Adversarial Robustness in Multi-Task Learning: Promises and Illusions

Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon

University of Luxembourg
salah.ghamizi@uni.lu, maxime.cordy@uni.lu, michail.papadakis@uni.lu, yves.letraon@uni.lu

Abstract

Vulnerability to adversarial attacks is a well-known weakness of Deep Neural networks. While most of the studies focus on single-task neural networks with computer vision datasets, very little research has considered complex multi-task models that are common in real applications. In this paper, we evaluate the design choices that impact the robustness of multi-task deep learning networks. We provide evidence that blindly adding auxiliary tasks, or weighing the tasks provides a false sense of robustness. Thereby, we tone down the claim made by previous research and study the different factors which may affect robustness. In particular, we show that the choice of the task to incorporate in the loss function are important factors that can be leveraged to yield more robust models.

We provide the appendix, all our algorithms, models, and open source-code at *https://github.com/yamizi/taskaugment*

Introduction

While most research on computer vision focuses on single-task learning, many real applications (especially in robotics (Radwan, Valada, and Burgard 2018), autonomous vehicle (Yu et al. 2020), privacy (Ghamizi et al. 2019) and medical diagnosis (Xu and Yang 2011)) require learning to perform different tasks from the same inputs. For instance, an autonomous vehicle processes multiple computer vision tasks to navigate properly (Sax et al. 2018; Savva et al. 2019), such as object segmentation and depth estimation.

To meet such requirements, one can build and train on a specific model for each task. However, previous studies have shown that multi-task learning, i.e. building models that learn to perform multiple tasks simultaneously, yields better performance than learning the individual tasks separately (Zhang and Yang 2017; Vandenhende et al. 2021). The intuition behind these results is that tasks that share similar objectives benefit from the information learned by each other. However, while the performance of multi-task learning approaches on clean images has seen major improvements recently(Standley et al. 2020), the security of these models, in particular their vulnerability to adversarial attacks, has been barely studied.

The phenomena of *adversarial attacks* has first been introduced by (Biggio et al. 2013) and (Szegedy et al. 2013) and has since gathered the interest of researchers to propose new attacks (Goodfellow, Shlens, and Szegedy 2014a; Madry et al. 2017), defense mechanisms (Kurakin, Goodfellow, and Bengio 2016; He et al. 2017), detection mechanisms (Metzen et al. 2017), or to improve transferability across different networks (Tramèr et al. 2017; Inkawhich et al. 2019).

(Mao et al. 2020) pioneered the research on adversarial attacks against multi-task models. Their main result is that, under specific conditions, making models learn additional *auxiliary* tasks leads to both an increase in clean performances *and* improves the robustness of models.

In this paper, we pursue the endeavor of Mao et al. and study whether their conclusions hold in a larger spectrum of settings. Surprisingly, our experimental study shows that adding more tasks does not consistently increase robustness, and may even have negative effects. We even reveal some experimental parameters, such as the attack norm distance, can annihilate the validity of the previous theoretical results. Overall, this indicates that increasing the number of tasks only gives *a false sense of robustness*.

In face of this disillusionment, we investigate the different factors that may explain the discrepancies between our results and that of the previous study. We demonstrate that the contribution of each task to the vulnerability of the model can be qualified, and that what matters is not the number of added tasks but the *marginal vulnerability* of these tasks.

Following this finding, we investigate remedies to reinstate the addition of auxiliary tasks as an effective means of improving robustness. Firstly, following the recommendation of previous research on increasing the clean performance of multi-task models (Chen et al. 2018; Standley et al. 2020), we show that adjusting the task weights to make vulnerable tasks less dominant yields drastic robustness improvement but does so only against non-adaptive adversarial attacks: Auto-PGD (Croce and Hein 2020) and Weighted Gradient Descent – a new adaptive attack that we propose to adapt the produced perturbation to task weights – annihilate the benefits of weight adjustment.

Secondly, we show that a careful selection of the tasks suffices to ensure that model robustness increases. Given a target model, determining which combination of tasks is optimal can be costly. We propose different methods to approximate the gain in robustness and show that they strongly correlate with the robustness of the target model.

To summarize, our contributions are:

- We show that adding auxiliary tasks is not a guarantee of improving robustness and identify the key factors explaining the discrepancies with the original study.
- We refine the theory of Mao et al. through the concept of *marginal adversarial vulnerability* of tasks. Leaning on this, we demonstrate that the inherent vulnerability of tasks plays a central role in the model robustness.
- We empirically show that a careful weighting of the tasks can act as a remedy and offer the benefits initially promised by previous research. However, it does not provide increased robustness against adaptive attacks.
- We propose a set of surrogates to efficiently evaluate the robustness of a combination of tasks.

Related Work

Multi-task learning (MTL). Multi-task learning leverages shared knowledge across multiple-tasks to learn models with higher efficiency (Vandenhende et al. 2021; Standley et al. 2020). A multi-task model is commonly made of an encoder block, that learns shared parameters across the tasks and a decoder part that branches out into task-specific heads.

