w_i = w_i - \lambda_i * (L_i(x + \delta, y_i) - L_i(x, y_i))$.
      if $L_i(x + \delta, y_i) \leq L_i(x, y_i)$ **then**
        $\lambda_i = \lambda_i * \alpha$
      end if
    end if
  end for
end for

Automated Weight Optimization. To overcome the inefficiency of manual tuning, we propose an automated weight optimization strategy, summarized in Algorithm 1. Unlike the manual approach, the automated weight searching approach dynamically updates task weights during each attack iteration, enabling real-time enforcement of preservation constraints. Specifically, at each attack step, we compute the loss difference between the adversarial and original inputs for each non-targeted task. The weight of each non-targeted task is updated proportionally to the loss difference. If adversarial loss increases (indicating potential degradation), the corresponding weight is adjusted to counteract this effect.

To accelerate convergence, we introduce a task-specific step size λ_i, initialized uniformly. Once a non-targeted task satisfies the preservation constraint (i.e., its adversarial loss no longer exceeds the original loss), its step size is attenuated by a coefficient $\alpha \in (0,1)$, reducing oscillations and stabilizing the search process. This adaptive attenuation mechanism allows the algorithm to quickly converge to a feasible region while maintaining stability.

Complexity Analysis. Let m denote the number of tasks and S denote the total attack iterations. Each step of the attack requires one forward and backward pass to update the perturbation, incurring the same dominant cost attacks based on Projected Gradient Descent (PGD) [29], Auto-PGD [9] or Iterative Fast Gradient Sign Method (IFGSM) [22]. The additional overhead of SMTA^2 comes from

updating task weights, which involves computing loss differences for (m-1) non-targeted tasks and simple scalar updates. This adds only $\mathcal{O}(m)$ operations per iteration, which is negligible compared to the backpropagation cost. Therefore, the overall time complexity remains $\mathcal{O}(S \cdot \text{Backprop})$, making SMTA^2 asymptotically equivalent to conventional attacks in complexity. Compared with manual weight searching strategy, the automated approach substantially improves efficiency and scalability, especially when the number of tasks increases.

Overall, the automated approach performs online constrained optimization over task weights, significantly reducing parameter tuning time and improving scalability to multi-task settings with larger task counts. Empirically, it consistently finds feasible and stable solutions that satisfy stealthy attack requirements while preserving the performance among the non-targeted tasks.

4 Experiments

To evaluate SMTA^2, we consider two schemes for searching optimal task weights: the *manual* scheme and the *automated* scheme. We adopt three widely-used attack algorithms: Projected Gradient Descent (PGD) [29], Iterative Fast Gradient Sign Method (IFGSM) [22], and a recently established stronger attack: Auto-PGD (APGD) [9]. For the attack norms, the PGD attack is implemented under both L_2 and L_∞ norms, the APGD is implemented under both L_1 and L_2 norms, and the IFGSM is under L_∞ norm. We compare manual and automated weight-search scheme under the proposed loss formulation, respectively. Due to space limitation, the results of IFGSM and more visualization results are provided in the supplementary material.

4.1 Experimental Setup

We evaluate SMTA^2 on two widely used multi-task image benchmarks, i.e., NYUv2 [34] and Cityscapes [7]. Both non-stealthy and stealthy attack settings are considered. Non-stealthy attacks degrade the targeted task and inevitably affects other tasks, causing global performance drops [17, 41]. However, SMTA^2 aims to significantly impair the targeted task while strictly maintaining non-targeted task performance. We further incorporate adversarial training to assess robustness against defended models. All the images are resized to 288×384 for NYUv2 and 256×512 for Cityscapes. All the experiments are conducted on 8 NVIDIA RTX A6000 GPUs.

4.2 Experimental Procedure

The proposed SMTA^2 aims to selectively degrade a targeted task while strictly preserving non-targeted tasks. Accordingly, we adopt an evaluation criterion that measures attack effectiveness under full preservation constraints.

