Adversarial Robustness in Multi-Task Learning: Promises and Illusions PAPER #2923

Anonymous

Sept 2021

Contents

1 Appendix A: Proofs for the Theoretical Analysis

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 call $\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)]$ the* adversarial vulnerability *of $\mathscr{M}$ on $\mathscr{T}'$ to an ϵ-sized $\|.\|_p$-attack.*

And we define it as the average increase of $\mathcal{L}_{\mathscr{T}'}$ after attack over the whole dataset, i.e.:

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

Lemma 2. *Under an ϵ-sized $\|.\|_p$-attack, the adversarial vulnerability of a multi-task model can be approximated through the first-order Taylor expansion, that is:*

$$\mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}')] \approx \epsilon \cdot \mathbb{E}_x[\,|| \,\partial_x\mathcal{L}'(x, \bar{y}) \,||_q] \tag{1}$$

Proof. **1.1 A.1**

From definition 1, we have:

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

Given the perturbation δ is minimal, we can approximate $\delta\mathcal{L}$ with a Taylor expansion up to a second order:

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] \quad \approx \quad \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \delta \cdot \partial_x\mathcal{L}'(x, \bar{y}) + \frac{\delta^2}{2} \cdot \partial_x^2\mathcal{L}'(x, \bar{y}) \mid\right]$$

The noise δ is optimally adjusted to the coordinates of $\partial_x\mathcal{L}'$ within an ϵ-constraint. By the definition of the dual-norm, we get:

$$\mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}')] \approx \mathbb{E}_x\left[\,|| \,\epsilon \cdot \partial_x\mathcal{L}'(x, \bar{y}) + \frac{\epsilon^2}{2} \cdot \partial_x^2\mathcal{L}'(x, \bar{y}) \,||_q\right] \tag{2}$$

where q is the dual norm of p and $\frac{1}{p} + \frac{1}{q} = 1$ and $1 \leq p \leq \infty$.
We obtain Lemma 2 by restricting the Taylor expansion to the first-order.

□

Theorem 3. *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*

$$\mathbb{E}_x[\delta\mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M}\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_j)}{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_i)}}{M}} \tag{3}$$

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. **1.2 A.2**

let $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, with a weight $w_i = \frac{1}{M}$ such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x \mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathbf{r}_i$. let $p = q = 2$

We have:

$$\begin{aligned}
\mathbb{E}_x[|| \epsilon \cdot \partial_x \mathcal{L}'(x, \bar{y}) ||_2^2] &= \mathbb{E}_x \left[|| \sum_{j=1}^{M} \frac{\epsilon}{M} \cdot r_i ||_2^2 \right] \\
&= \frac{\epsilon^2}{M^2} \mathbb{E}_x \left[\sum_{i=1}^{M} || \mathbf{r}_i ||_2^2 + 2 \sum_{i=1}^{M} \sum_{j=1}^{i-1} || \mathbf{r}_i ||_2 || \mathbf{r}_j ||_2 \right] \\
&= \frac{\epsilon^2}{M^2} \left(\sum_{i=1}^{M} \mathbb{E}_x[\mathbf{r}_i^2] + 2 \sum_{i=1}^{M} \sum_{j=1}^{i-1} \mathbb{E}_x[\mathbf{r}_i \mathbf{r}_j] \right)
\end{aligned} \tag{4}$$

We know:

$$\mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right) = \mathbb{E}_x\left[\mathbf{r}_i \mathbf{r}_j\right] - \mathbb{E}_x\left[\mathbf{r}_i\right] \mathbb{E}_x\left[\mathbf{r}_j\right] \tag{5}$$

According to the assumptions, the gradient of each task is i.i.d with zero means: $\mathbb{E}_x\left[\mathbf{r}_i\right] = 0$ Then $\mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right) = \mathbb{E}_x\left[\mathbf{r}_i \mathbf{r}_j\right]$ and $\sigma_i^2 = \mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right) = \mathbb{E}_x\left[\mathbf{r}_i^2\right]$.

$$\begin{aligned}
\mathbb{E}_x[|| \epsilon \cdot \partial_x \mathcal{L}'(x, \bar{y}) ||_2^2] &= \frac{\epsilon^2}{M^2} \sum_{i=1}^{M} \left(\sigma_i^2 + 2 \sum_{j=1}^{i-1} \mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right) \right) \\
&\propto \frac{1}{M} \left(1 + 2 \sum_{i=1}^{M} \sum_{j=1}^{i-1} \frac{\mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right)}{M \sigma_i^2} \right) \\
\mathbb{E}_x[|| \epsilon \cdot \partial_x \mathcal{L}'(x, \bar{y}) ||_2] &\propto \sqrt{\frac{\left(1 + 2 \sum_{i=1}^{M} \sum_{j=1}^{i-1} \frac{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_i)}{M \sigma_i^2} \right)}{M}}
\end{aligned} \tag{6}$$

Using the first order adversarial vulnerability (Lemma 2), we then have:

$$\mathbb{E}_x[\delta \mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M} \sum_{i=1}^{M} \sum_{j=1}^{i-1} \frac{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_j)}{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_i)}}{M}} \tag{7}$$

with K a constant dependant of ϵ and the attacked tasks.