(Vandenhende et al. 2021) recently proposed a new taxonomy of MTL approaches to split the approaches based on where the task interactions take place. They differentiated between approaches that are *encoder-focused* where some information is shared across tasks at the encoder stage and approaches that are *decoder-focused* where some interactions still happen across the heads of the tasks. They organized MTL research around three main questions: (1) when does the task learning interact, (2) how can we optimize the learning, and (3) which tasks should be learned together.

Our work complement this research by tackling a fourth question: How to optimize the robustness of MTL.

Adversarial attacks An adversarial attack is the process of intentionally introducing perturbations on the inputs of a machine learning model to cause wrong predictions. One family of adversarial attacks is *poisoning attacks* (Biggio, Nelson, and Laskov 2012) where the inputs targeted are the training set and occur during the learning step, while *evasion attacks* (Biggio et al. 2013) focus on the inference step.

One of the earliest attacks is the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014a). It adds a small perturbation η to the input of a neural network, which is defined as:

$$\eta = \epsilon \operatorname{sign}(\nabla_x \mathscr{L}_i(\theta, x, y_i)), \tag{1}$$

where θ are the parameters of the network, x is the input data, y_i is its associated target, $\mathscr{L}(\theta, x, y_i)$ is the loss function used, and ϵ the strength of the attack. Following Goodfellow, other attacks were proposed, first by adding iterations (I-FGSM) (Kurakin, Goodfellow, and Bengio 2016), projections and random restart (PGD) (Madry et al. 2017), momentum (MIM) (Dong et al. 2018) and constraints (CoEva2) (Ghamizi et al. 2020; Dyrmishi et al. 2022).

These algorithms can be used without any change on a multi-task model if the attacker only focuses on a single task.

Problem Formulation

Preliminaries

Let $\mathscr{M}$ a multi-task model with tasks $\mathscr{T} = \{t_1, ..., t_M\}$. For each input example x, we denote by $\bar{y}$ the corresponding ground-truth label and we have $\bar{y} = (y_1, ..., y_i, y_M)$ where y_i is the corresponding ground truth for task i.

We focus on hard sharing multi-task learning, where the models are made of one encoder (backbone network common to all tasks) $E(.)$ and task-specific decoders $D_i(.)$. Each task is associated with a loss $\mathscr{L}_i$; $\mathscr{L}_i(x, y_i) = l_i(D_i(E(x)), y_i)$ where l_i is a loss function tailored to the task i. For instance, we can use cross-entropy losses for classification tasks, and L1 losses for regression tasks.

The total loss $\mathscr{L}$ of our multi-task model is a weighted sum of the individual losses $\mathscr{L}_i$ of each task:

$$\mathscr{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathscr{L}_i(x, y_i)$$

where $\{w_1 \dots w_M\}$ are the weights of the tasks, either set manually or optimized during training (Chen et al. 2018).

Single-task adversarial attacks In this use case, the adversary tries to increase the error of one single task. This threat model can represent scenarios where the attacker has access to one task only or aims to perturb one identified task. The objective of the attacker can then be modeled as:

$$\underset{\delta}{\operatorname{argmax}}\, \mathscr{L}_i(x + \delta, y_i) \text{ s.t. } \|\delta\|_p \leq \epsilon \tag{2}$$

where i is the index of the targeted task and ϵ the maximum perturbation size using a norm p.

Multi-task adversarial attacks In multi-task adversarial attacks, the adversary aims to increase the error of multiple outputs all at once. This captures scenarios where the adversary does not have fine-grained access to the individual tasks or where the final prediction of the system results from the combination of multiple tasks. Therefore, the adversary has to attack all tasks together.

Given a multi-task model $\mathscr{M}$, an input example x , and $\bar{y}$ the corresponding ground-truth label, the attacker seeks the perturbation δ that will maximize the joint loss function of the attacked tasks – i.e. the summed loss, within a p-norm bounded distance ϵ.

The objective of the attack is then:

$$\underset{\delta}{\operatorname{argmax}}\, \mathscr{L}(x + \delta, \bar{y}) \text{ s.t. } \|\delta\|_p \leq \epsilon \tag{3}$$

Adversarial vulnerability of multi-task models (Simon-Gabriel et al. 2019) introduced the concept of adversarial vulnerability to evaluate and compare the robustness of single-task models and settings. Mao et al. extended it to multi-task models as follow:

Definition 1. *Let $\mathscr{M}$ be a multi-task model. $\mathscr{T}' \subseteq \mathscr{T}$ a subset of its tasks and $\mathcal{L}_{\mathscr{T}'}$ the joint loss of tasks in $\mathscr{T}'$. Then, we denote by $\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)]$ the* adversarial vulnerability *of $\mathscr{M}$ on $\mathscr{T}'$ to an ϵ-sized $\|.\|_p$-attack, and define it as the average increase of $\mathcal{L}_{\mathscr{T}'}$ after attack over the whole dataset:*

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] = \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \mathcal{L}_{\mathscr{T}'}(x+\delta, \bar{y}) - \mathcal{L}_{\mathscr{T}'}(x, \bar{y}) \mid\right]$$

This definition matches the definitions of previous work (Goodfellow, Shlens, and Szegedy 2014b; Sinha et al. 2017) of the robustness of deep learning models: the models are considered vulnerable when a small perturbation causes a large average variation of the joint loss.