The attack strength is controlled via perturbations according to different L_p norms. Specifically, for the L_∞ norm in PGD and IFGSM, ϵ is set to $2/255$, $4/255$,

Table 1: Experimental results of non-stealthy and SMTA^2 framework by manual and automated solutions of **PGD** L_∞ attack algorithm on **NYUv2** dataset. Non-attacked performance is represented by ϵ=0. The performances of the targeted task are underlined. Values affected by the attack (i.e., degraded during the attack) are marked in blue, while those not influenced by the attack are highlighted in red. (↑) means higher is better and (↓) means lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]
ϵ=0	46.56	40.57	23.41	46.56	40.57	23.41	46.56	40.57	23.41
ϵ=2/255	26.21	51.78	26.65	26.54	37.92	23.00	26.46	40.28	23.12
	35.08	106.93	28.45	47.53	106.47	23.19	47.63	106.45	23.40
	37.76	54.99	40.36	48.17	38.23	41.10	46.76	34.76	39.99
ϵ=4/255	17.04	63.80	30.97	18.30	37.30	23.22	18.33	37.45	23.31
	24.15	167.40	34.39	47.08	165.27	22.83	47.67	168.69	23.36
	28.88	71.44	53.88	47.86	37.26	53.47	46.98	39.58	53.87
ϵ=8/255	10.11	78.36	36.23	12.96	37.86	22.60	12.32	39.56	23.21
	14.58	240.52	41.16	47.14	229.64	21.96	52.83	225.11	22.36
	18.91	93.36	68.05	47.66	39.93	66.60	46.93	38.03	63.29

and 8/255, respectively. The perturbation ϵ is set to 10 and 20 for L_1 norm in APGD and is set to 5 and 10 for L_2 norm in APGD and PGD. The manual strategy searches for feasible task weights, where negative weights for non-targeted tasks help enforce preservation. The automated strategy (Algorithm 1) could dynamically update task weights for efficient and stable convergence. Furthermore, we evaluate SMTA^2 on Adversarially Trained (AT) models. Results show that both strategies maintain non-targeted task performance while achieving strong targeted attacks, even under enhanced robustness.

Evaluation Metrics. On NYUv2, we evaluate semantic segmentation, depth estimation, and surface normal prediction tasks using mIoU, absolute error (aErr.), and mean angular distance (mDist.), respectively. On Cityscapes, mIoU is used for semantic segmentation and part segmentation tasks, and aErr. is used for disparity estimation. All the settings regarding metrics follow [26]. Each task is treated as the targeted task in turn, with others considered as non-targeted.

4.3 Main Results and Analysis

Attack Results of the Undefended Model Results on NYUv2 under PGD L_∞ attacks (Table 1) show that non-stealthy attacks degrade all tasks, whereas both manual and automated SMTA^2 selectively degrade the targeted task while preserving non-targeted tasks at baseline or better performance. Even at small perturbation levels ($\epsilon = 2/255$ or $4/255$), SMTA^2 achieves attack strength comparable to non-stealthy methods. As ϵ increases, the performance gap remains limited, indicating stable and robust selective attack behavior.

Table 2: Results of non-stealthy attacks and SMTA^2 on **Cityscapes** dataset under **PGD** L_∞ attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]
ϵ=0	54.20	51.82	81.51	54.20	51.82	81.51	54.20	51.82	81.51
ϵ=2/255	27.98	43.82	98.72	25.34	52.51	79.71	25.38	52.97	80.30
	41.24	30.00	92.13	54.57	27.29	80.04	54.25	27.22	76.25
	38.47	42.05	278.11	54.39	51.93	367.34	56.16	52.30	359.65
ϵ=4/255	20.48	35.01	119.67	17.25	52.67	81.00	17.08	52.77	81.21
	33.98	21.69	111.19	54.59	18.02	80.23	54.36	18.19	80.00
	25.82	26.68	490.15	55.22	51.85	679.89	56.44	52.02	644.12
ϵ=8/255	11.31	19.24	164.19	10.62	51.84	78.72	10.02	52.07	81.50
	25.77	14.65	153.10	55.13	12.63	79.12	54.27	12.22	80.97
	16.05	14.18	824.75	55.10	52.54	1014.74	57.36	52.10	1008.44