□

Definition 4. *Let $\mathscr{M}$ be a multi-task model with $\mathscr{T}_M = \{t_1, ..., t_M\}$ 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 call marginal adversarial vulnerability *of the model to an $\mathscr{T}', \epsilon$-sized $\|.\|_p$-attack the difference between the adversarial vulnerability over the task set $\mathscr{T}_{N+1}$ and the adversarial vulnerability over the task set $\mathscr{T}_N$.*

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

Lemma 5. *Under an ϵ-sized $\|.\|_p$-attack, the marginal adversarial vulnerability of a multi-task model can be approximated through the first-order Taylor expansion, that is:*

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

Proof. **1.3 A.3**

From Definition 4, we have:

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

using the lemma 2 at the first order expansion on each term of the right side, we get:

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

□

Lemma 6. *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 such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x \mathcal{L}(x, \bar{y}) = \sum_{i=1}^M w_i \mathbf{r}_i$. Let $\|.\|_q$ be a norm and p an integer. We have:*

$$\mathbb{E}_x[||\sum_{i=1}^M w_i \mathbf{r}_i\,||_q^p] \leq \sum_{i=1}^M w_i^p \mathbb{E}_x\left[||\ \mathbf{r}_i\ ||_q^p\right] \tag{8}$$

Proof. **1.4 A.4**

$$\begin{aligned} \mathbb{E}_x[||\sum_{i=1}^M w_i \mathbf{r}_i\,||_q^p] &\leq \mathbb{E}_x\left[\sum_{i=1}^M ||\ w_i \mathbf{r}_i\ ||_q^p\right] \\ &\leq \sum_{i=1}^M w_i^p \mathbb{E}_x\left[||\ \mathbf{r}_i\ ||_q^p\right] \end{aligned} \tag{9}$$

□

This lemma provides an upper-bound of the average norm of the gradients that we use to evaluate the upper bounds of the adversarial vulnerability in the following theorem:

Theorem 7. *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 marginal vulnerability is bounded as follow:*

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

Proof. **1.5 A.5**

Using Lemma 5, we have:

$$\begin{aligned}\Delta_N \mathbb{E}_x[\delta\mathcal{L}'] &\approx \epsilon \cdot \mathbb{E}_x[\,||\,\partial_x\mathcal{L}'(x,\bar{y},\epsilon,\mathscr{T}_{N+1})\,||_q] - \epsilon \cdot \mathbb{E}_x[\,||\,\partial_x\mathcal{L}'(x,\bar{y},\epsilon,\mathscr{T}_N)\,||_q] \\ &\leq \epsilon\,(\,|\,\mathbb{E}_x[||\,\partial_x\mathcal{L}'(\mathscr{T}_{N+1})\,||_q]\,|\,+\,|\,\mathbb{E}_x[||\,\partial_x\mathcal{L}'(\mathscr{T}_N)\,||_q]\,|\,)\end{aligned} \tag{10}$$

We use lemma 6 with p=1 and N+1:

$$\begin{aligned}\mathbb{E}_x[||\sum_{i=1}^{N+1} w_i\mathbf{r}_i\,||_q] &\leq (N+1)\sum_{i=1}^{N+1} w_i\mathbb{E}_x\,[||\,\mathbf{r}_i\,||_q] \\ &\leq (N+1)\left(\sum_{i=1}^{N} w_i\mathbb{E}_x\,[||\,\mathbf{r}_i\,||_q] + w_{N+1}\mathbb{E}_x\,[||\,\mathbf{r}_{N+1}\,||_q]\right) \\ \leq N\sum_{i=1}^{N} w_i\mathbb{E}_x\,[||\,\mathbf{r}_i\,||_q] &+ \sum_{i=1}^{N} w_i\mathbb{E}_x\,[||\,\mathbf{r}_i\,||_q] + (N+1)\cdot w_{N+1}\mathbb{E}_x\,[||\,\mathbf{r}_{N+1}\,||_q]\end{aligned} \tag{11}$$