Similarly, we call *adversarial task vulnerability* of a task i the average increase of $\mathcal{L}_{\mathscr{T}'}(x, y_i)$ after an attack.

Assuming that the variation δ is small, (Mao et al. 2020) proposed the following theorem using the (Simon-Gabriel et al. 2019) first-order Taylor expansion in ϵ:

Theorem 2. *Consider a multi-task model $\mathscr{M}$ where an attacker targets $\mathscr{T} = \{t_1, ..., t_M\}$ tasks uniformly weighted, with an ϵ-sized $\|.\|_p$-attack. If the model is converged, and the gradient for each task is i.i.d. with zero mean and the tasks are correlated, the adversarial vulnerability of the model can be approximated as*

$$\mathbb{E}_x[\delta\mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M}\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_j)}{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_i)}}{M}} \qquad (4)$$

where K is a constant dependant of ϵ and the attacked tasks, and $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, and $\mathrm{Cov}\,(\mathbf{r}_i, \mathbf{r}_j)$ the covariance between the two gradients $\mathbf{r}_i, \mathbf{r}_j$.

Proof. Appendix A.1 and A.2 □

This theorem indicates that under the specific assumptions described above, (1) increasing the number of tasks reduces the adversarial vulnerability of a multi-task model and (2) even more when these tasks are uncorrelated.

Research Questions and Methodology

Our research endeavor stems from the hypothesis that the assumptions supporting the above theorem are too restrictive to be met in practice. The existence of settings where these assumptions do not hold would tone down the validity of the results of (Mao et al. 2020) and raise anew the question of how to achieve robust multi-task learning.

Accordingly, our first research question investigates if the assumptions and results of Theorem 2 are confirmed in a variety of multi-task models and settings. We ask:

RQ1: *Are multi-task models reliably more robust than single-task models?*

To answer this question, we study whether the results of Theorem 2 (i.e. adding tasks increases robustness) generalize to other settings. In particular, Theorem 2 was proven for l_2-normed attacks and we investigate if their results remain valid for l_∞-normed attacks. Finally, we investigate the impact of various experimental settings covering different perturbation budgets ϵ, architectures, and number of training steps.

Following this, we formulate an alternate hypothesis that could explain the apparent robustness of multi-task models and the evidence brought by our study: what matters most is not the number of tasks or how they correlate but how much the tasks individually impact the vulnerability of the model. Thus, making more robust a model with one task that is "marginally more vulnerable" requires adding robust tasks that make this model "marginally less vulnerable".

Our second research question studies this hypothesis and attempts to quantify this *marginal vulnerability* :

RQ2: *How to quantify the individual contribution of each task on the robustness of the model?*

To answer this question, we define the concept of marginal adversarial vulnerability of a model to a task i as the variation between the adversarial vulnerability of the model with this newly added task i and its vulnerability before this task.

Based on this concept of marginal adversarial vulnerability, we look for ways to improve robustness. Past research on multi-task learning suggests that carefully weighing tasks can drastically improve clean performance (Vandenhende et al. 2021). Leaning on this idea, we hypothesize that one can improve model robustness by adjusting the weights of the tasks. We ask:

RQ3: *Can one improve the robustness of multi-tasks models by adjusting the weights of the tasks?*

We show that optimizing the weights significantly improves the robustness of multi-tasks models against PGD. However, this apparent robustness may actually result from a gradient masking effect (Athalye, Carlini, and Wagner 2018) caused by the weight adjustment. We, therefore, investigate if adaptive attacks – which are known to circumvent gradient masking – can successfully attack the weight-optimized model. We use Auto-PGD (Croce and Hein 2020) and propose a new attack that adjusts at each step of the attack which tasks are attacked and how much the gradient of each task is penalized to compute the optimal perturbation.

Finally, we investigate the practical question of how to identify the combinations of tasks that yield the highest robustness. Given two multi-task models, we propose a set of guidelines that help practitioners to infer the robustness of the task combinations of each model from cheaper models. Our final question is:

RQ4: *How to efficiently find combinations of tasks giving the best robustness?*

Experimental Setup

The following describes our general experimental setup used across all RQs. It must be noted that the setups specific to the RQs are presented in their dedicated sections.