Table 3: Results of non-stealthy attacks and SMTA^2 on **NYUv2** dataset under **PGD** L_2 attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]
ϵ=0	46.56	40.57	23.41	46.56	40.57	23.41	46.56	40.57	23.41
ϵ=5	14.70	70.56	33.29	19.30	40.18	23.25	19.44	37.65	23.35
	21.04	197.61	36.48	46.71	174.81	23.26	46.74	174.57	22.88
	24.29	81.77	63.31	50.21	40.01	71.54	47.15	39.37	54.28
ϵ=10	7.70	87.83	39.33	13.03	39.78	23.19	12.84	38.05	23.22
	11.20	290.69	44.18	46.79	251.49	23.30	46.68	251.15	22.56
	14.18	107.80	81.43	47.79	39.22	71.54	49.71	38.16	68.21

The results using Cityscapes dataset (Table 2) exhibit consistent trends. SMTA^2 maintains non-targeted task performance while achieving strong targeted degradation, often matching or surpassing non-stealthy baselines. The automated weight search performs comparably to manual tuning with significantly higher efficiency, demonstrating scalability and practicality.

Figures 2 and 3 visualize the results of stealthy adversarial attacks on the NYUv2 and Cityscapes datasets with ϵ=4/255, respectively. The perturbations introduce only subtle pixel-level changes, yet they significantly degrade the performance of the targeted task while effectively preserving the outputs of non-targeted tasks. These examples demonstrate that SMTA^2 successfully achieves selective task degradation without introducing noticeable visual artifacts, particularly for safety-critical objects such as vehicles and pedestrians.

Table 4: Results of non-stealthy attacks and SMTA^2 on **Cityscapes** dataset under **PGD** L_2 attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]
ϵ=0	54.20	51.82	81.51	54.20	51.82	81.51	54.20	51.82	81.51
ϵ=5	18.84	33.17	126.93	21.48	53.70	80.36	21.20	52.36	81.44
	31.08	16.60	144.51	55.72	19.56	80.21	56.49	19.71	81.32
	22.03	17.73	921.67	56.39	53.45	652.57	57.69	53.47	607.56
ϵ=10	10.52	17.98	167.64	14.85	55.68	81.43	14.25	52.99	80.35
	22.95	9.86	274.18	55.58	13.70	81.18	56.11	13.49	81.39
	12.93	8.33	1611.60	55.24	52.00	1057.77	57.53	51.99	1026.31

Table 5: Experimental results of SMTA^2 Auto method with APGD attacks with L_1 & L_2 norms on Cityscapes. The targeted results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

APGD L_1	**Seg**(↑)	**Part**(↑)	**Disp**(↓)	APGD L_2	**Seg**(↑)	**Part**(↑)	**Disp**(↓)
ϵ=0	54.20	51.82	81.51	ϵ=0	54.20	51.82	81.51
ϵ=10	18.84	51.88	78.60	ϵ=5	23.21	51.98	80.76
	55.69	19.90	80.16		54.44	24.08	81.26
	55.85	52.48	461.82		54.87	52.71	387.33
ϵ=20	13.55	52.84	80.64	ϵ=10	15.03	52.53	81.10
	55.28	13.58	80.23		56.20	14.12	80.36
	56.74	53.04	827.69		55.99	52.08	690.18

The SMTA^2 framework is further evaluated under PGD-L_2 and stronger APGD attacks with both L_1 and L_2 norms. As shown in Tables 3, 4, and 5, SMTA^2 consistently achieves effective task-selective attacks across different perturbation budgets and attack algorithms. The targeted task is significantly degraded, while non-targeted tasks remain hold clean performance. Moreover, the automated searching generally matches or outperforms manual tuning. These results demonstrate that SMTA^2 generalizes well across different attack paradigms and norm constraints, maintaining strong attack effectiveness and stealthy task preservation under multiple attack algorithms.