We use similarly lemma 6 with p=1 and N:

$$\mathbb{E}_x[||\sum_{i=1}^{N} w_i\mathbf{r}_i\,||_q] \leq N\sum_{i=1}^{N} w_i\mathbb{E}_x\,[||\,\mathbf{r}_i\,||_q] \tag{12}$$

We inject (11) and (12) in (10) and we have:

$$\Delta_N \mathbb{E}_x[\delta\mathcal{L}'] \leq \epsilon\left((N+1)\cdot w_{N+1}\mathbb{E}_x[||\,\mathbf{r}_{N+1}\,||] + \sum_{i=1}^{N} w_i\mathbb{E}_x[||\,\mathbf{r}_i\,||]\right)$$

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

□

2 Appendix B: Experimental Settings

General training We use the same learning rate schedule for all the models: SGD with learning rate 0.01 and momentum 0.99. We decrease the learning rate at 100 epoch by 10 times, then successively at epoch=120 and epoch=140, we decrease again by 10 times. We train all the models for 150 epochs.

We train on 80% of the rooms (9464 images from 1500 different rooms) and test on the remaining 20%.

Experimental Settings We train different combinations of encoders and task decoders: Resnet18, Resnet50, W-Resnet50, Resnet152 and Xception. This allows us to check that our hypothesis of the limited impact of multi-task learning to generalize across different families of architectures and sizes. Table 1 lists our different settings. We evaluate the cost of the models as number of FLOPS (Floating Points Operations) required for one image inference, while the size is the number of weights of the model. Each task is handled by a specific decoder. The decoders are 8 layers (Convolution Dense).

Setting	Encoder	Weighted	Tasks	#Models	# Epochs	Size	Cost
S1	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	150	14.19M	6.09B
S2	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	50	14.19M	6.09B
S3	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	100	14.19M	6.09B
S4	Resnet18	Optimal	(s,d,D,n,E)	25	150	14.19M	6.09B
S5	Resnet50	Uniform	(s,d,D,n,E)	25	150	29.66M	9.90B
S6	Xception	Uniform	(s,d,D,n,E)	25	150	4.33M	3.64B
S7	Resnet152	Uniform	(s,d,D,n,E)	25	150	64.30M	19.64B
S8	Wide-Resnet50	Uniform	(s,d,D,n,E)	25	150	72.99M	19.45B

Table 1: The experimental settings we evaluated.

List of weights for the weighted models For the weighted setting (S4), we use these weights. For each of the main tasks, we use 1 as a weight and the following for the auxiliary tasks weights:

- **Semantic segmentation (s)**: 0.01 for sd combination, 0.1 otherwise.
- **Z-Depth (d)**: 0.01 for dn combination, 0.1 otherwise.
- **Normal (n)**: 0.01 for nd combination, 0.1 otherwise.
- **Euclidian Depth (D)**: 0.1 for Ds combination, 0.01 otherwise.
- **Edge detection (E)**: 0.1 for all combinations.

3 Appendix C: Detailed evaluation of the settings and tasks

3.1 Relative task robustness of architectures

We provide in Tables 2 to 6 the clean performance of each task combination. We also provide the relative task vulnerability against single-task and multi-task attacks.

While the multi-task models are not reliably more robust than single-task models across all architectures, we see that robust combinations are similar across different models.

Auxiliary →		s	d	D	n	E
Clean	s	50.50	49.20	47.28	47.02	48.63
	d	101.08	97.50	98.48	93.94	100.53
	D	96.71	91.30	92.36	91.23	87.08
	n (e^{-2})	71.89	57.66	58.59	54.56	57.41
	E (e^{-2})	16.43	10.30	10.49	11.68	8.78
Single	s	0.81	0.84	0.97	0.96	0.93
	d	7.73	7.87	7.46	10.67	10.55
	D	7.78	8.92	8.44	10.95	11.78
	n	9.43	13.49	11.61	15.02	13.40
	E	16.32	26.85	25.43	26.62	31.37
Multi	s	0.81	0.81	0.94	0.92	0.90
	d	2.90	7.87	7.71	5.96	4.36
	D	3.01	8.91	8.44	5.83	5.61
	n	5.97	13.44	11.82	15.02	11.43
	E	10.02	24.95	24.15	18.62	31.37