Dataset. We use the Taskonomy dataset, an established dataset for multi-task learning (Zamir et al. 2018). From the original paper, we focus on 11 tasks : Semantic Segmentation (s), Depth z-buffer Estimation (d), Depth euclidian Estimation (D), Surface Normal Prediction (n), SURF Keypoint Detection in 2D (k) and 3D (K), Canny Edge Detection (e), Edge Occlusion (E), Principal Curvature (p), Reshading (r) and Auto-Encoders (A). This subset of tasks is at the intersection of the major studies (Zamir et al. 2018; Vandenhende et al. 2019; Standley et al. 2020) about tasks similarity.

Attacks. We focus our research on gradient-based attacks. In particular, we use as base setting the l_∞ Projected Gradient Descent attack (PGD) (Madry et al. 2017) with 25 steps attacks, a strength of $\epsilon = 8/255$ and a step size $\alpha = 2/255$.

We also study in RQ3 the impact of adaptive attacks on the robustness of multi-task models. We evaluate the robustness of weighted multi-task models against Auto-PGD (Croce and Hein 2020), the strongest parameter-free gradient attack.

Finally, we design a new attack, *WGD*, that takes into account individual task weights and show that multi-task learning is vulnerable against our adaptive attack.

Models. We use the architectures and training settings of the original Taskonomy paper (Zamir et al. 2018): A Resnet18 encoder and a custom decoder for each task.We use a uniform weights, Cross-entropy loss for the semantic segmentation task and an L1 loss for the other tasks.

Our evaluation covers all possible combinations of tasks, i.e 935 multi-task models. Each model is a combination of a main task and one or multiple auxiliary tasks. We present in our evaluation the case where the attacker aims to attack only the main task (single-task attacks) and when all the tasks are attacked simultaneously (multi-task attacks).

We provide in the Appendix B. of supplementary material the detailed setup of each setting and the detailed results.

Robustness Metrics

Task robustness The common way to evaluate empirically the adversarial robustness of a DNN is to compute the success rate of the attack, i.e. the percentage of inputs for which the attack can produce a successful adversarial example under a constrained perturbation budget ϵ.

While this metric is suited for classification tasks that rely on classification accuracy, most dense tasks rely on metrics where a success rate is hard to define objectively or requires a hand-picked threshold. For instance, "image segmentation" uses Intersection Over Union (IoU, between 0 and 1) as a metric, while "pose estimation" relies on the number of correct keypoints and their orientation, and "depth estimation" uses the mean square error. To account for this diversity of metrics we define a generic metric that reflects how much the performance has degraded (how much relative error) after an attack: *the relative task vulnerability* metric.

Relative task vulnerability Given a model $\mathscr{M}$, we define the relative task vulnerability v_i of a task i as the average relative error increase when changing clean inputs $x^{(k)}$ into adversarial inputs $x^{(k)} + \delta$, given their associated ground truth $y_i^{(k)}$. Hence, v_i is given by:

$$v_i = \mathbb{E}_k \left[\frac{f_i(x^{(k)} + \delta, y_i^{(k)}) - f_i(x^{(k)}, y_i^{(k)})}{f_i(x^{(k)}, y_i^{(k)})} \right]$$

where error function $f_i(x^{(k)}, y_i^{(k)})$ is a task-specific error between the ground truth $y_i^{(k)}$ and the predicted value of $x^{(k)}$ (e.g., MSE, 1-IoU, etc.).

The concept of relative task vulnerability enables the comparison of two models in terms of the robustness they achieve for any of their task(s). A model with a smaller value of v_i indicates that it is more robust to an attack against task i.

RQ1: Adding Auxiliary Tasks

Theoreom 2 showed that adversarial vulnerability decreases with the number of uncorrelated tasks. We argue that the results do not generalize to different settings (norms, attack strength, ...) and investigate the other factors which may affect the robustness of the models. This allows us to identify the key factors confirming or refuting the results of Theorem3. More precisely, we consider the attack norm p, the perturbation budget ϵ, the model architecture, and the number of training steps (the convergence of learning). We summarize our findings below while detailed results are in Appendix C.

Attack norm p. We evaluate the relative task vulnerability against single-task and multi-task attacks with a limited perturbation budget (ϵ = 8/255, 25 attack steps) under l_2 and l_∞ attacks. Under l_2 attacks, the previous conclusions of (Mao et al. 2020) are confirmed. Under l_∞ attacks, however, adding auxiliary tasks does not reliably increase robustness against neither single-task nor multi-task adversarial attacks. We indeed observe for each task t that the single-task model of t is not more vulnerable than multi-task models with t as the main task – regardless of the fact that a single-task attack or a multi-task attack was used (see, e.g., also Table 1).

Attack budget ϵ. We evaluate the robustness of multitask models against attack size $\epsilon \in \{4/255, 8/255, 16/255, 32/255\}$ Under strong adversarial attacks ($\epsilon > 4/255$), multi-task learning does not provide reliable robustness both against single-task and multi-task adversarial attacks.