Visualized Results of Attack Effectiveness Figure 4 visualizes the attack effectiveness of non-stealthy and automated SMTA^2 methods under PGD L_∞ attacks using radar charts, which depict the ratio between post-attack and pre-attack performance. For metrics where higher values indicate better performance (e.g., mIoU), the ratio is computed as post-attack divided by pre-attack performance, while for metrics where lower values are preferable (e.g., aErr), the inverse ratio is used. Smaller ratios (closer to the center) indicate stronger attack effectiveness, whereas larger ratios (closer to the boundary) correspond to better

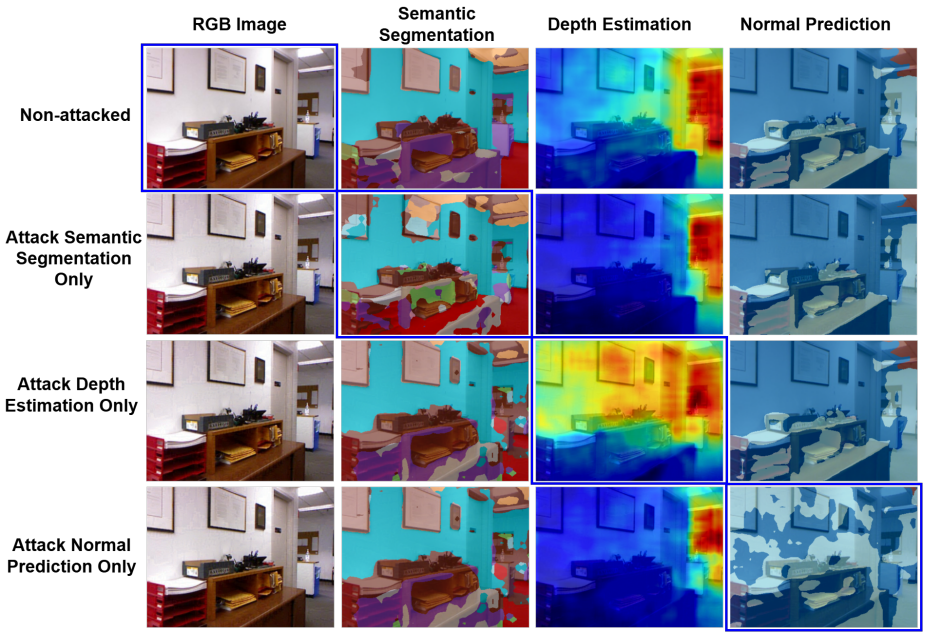

Fig. 2: Visualization results of non-attacked outputs and SMTA^2 attacks on NYUv2 dataset (ϵ=4/255). Each row corresponds to a targeted task, while columns show the targeted and non-targeted task outputs. SMTA^2 effectively degrades the targeted task while preserving non-targeted tasks.