Table 2: Relative task vulnerability (lower is better) for the **Resnet18** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	51.63	49.50	49.66	47.48	46.90
	d	106.57	98.28	95.92	95.35	94.38
	D	99.65	89.81	90.58	89.17	91.26
	n (e^{-2})	73.46	46.47	44.97	49.91	48.34
	E (e^{-2})	15.30	7.48	8.93	9.04	6.99
Single	s	0.82	0.90	0.90	0.97	0.99
	d	7.89	6.06	7.73	11.09	10.59
	D	3.13	2.19	3.02	4.46	1.76
	n	10.36	17.10	18.28	16.26	16.52
	E	22.67	29.75	22.11	31.07	28.41
Multi	s	0.82	0.88	0.89	0.94	0.96
	d	3.79	6.06	7.78	6.93	4.78
	D	1.57	2.16	3.02	2.27	0.89
	n	7.46	17.13	18.21	16.26	14.03
	E	14.00	28.26	21.95	22.54	28.41
-						

Table 3: Relative task vulnerability (lower is better) for the **Resnet50** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	42.91	42.14	43.19	44.35	44.17
	d	109.27	88.66	90.67	92.55	93.21
	D	103.55	98.43	85.70	93.51	88.58
	n (e^{-2})	5183.49	3932.54	4065.30	3803.32	4178.78
	E (e^{-2})	1817.37	140453.97	142222.13	2028.51	543.58
Single	s	1.21	1.26	1.20	1.11	1.14
	d	7.87	14.39	14.47	28.00	25.66
	D	8.18	19.31	12.88	7.33	23.42
	n	14.09	20.08	17.40	22.03	19.23
	E	18.03	0.15	0.14	22.44	193.38
Multi	s	1.21	1.23	1.16	1.07	1.09
	d	3.69	14.39	14.38	15.62	9.43
	D	3.58	17.66	12.88	3.91	9.18
	n	8.15	19.97	17.49	22.03	16.68
	E	11.38	0.10	0.10	22.75	193.38

Table 4: Relative task vulnerability (lower is better) for the **Xception** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	46.87	54.05	54.78	54.17	56.42
	d	116.91	180.59	403.70	108.81	114.66
	D	127.78	160.12	130.55	105.36	117.93
	n (e^{-2})	85.78	52.34	60.13	54.96	52.68
	E (e^{-2})	20.95	11.74	8.66	10.35	14.14
Single	s	1.01	0.75	0.74	0.74	0.67
	d	9.88	9.95	3.35	12.61	6.71
	D	5.91	9.65	11.65	9.93	12.70
	n	9.14	15.62	13.26	17.94	15.10
	E	11.95	20.23	30.85	25.41	17.81
Multi	s	1.01	0.74	0.73	0.72	0.65
	d	3.76	9.95	3.19	10.64	3.13
	D	2.25	8.54	11.65	5.67	6.49
	n	6.29	15.11	13.07	17.94	13.58
	E	8.16	19.67	32.11	17.73	17.81

Table 5: Relative task vulnerability (lower is better) for the **WideResnet50** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	46.15	45.17	50.48	51.01	50.86
	d	109.27	98.27	107.10	95.34	171.99
	D	103.55	92.58	96.60	93.51	90.10
	n (e^{-2})	110.84	48.20	56.71	59.22	116.94
	E (e^{-2})	18.17	18.20	6.97	20.29	10.52
Single	s	1.03	1.07	0.87	0.82	0.87
	d	7.87	6.87	6.93	9.03	2.65
	D	8.18	7.45	7.19	7.33	8.03
	n	11.10	14.29	13.51	10.87	22.06
	E	18.03	54.91	16.02	22.44	9.41
Multi	s	1.03	1.05	0.85	0.80	0.85
	d	3.69	6.87	7.18	5.60	1.83
	D	3.58	7.30	7.19	3.91	3.44
	n	8.64	14.35	13.43	10.87	21.94
	E	11.38	51.44	15.80	22.75	9.41

Table 6: Relative task vulnerability (lower is better) for the **Resnet152 models**. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

3.2 Performance over all 11 tasks

Figures 1-11 show the performance of the Resnet18 models after attack (25-steps PGD l_∞ with $\epsilon = 8/255; \alpha = 2/255$).

Across all 11 tasks, the multi-task models are not more robust than their single-task counterparts. Some are, while most are not.