Model architecture. We evaluate the vulnerability of multi-task models on three families of encoders (Xception, Wide-Resnet, and Resnet) and for the latter, three sizes of encoders (Resnet18, Resnet50 and Resnet152). We train each architecture on a pair of tasks from s, d, D, E, n. We evaluate the robustness of each combination of tasks and compare it with the robustness of the same architecture trained using only one of the tasks. For all architectures, multi-task models are not reliably less vulnerable than single-task models. On the contrary, when one task is targeted, 80% of the multi-task models using Resnet50 and Resnet152 architectures are more vulnerable than their single-task counterparts.

Answer to RQ1: For large perturbation budgets ϵ, l_∞ norms, or large models, multi-task learning does not reliably improve the robustness against adversarial attacks.

RQ2: Marginal Adversarial Vulnerability

Theoretical Analysis

To better understand the impact of additional tasks on the multi-task vulnerability, we define the concept of marginal adversarial vulnerability of tasks, and propose a new theorem that bounds the contribution of the additional tasks to the model's vulnerability.

Definition 3. *Let $\mathscr{M}$ be a multi-task model with $\mathscr{T} = \{t_1, ..., t_M\}$ tasks, an input x, $\bar{y} = (y_1, ..., y_M)$ its corresponding ground-truth. We denote the set of attacked tasks $\mathscr{T}_N$ and $\mathscr{T}_{N+1}$, two subsets of the model's tasks $\mathscr{T}$ such as $\mathscr{T}_{N+1} = \mathscr{T}_N \cup \{t_{N+1}\}$ and $N + 1 \leq M$, and let $\mathcal{L}'$ be the joint task loss of attacked tasks.*

We define marginal adversarial vulnerability *of the model to an ϵ-sized $\|.\|_p$-attack as the difference between the adversarial vulnerability over the task set $\mathscr{T}_{N+1}$ and the adversarial vulnerability over the task set $\mathscr{T}_N$. It is given by:*

$$\Delta_N \mathbb{E}_x[\delta\mathcal{L}'] = \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})] - \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)]\big]$$

Similarly to the adversarial vulnerability of a model, for a small δ value we propose to use Taylor expansions to approximate the marginal vulnerability of the model to a given task. We propose the following theory for the marginal change of the vulnerability of a model when we add a task:

Theorem 4. *For a given multi-task model $\mathscr{M}$, let $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, with a weight w_i and zero mean such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x \mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathbf{r}_i$. The first-order approximation of the marginal vulnerability is bounded as follow:*

$$\widetilde{\Delta_N \mathbb{E}_x[\delta\mathcal{L}']} \leq \epsilon \cdot ((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + N \cdot \max_{i<N+1} w_i \mathbb{E}_x[|| \mathbf{r}_i ||])$$

Proof. Appendix A.4 and A.5 □

When all the tasks have the same weight, $w_i = \frac{1}{N}$ and $w_{N+1} = \frac{1}{N+1}$ and we have:

$$\widetilde{\Delta_N \mathbb{E}_x[\delta\mathcal{L}']} \leq \epsilon \cdot (\mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + \max_{i<N+1} \mathbb{E}_x[|| \mathbf{r}_i ||])$$

This theorem shows that the increase in adversarial vulnerability when we add more tasks does not depend on the number of tasks already attacked but relates to how robust is the new task we are adding and how weak is the most vulnerable task of the model.

Empirical Evaluation

To confirm our hypothesis that increasing the number of tasks does not guarantee the increase of robustness, we empirically evaluate how the adversarial vulnerability of a model changes when successively adding more tasks (up to 5 tasks). The model is trained with all 5 tasks then we successively enable the tasks. We show below the results for four combinations. Other combinations of tasks are provided in Appendix C.

Figure 1 shows the results. We observe that increasing the number of tasks often increases the vulnerability of the model. This confirms again that the main claim of (Mao et al. 2020) does not generalize to any combinations of tasks. We also observe that there is no monotonic relationship between the model vulnerability and its number of tasks for cases (2) and (3), whereas in cases (1) and (4) the increase is not linear and mostly occurs at one specific point (i.e., when the segmentation task s is added). More generally, across all four

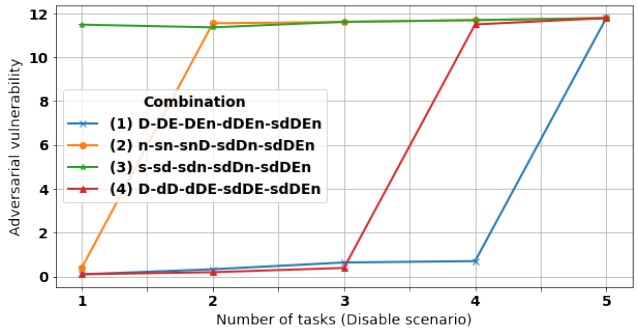

Figure 1: Adversarial vulnerability for 4 different combinations of tasks. In each combination, we enable one additional task and report the exact adversarial vulnerability of the new model. Evaluated tasks: *s*: Semantic segmentation, *d*: Z-depth, *D*: Euclidian depth, *n*: Normal estimation, *E*:Edge detection.

cases, task s appears to be the main factor contributing to the increased vulnerability of the model. This supports our claim that the most marginally vulnerable tasks are the dominant factors to increasing the model vulnerability.