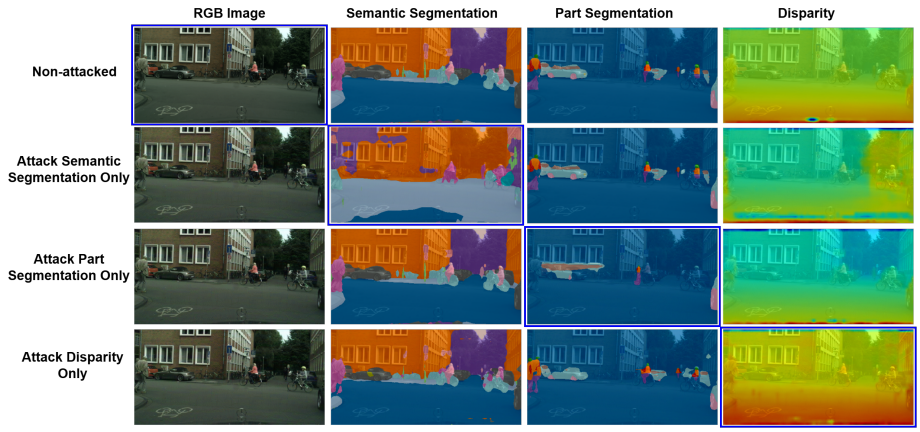

Fig. 3: Visualization results of non-attacked outputs and SMTA^2 attacks on Cityscapes dataset (ϵ=4/255). Each row corresponds to a targeted task, while columns show the targeted and non-targeted task outputs..

retained performance. The dashed blue line denotes the clean baseline with all ratios equal to 1.

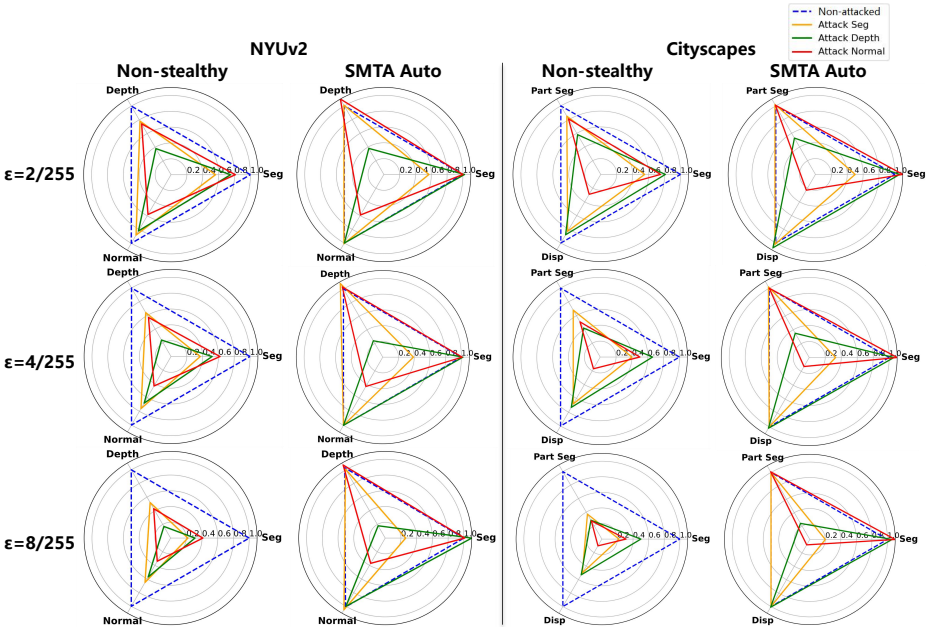

Fig. 4: Radar images of undefended PGD L_∞ attack effectiveness of non-stealthy and SMTA^2 Automated methods of NYUv2 and Cityscapes datasets.

As shown, non-stealthy attacks significantly degrade all tasks, even when only a single task is targeted, indicating substantial collateral damage. In contrast, automated SMTA^2 selectively degrades the targeted task while strictly preserving, or even improving, the performance of non-targeted tasks, demonstrating its strong task-selective attack capability and stealthiness.

Attack Results of Adversarially Trained Models To evaluate robustness under defensive settings, we further conduct experiments on Adversarially Trained (AT) models obtained via PGD-based adversarial training. Tables 6 and 7 present results under PGD L_∞ attacks on NYUv2 and Cityscapes using the same perturbation budgets (ϵ=2/255, 4/255, 8/255). Across all settings, SMTA^2 consistently preserves non-targeted tasks while effectively degrading the targeted one, demonstrating stable task-selective attack capability. Both manual and automated variants show reliable performance.