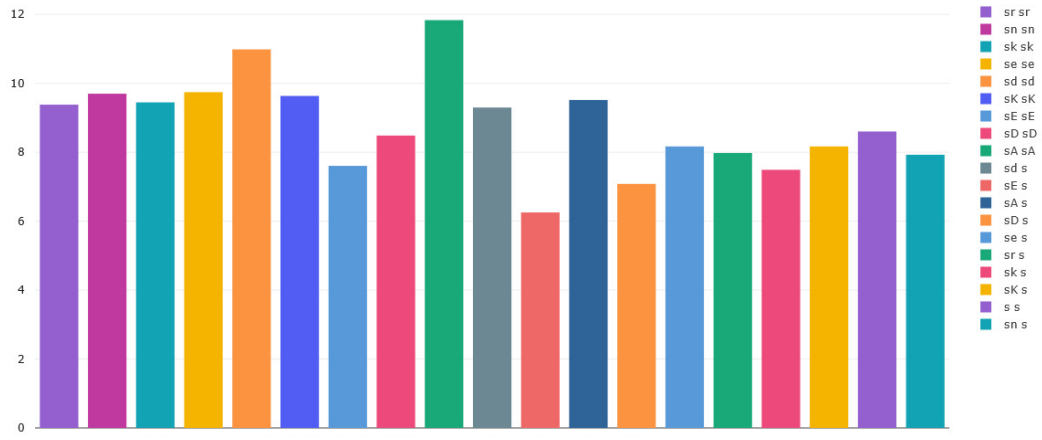


Figure 1: mIoU Semantic Segmentation (s) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. For instance "sn s" means model trained on both tasks s and n but only task s attacked. "s s" is the single-task baseline.

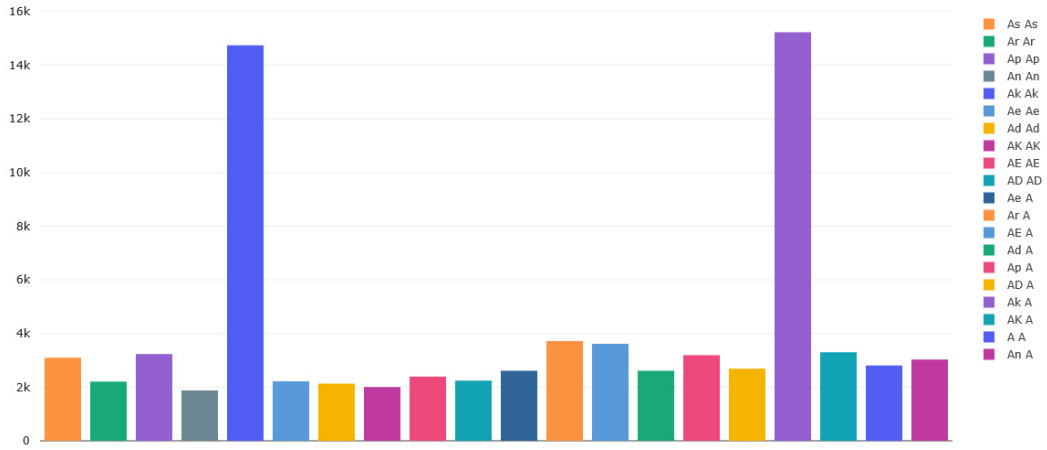


Figure 2: MSE of the Auto-encoder task (A) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "A A" is the single-task baseline.

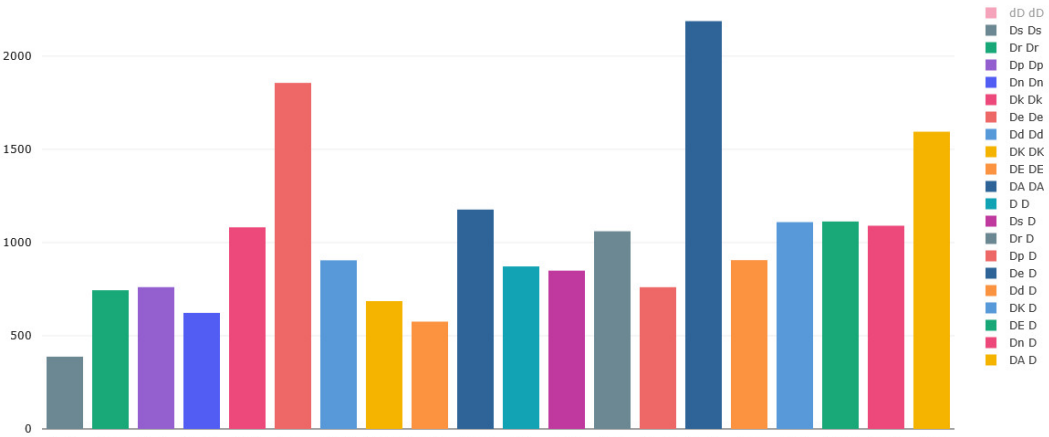


Figure 3: MSE of the Euclidian Depth (D) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "D D" is the single-task baseline.