Answer to RQ2: The marginal vulnerability increase of the model mainly depends on the vulnerability of the newly added task and the most vulnerable previous task. This implies that, (1) the more vulnerable the tasks in the model are, the less likely adding new tasks increases the robustness of the model; and (2) adding a vulnerable task may actually decrease the robustness of the whole model.

RQ3: Task Weight Optimization

We evaluate how the optimization of weights of the losses of each task can be used as a defense, through the selection of optimal weights. We investigate and as an adaptive attack to overcome the gradient masking of multi-task learning.

Robustification Through Optimal Weights

Given that simply adding tasks is not a promise of increased robustness, we suggest that a better way would be to adjust the weights of the tasks (in the loss function). The problem of setting the task weights has been previously studied in the context of optimizing the clean performance of multi-task models (Chen et al. 2018; Vandenhende et al. 2021; Standley et al. 2020). Specifically, Zamir et al. (Zamir et al. 2018) provided the optimal weights for all combinations of tasks of their dataset (see Appendix B.).

To evaluate the potential benefits of adjusting the weights, we conduct an empirical evaluation and compare the vulnerability of models using equal task weights (the *Baseline* models) with the equivalent model using the optimized task weights suggested by (Zamir et al. 2018) (the *Weighted* models). More precisely, we consider any pair of tasks where the first task is the main task and the second is the auxiliary task (added for the specific purpose of making the model less vulnerable on the main task). To compare the equal-weight

Attack		Baseline (A)					Weighted (B)				
Auxiliary →		s	d	D	n	E	s	d	D	n	E
Single	s	*0.82*	0.86	0.97	0.96	0.93	-	0.52	0.57	0.58	0.54
	d	5.74	*5.61*	5.28	6.88	6.41	1.25	-	2.00	1.65	1.92
	D	5.93	6.14	*6.4*	7.12	8.31	2.16	1.88	-	2.11	1.78
	n	7.43	9.48	8.93	*10.82*	9.08	6.61	9.46	8.09	-	8.14
	E	12.93	19.29	18.44	15.16	*22.57*	6.44	5.50	8.02	5.49	-
Multi	s	-	0.85	0.96	0.95	0.91	-	0.45	0.49	0.48	0.43
	d	1.99	-	5.42	4.8	4.46	0.56	-	2.06	0.82	0.75
	D	2.14	6.02	-	5.02	6.07	0.88	1.74	-	0.94	0.68
	n	4.61	9.4	8.93	-	8.7	3.65	9.22	7.99	-	6.32
	E	7.58	18.5	17.44	12.48	-	3.24	4.93	7.00	3.36	-

Table 1: Relative task vulnerability (lower is better). (A): adversarial attack against uniformly weighted tasks. (B): Adversarial attack against weighted tasks. Each row is the main task evaluated and the column is the auxiliary task. In the top half (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

model with the optimized-weight model, we use the relative task vulnerability metrics as this metric is independent of the task weights. We use both single-task PGD (Madry et al. 2017) and multi-task PGD (Mao et al. 2020).

Results are shown in Table 1 (Baseline A vs Weighted B). We see that for each pair of tasks, the weighted model is less vulnerable than the baseline model. This confirms our hypothesis that a careful setting of the weights can reduce model vulnerability, even where the addition of tasks with equal weights has a negative effect. For instance, in the baseline models, we observe that the vulnerability of the model on task s is lower when s is the only task than when any other task is added. When weights are optimized though, the vulnerability of the weighted multi-task model on s is always lower than s alone *regardless of the task that is added*. These results can be explained by the fact that the weight optimization proposed in (Zamir et al. 2018) aims to reduce the influence of dominant tasks during learning. As the two attacks inherently target the most dominant tasks (which have a higher average contribution to the loss function), the weight optimization improves both clean performance *and* robustness.

Adaptive Gradient Attacks

We investigate whether weight optimization remains an effective defense against adaptive attacks. We consider Auto-PGD (Croce and Hein 2020) – the strongest adaptive gradient attack in the literature – which adjust the step of the attack and the weight of the momentum at each attack iteration.

We also propose another way to make an attack adaptive. The principle of our new attack is to weight the contribution of each task when computing the perturbation to guarantee that the attack affects all targeted tasks – including those that have a smaller contribution to the joint loss.

We introduce the concept of Task Attack Rate to optimally weigh the gradient of each attacked task. Task Attack Rate is inspired by the inverse task learning rate proposed by (Chen et al. 2018) for the GradNorm optimization for training.