Similar trends are observed under PGD L_2 attacks (Tables 8 and 9). While non-stealthy attacks degrade all tasks, SMTA^2 maintains baseline performance for non-targeted tasks and achieves consistent targeted degradation. Despite enhanced robustness from adversarial training, SMTA^2 remains effective, confirming its reliability under strong defense settings.

Table 6: Results of non-stealthy attacks and SMTA^2 under PGD adversarial training on **NYUv2** dataset using **PGD** L_∞ attack. Non-attacked performance is denoted by ϵ=0. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]
ϵ=0	46.56	40.57	23.41	46.56	40.57	23.41	46.56	40.57	23.41
ϵ=2/255	33.21	52.27	28.00	35.67	36.82	23.00	33.66	39.94	23.36
	31.00	61.97	30.03	47.40	54.65	23.01	47.39	56.05	22.41
	31.04	58.55	31.12	49.54	40.17	32.82	47.79	38.76	32.87
ϵ=4/255	27.37	59.42	30.28	30.91	39.26	23.37	30.99	38.98	23.31
	34.58	63.36	28.46	47.87	59.98	23.37	48.13	61.50	23.20
	30.41	59.41	32.59	46.84	40.64	31.06	46.76	39.17	30.73
ϵ=8/255	23.33	61.37	30.97	27.45	39.87	22.86	26.83	38.67	23.17
	28.68	76.32	31.15	48.38	69.60	22.37	46.78	73.72	23.01
	28.90	61.44	35.85	48.00	39.95	35.27	47.38	39.76	34.42

Table 7: Results of non-stealthy attacks and SMTA^2 under PGD adversarial training on **Cityscapes** dataset using **PGD** L_∞ attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]
ϵ=0	54.20	51.82	81.51	54.20	51.82	81.51	54.20	51.82	81.51
ϵ=2/255	33.21	38.61	101.67	42.26	52.01	80.69	42.09	52.00	81.23
	40.59	41.09	94.69	55.12	43.61	80.68	55.20	43.18	80.99
	33.69	37.34	103.34	54.78	52.19	98.89	54.83	52.52	99.63
ϵ=4/255	30.98	37.93	102.75	30.29	52.45	81.41	31.22	53.10	81.30
	34.17	34.78	101.93	54.78	30.82	81.22	54.96	31.16	80.66
	34.41	38.24	112.85	54.68	51.94	115.52	55.34	52.33	114.18
ϵ=8/255	27.12	36.25	105.51	32.56	53.10	81.21	28.69	52.79	81.08
	32.32	30.38	103.75	54.77	33.95	81.04	56.05	30.62	80.21
	32.80	36.69	127.97	54.67	51.97	112.91	56.18	52.56	122.42

5 Conclusion

We propose Stealthy Multi-Task Adversarial Attacks (SMTA^2), a principled framework for selectively degrading a targeted task while strictly preserving non-targeted tasks in multi-task learning. Unlike conventional adversarial attacks that cause indiscriminate performance collapse, SMTA^2 formulates selective degradation as a constrained multi-objective optimization problem and solves it via adaptive weight optimization. Despite the strong inter-task coupling in shared representations in multi-task learning, SMTA^2 reliably achieves

Table 8: Results of non-stealthy attacks and SMTA^2 under PGD adversarial training on **NYUv2** dataset using **PGD** L_2 attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]	Segment [mIoU(↑)]	Depth [aErr(↓)]	Normal [mDist(↓)]
ϵ=0	46.56	40.57	23.41	46.56	40.57	23.41	46.56	40.57	23.41
ϵ=5	25.65	60.49	30.72	34.25	40.25	23.02	34.20	40.33	23.26
	29.26	71.26	30.96	46.83	61.80	23.34	46.66	61.97	22.94
	29.32	60.86	34.46	46.93	39.69	31.52	47.02	40.56	35.21
ϵ=10	20.16	64.05	32.24	24.88	39.68	23.20	24.58	38.93	22.42
	29.97	88.34	30.98	46.92	78.81	22.82	46.94	83.19	22.37
	26.35	64.80	40.39	47.00	39.36	38.81	47.25	38.66	38.55

Table 9: Results of non-stealthy attacks and SMTA^2 under PGD adversarial training on **Cityscapes** dataset using **PGD** L_2 attack. The targeted-task results are underlined. Blue and red indicate affected and preserved values, respectively. (↑)/(↓) denote higher/lower is better.