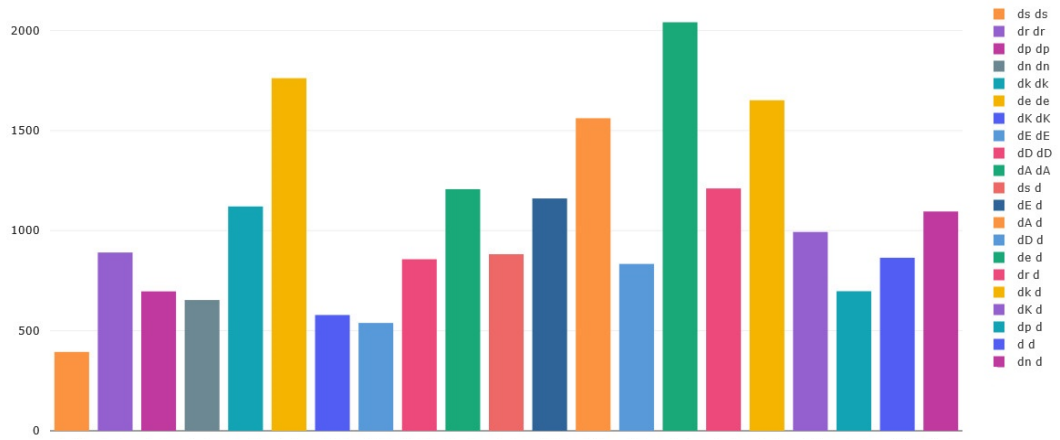


Figure 4: MSE of the Z-Depth (d) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "d d" is the single-task baseline.

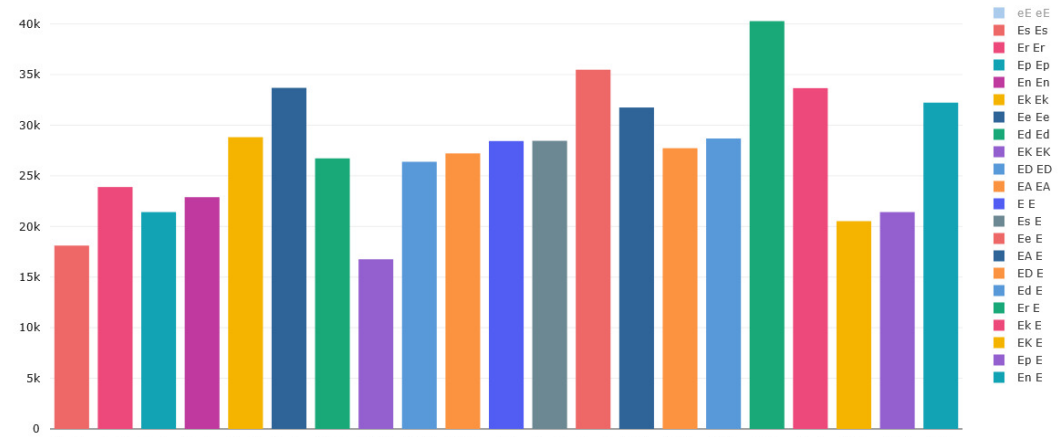


Figure 5: MSE of the Edge Occlusion (E) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "E E" is the single-task baseline.

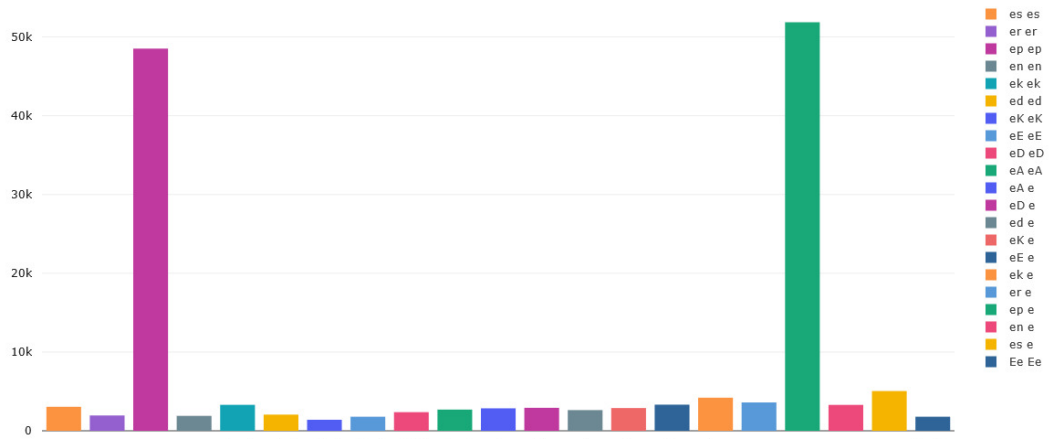


Figure 6: MSE of the Edge Texture (e) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "e e" is the single-task baseline.