Definition 5. *We define the* Inverse task attack rate *of the task i under an ϵ-sized $\|.\|_p$- iterative attack on $\mathscr{T}'$ at step t the loss ratio for a specific task i at step t: $\tilde{\mathcal{L}}_i(t) = \frac{\mathcal{L}_i(t)}{\mathcal{L}_i(0)}$*

Smaller $\tilde{\mathcal{L}}_i$ implies that task i is faster to perturb. Similarly, we define the relative inverse attack rate as $r_i(t) = \frac{\tilde{\mathcal{L}}_i(t)}{\mathbb{E}_i[\tilde{\mathcal{L}}_i(t)]}$

We leverage this optimization in our new attack, *Multi-task Weighted Gradient Attack (WGD)*, a multi-step attack where the gradient of each task is weighted by the relative inverse task attack rate. We describe full algorithm in Appendix D.

We empirically assess whether the two adaptive attacks can overcome the robustification mechanism based on optimal weights. Hence, we measure the relative task vulnerability of the baseline (uniformly weighted) model and the optimally weighted model against WGD and Auto-PGD (Table 2).

We observe that, on both models, WGD and Auto-PGD are much stronger than PGD (compared to Table 1). For instance, Auto-PGD increased the error against task *E* up to five times in comparison with PGD on the same combination of tasks, and WGD caused up to two times more error than PGD for the combination of tasks supporting *n*.

Table 2 also reveals that the weighted model is as vulnerable as the baseline model. This confirms that the adaptive attacks negate the benevolent effects of weight optimization.

In the end, the only viable way to improve model robustness remains to add less vulnerable auxiliary tasks. Indeed, in Table 2 we observe that for each single-task model (i.e. the diagonal elements) there is at least one multi-task model (with the same main task) that is less vulnerable. Our previous (RQ2) conclusion remains, therefore, valid.

Answer to RQ3: Weight optimization in multi-task learning decreases model vulnerability against non-adaptive attacks only. The only way to improve the robustness of multi-task models remains to add less vulnerable auxiliary tasks.

RQ4: Task Selection

Our previous results imply that one should carefully select the auxiliary tasks added to reduce model vulnerability. Generally speaking, the addition of auxiliary tasks can even have negative effects. Auxiliary task selection, however, comes with three drawbacks (Standley et al. 2020): the size of the model is bigger (due to the addition of the task-specific decoder), the convergence of the common encoder layers is

Attack		Baseline (A)					Weighted (B)				
Auxiliary →		s	d	D	n	E	s	d	D	n	E
APGD	s	0.89	0.91	0.90	0.88	0.92	0.89	0.92	0.93	0.88	0.90
	d	17.17	23.88	13.50	24.10	24.98	18.27	23.19	13.08	15.92	23.9
	D	15.50	15.08	20.15	26.00	22.74	20.72	24.93	18.29	28.21	23.68
	n	12.99	17.76	17.27	19.02	17.24	12.35	17.14	16.72	18.49	16.46
	E	135.4	171.8	159.7	138.8	81.77	125.8	78.65	377.8	110.4	68.06
WGD	s	0.90	0.91	0.91	0.90	0.91	0.90	0.93	0.94	0.93	0.91
	d	12.86	13.57	12.39	16.18	18.13	12.66	13.55	12.8	11.7	12.96
	D	14.05	14.04	15.57	17.03	19.24	15.65	14.85	15.55	16.95	13.56
	n	13.05	17.06	16.32	18.12	16.57	17.00	20.26	17.32	18.13	17.68
	E	45.35	90.67	86.04	57.43	90.19	106.9	89.89	116.3	71.42	90.59

Table 2: Relative task vulnerability under two different attacks (lower is more robust). (A): adversarial attack against uniformly weighted tasks. (B): Adversarial attack against optimally weighted tasks. Each row refers to the task attacked and evaluated and the column the auxiliary task.

slower, and the clean performance risk deteriorating as more tasks are added. This raises the question of how to select the combination that yields the lowest vulnerability *without* evaluating the vulnerability of all the possible combinations.

We propose three methods to make this selection more efficient. Their common idea is to compute a proxy of the adversarial vulnerability which is fast to get and correlated to the real adversarial vulnerability. We use the relative task vulnerability of the models on the main task, as this metric is independent of the task (unlike, the task loss whose scale depends on the task).

We use the Pearson coefficient to measure correlation between variables. We compute the correlation between the relative task vulnerability of one combination of tasks in the expensive model with the relative task vulnerability of the same combination on the cheaper surrogate.

The first method is *early stopping*, that is, training the model after a predefined (small) number of epochs. Here, we stop after 50 epochs while the full training lasts for 150 epochs. Strong correlations between the vulnerability of the models would indicate that one can decide which task combination is optimal after few epochs.

The second is to use a surrogate (less expensive) encoder and evaluate all task combinations on this encoder. We hypothesize that combinations of tasks that are effective when joint to the surrogate encoder are also effective with the original model. We use ResNet18 as the surrogate encoder and ResNet50 as the target encoder.