Method	Non-stealthy			SMTA^2 Manual			SMTA^2 Auto		
Tasks	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]	Segment [mIoU(↑)]	Part Seg [mIoU(↑)]	Disparity [aErr(↓)]
ϵ=0	54.20	51.82	81.51	54.20	51.82	81.51	54.20	51.82	81.51
ϵ=5	26.03	35.76	108.67	26.28	52.84	81.03	26.22	52.13	81.44
	35.67	33.06	98.76	55.72	30.26	80.81	56.13	31.17	81.11
	36.84	39.65	135.95	56.20	52.11	133.21	56.92	53.41	128.56
ϵ=10	21.06	31.83	119.86	29.46	52.88	80.93	28.51	52.52	80.78
	26.90	21.37	116.82	54.96	28.57	81.40	55.15	26.76	80.58
	26.22	27.53	229.00	56.31	53.46	160.88	56.74	53.40	165.08

targeted attack effectiveness without collateral damage, even under adversarial trained scenarios. Experiments demonstrate that SMTA^2 consistently achieves strong attack effectiveness on targeted tasks while preserving non-targeted tasks across undefended and adversarially trained settings. This work demonstrates that stealthy, task-selective attacks are both feasible and practically solvable, opening a new direction for controlled adversarial machine learning in multi-task systems.

Acknowledgments

This research was supported by the National Science Foundation under award CNS-2245765.

References

1. Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2442–2452 (2016)
2. Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W.: Hidden voice commands. In: 25th USENIX security symposium (USENIX security 16). pp. 513–530 (2016)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
4. Caruana, R.: Multitask learning. Machine learning **28**, 41–75 (1997)
5. Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D.: A survey on adversarial attacks and defences. CAAI Transactions on Intelligence Technology **6**(1), 25–45 (2021)
6. Chen, Z., Wang, C., Crandall, D.: Semantically stealthy adversarial attacks against segmentation models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4080–4089 (2022)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
8. Crawshaw, M.: Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796 (2020)
9. Croce, F., Hein, M.: Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In: International Conference on Machine Learning. pp. 2201–2211. PMLR (2021)
10. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing **20**(1), 30–42 (2011)
11. Du, A., Chen, B., Chin, T.J., Law, Y.W., Sasdelli, M., Rajasegaran, R., Campbell, D.: Physical adversarial attacks on an aerial imagery object detector. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1796–1806 (2022)
12. Ghamizi, S., Cordy, M., Papadakis, M., Le Traon, Y.: Adversarial robustness in multi-task learning: Promises and illusions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 697–705 (2022)
13. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
14. Guo, J., Lei, L., Sun, H., Qin, M., Yu, H., Zhang, T.: A min-max optimization framework for sparse multi-task deep neural network. Neurocomputing p. 130865 (2025)
15. Guo, J., Li, L., Yang, H., Geng, B., Yu, H., Qin, M., Zhang, T.: Robust multi-task adversarial attacks using min-max optimization. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
16. Guo, J., Sun, H., Qin, M., Yu, H., Zhang, T.: A min-max optimization framework for multi-task deep neural network compression. In: 2024 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–5. IEEE (2024)