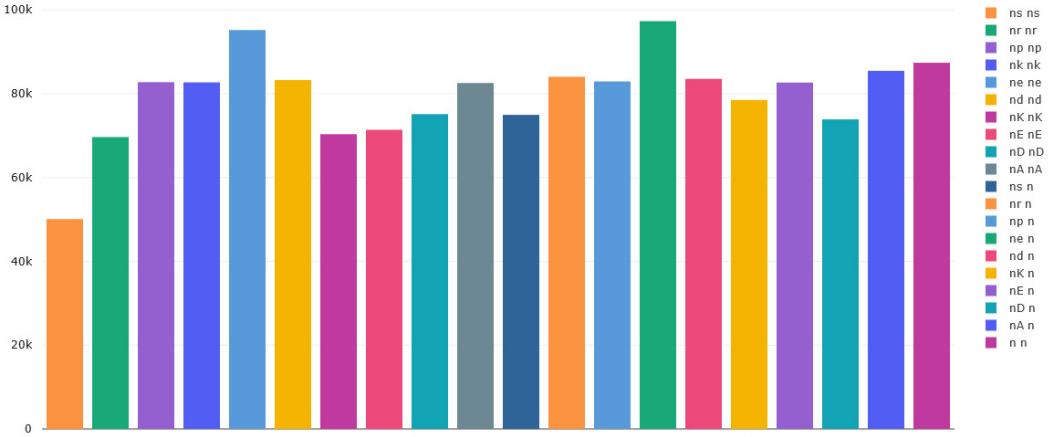


Figure 7: MSE of the Edge Normal estimation (n) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "n n" is the single-task baseline.

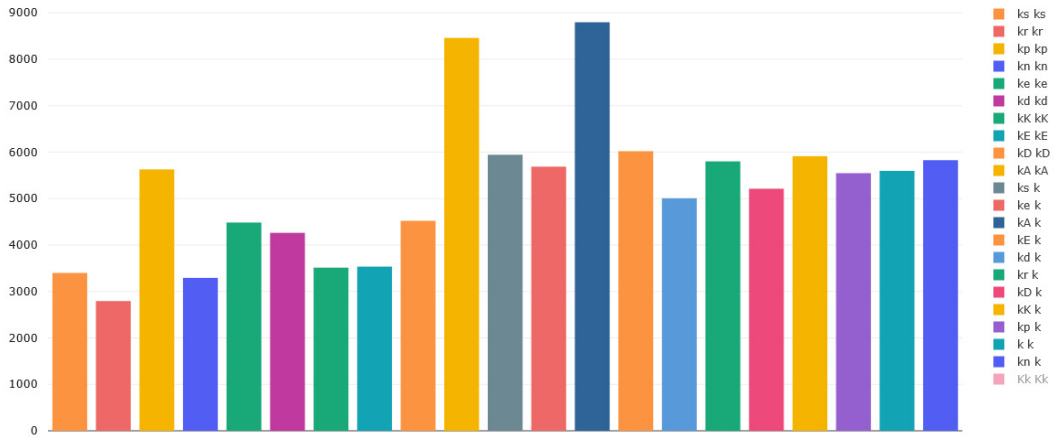


Figure 8: MSE of the Keypoints 2d (k) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "k k" is the single-task baseline.