Our third selection method is guided by the clean performance on the main task when the auxiliary tasks are added. The existence of a correlation between clean performance and vulnerability would allow avoiding the cost of evaluating adversarial vulnerability (e.g. applying PGD) and reuse existing results on clean performance predictions (Standley et al. 2020) to predict adversarial vulnerability.

Table 3 shows the Pearson correlation coefficient with the associated p-value. We observe that all three proxy methods are correlated with the real adversarial vulnerability values. Specifically, early stopping offers a medium correlation (0.55) while the methods based on the surrogate encoder and the clean performance achieve a very strong correlation (0.94).

Answer to RQ4: While exhaustively computing the adversarial vulnerability for all task combinations is computationally expensive, guiding the auxiliary task selection by the clean performance or the vulnerability of a smaller surrogate model offers cheap and reliable indications of the benefits achieved by adding these tasks.

Proxy	Target	Pearson	p-value
50 epochs	150 epochs	0.55	4.23e-3
Resnet18	Resnet50	0.94	1.42e-12
Clean performance	Robustness	0.98	1.91e-17

Table 3: Pearson correlation between the real adversarial vulnerabilities and proxy values from three different methods.

Conclusion

We have presented what is to date the largest evaluation of the vulnerability of multi-task models to adversarial attacks. Our study does not entirely reject the benefits of adding auxiliary tasks to improve robustness, but rather tones down the generality of this proposition.

We evaluate different settings of multi-task learning, with a large combination of tasks, architectures, attack strengths and norms and show that in multiple settings, multi-task learning fails to protect against gradient attacks.

We also demonstrate that weight optimization can significantly improve the robustness of multi-task models, however, falls short to protecting against adaptive attacks for some tasks. In particular, we propose a new adaptive attack, WGD, that balances the gradient of the tasks and overcomes the gradient masking defense of multi-task learning.

Taking the perspective of the defender, we show that one can identify the most robust combinations of tasks efficiently by working on cheap surrogates.

Overall, our research contributes to guiding practitioners in the development of robust multi-task models and paves the way for methods to improve together the clean performance and the robustness of multi-task models.

Acknowledgments

This work is mainly supported by the Luxembourg National Research Funds (FNR) through CORE project C18/IS/12669767/ STELLAR/LeTraon

References

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. arXiv:1802.00420.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8190 LNAI, 387–402. ISBN 9783642409936.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. arXiv:1711.02257.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. PMLR.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Dyrmishi, S.; Ghamizi, S.; Simonetto, T.; Traon, Y. L.; and Cordy, M. 2022. On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks. arXiv:2202.03277.

Ghamizi, S.; Cordy, M.; Gubri, M.; Papadakis, M.; Boystov, A.; Le Traon, Y.; and Goujon, A. 2020. Search-Based Adversarial Testing and Improvement of Constrained Credit Scoring Systems. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, 1089–1100. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370431.

Ghamizi, S.; Cordy, M.; Papadakis, M.; and Traon, Y. L. 2019. Adversarial Embedding: A robust and elusive Steganography and Watermarking technique. *arXiv preprint arXiv:1912.01487*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014a. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014b. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, W.; Wei, J.; Chen, X.; Carlini, N.; and Song, D. 2017. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} workshop on offensive technologies ({WOOT} 17)*.

Inkawhich, N.; Wen, W.; Li, H. H.; and Chen, Y. 2019. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7066–7074.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mao, C.; Gupta, A.; Nitin, V.; Ray, B.; Song, S.; Yang, J.; and Vondrick, C. 2020. Multitask Learning Strengthens Adversarial Robustness. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, 158–174. Springer.

Metzen, J. H.; Genewein, T.; Fischer, V.; and Bischoff, B. 2017. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

Radwan, N.; Valada, A.; and Burgard, W. 2018. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4): 4407–4414.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9339–9347.

Sax, A.; Emi, B.; Zamir, A. R.; Guibas, L.; Savarese, S.; and Malik, J. 2018. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv:1812.11971*.

Simon-Gabriel, C.-J.; Ollivier, Y.; Bottou, L.; Schölkopf, B.; and Lopez-Paz, D. 2019. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, 5809–5817. PMLR.

Sinha, A.; Namkoong, H.; Volpi, R.; and Duchi, J. 2017. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, 9120–9132. PMLR.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tramèr, F.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*.

Vandenhende, S.; Georgoulis, S.; De Brabandere, B.; and Van Gool, L. 2019. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*.

Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; and Van Gool, L. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu, Q.; and Yang, Q. 2011. A survey of transfer and multitask learning in bioinformatics. *Journal of Computing Science and Engineering*, 5(3): 257–268.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zamir, A. R.; Sax, A.; Shen, W. B.; Guibas, L. J.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling Task Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Zhang, Y.; and Yang, Q. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.