17. Guo, P., Xu, Y., Lin, B., Zhang, Y.: Multi-task adversarial attack. arXiv preprint arXiv:2011.09824 (2020)
18. He, K., Zhang, J., Yan, Y., Xu, W., Niu, C., Zhou, J.: Contrastive zero-shot learning for cross-domain slot filling with adversarial attack. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 1461–1467 (2020)
19. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine **29**(6), 82–97 (2012)
20. Hu, J., Ran, H., Sun, C., Zhang, Q., Wang, G.: Dynamic gradient fusion method with local spatial and multi-scale frequency transformations for transferable adversarial attacks. Expert Systems with Applications p. 131677 (2026)
21. Jia, J., Zhang, W., Wang, R., Yan, D.: Adaptive multi-task adversarial attacks on audio tasks. Expert Systems with Applications p. 133000 (2026)
22. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. International Coference of Learning Representations (2018)
23. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
24. Li, L., Yang, H., Guo, J., Yu, H., Qin, M., Zhang, T.: Task-aware federated multi-task learning. In: Proceedings of the 7th ACM International Conference on Multimedia in Asia. pp. 1–7 (2025)
25. Li, L., Yang, H., Guo, J., Yu, H., Qin, M., Zhang, T.: Fedsta: Spatio-temporal alternation for efficient federated multi-task learning. IEEE Transactions on Circuits and Systems for Video Technology (2026)
26. Liu, S., James, S., Davison, A.J., Johns, E.: Auto-lambda: Disentangling dynamic task relationships. arXiv preprint arXiv:2202.03091 (2022)
27. Liu, X., Li, J., Ma, J., Sun, H., Xu, Z., Zhang, T., Yu, H.: Deep transfer learning for intelligent vehicle perception: A survey. Green Energy and Intelligent Transportation p. 100125 (2023)
28. Long, K., Sheng, Z., Shi, H., Li, X., Chen, S., Ahn, S.: Physical enhanced residual learning (perl) framework for vehicle trajectory prediction. Communications in Transportation Research **5**, 100166 (2025)
29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
30. Mao, C., Chiquier, M., Wang, H., Yang, J., Vondrick, C.: Adversarial attacks are reversible with natural supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 661–671 (2021)
31. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2017)
32. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 427–436 (2015)
33. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. Advances in neural information processing systems **31** (2018)
34. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12. pp. 746–760. Springer (2012)

35. Sun, H., Fu, L., Li, J., Guo, Q., Meng, Z., Zhang, T., Lin, Y., Yu, H.: Defense against adversarial cloud attack on remote sensing salient object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 8345–8354 (2024)
36. Xu, K., Cheng, X., Qiao, J., Li, J.T., Ji, K.X., Zhong, J.Y., Tian, P., Mi, J.X.: Adversarial attacks to manipulate target localization of object detector. IEEE Access (2024)
37. Xu, Q., Zhou, L., Zhong, X., Zhang, F., Tian, J., Yu, X., Huang, R.: Etv-attack: Efficient text-driven visual-variable adversarial attacks on visual question answering with pre-trained language models. Pattern Recognition p. 113202 (2026)
38. Yang, H., Li, L., Guo, J., Li, B., Qin, M., Yu, H., Zhang, T.: Da3d: Domain-aware dynamic adaptation for all-weather multimodal 3d detection. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 2150–2158 (2025)
39. Yu, J., Dai, Y., Liu, X., Huang, J., Shen, Y., Zhang, K., Zhou, R., Adhikarla, E., Ye, W., Liu, Y., et al.: Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras. arXiv preprint arXiv:2404.18961 (2024)
40. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International conference on machine learning. pp. 7472–7482. PMLR (2019)
41. Zhang, L., Liu, X., Mahmood, K., Ding, C., Guan, H.: Attacking all tasks at once using adversarial examples in multi-task learning. Neurocomputing p. 131503 (2025)
42. Zhang, Z., Yu, W., Yu, M., Guo, Z., Jiang, M.: A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. pp. 943–956 (2023)
43. Zhe, Y., Nagaike, R., Nishiyama, D., Fukuchi, K., Sakuma, J.: Adversarial attacks on hidden tasks in multi-task learning. arXiv preprint arXiv:2405.15244 (2024)