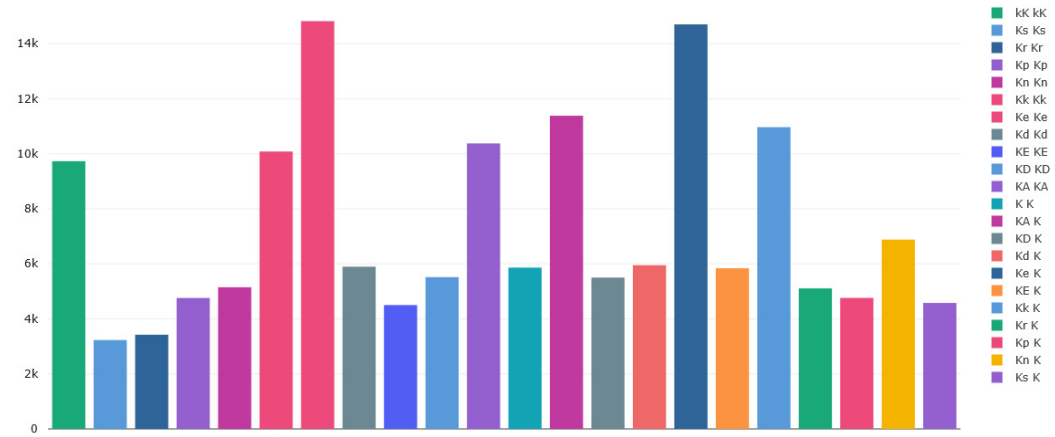


Figure 9: MSE of the Keypoints 3d (K) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "K K" is the single-task baseline.

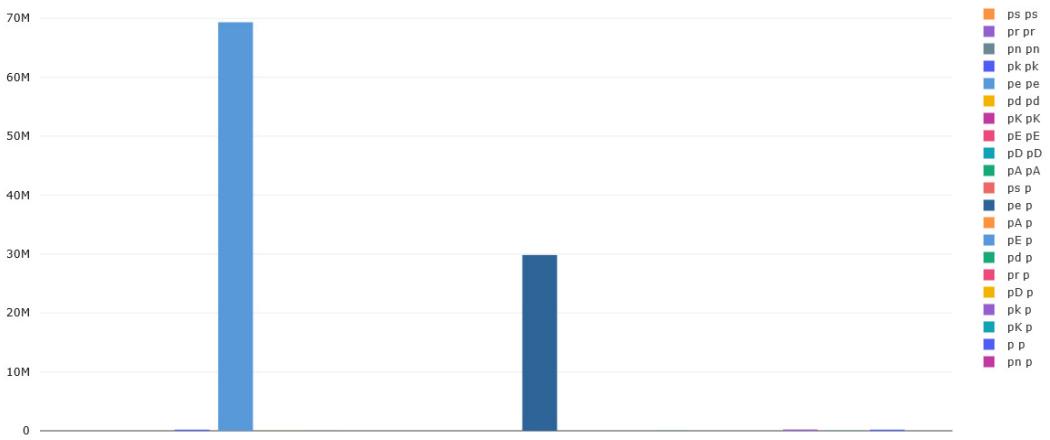


Figure 10: MSE of the Principal curvature (p) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "p p" is the single-task baseline.

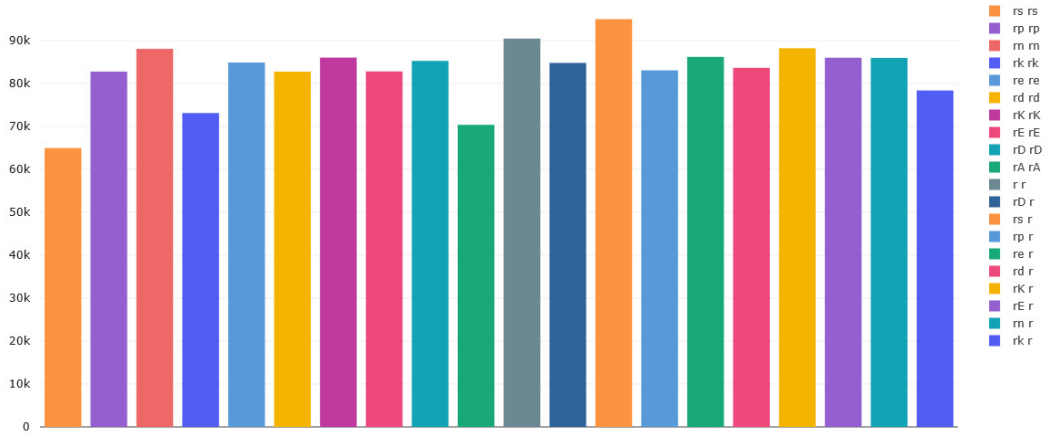


Figure 11: MSE of the Reshading (r) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. "r r" is the single-task baseline.

3.3 Adversarial Vulnerability and number of tasks

In addition to the main paper Figure 1, we evaluate the scenario where the tasks are added successively and the whole model is trained.

Figures 12 and 13 show how adversarial vulnerability changes when adding additional tasks. When the tasks are not weighted, the additional tasks do not improve the robustness of the models. When the tasks are weighted however, we can see that except when adding vulnerable tasks (s or E in our examples), the models vulnerability tends to decrease when adding supplementary tasks.

These results confirm our main claims that the number of tasks is not the main factor of the vulnerability of multi-task models but how we choose the tasks and how we weigh them.

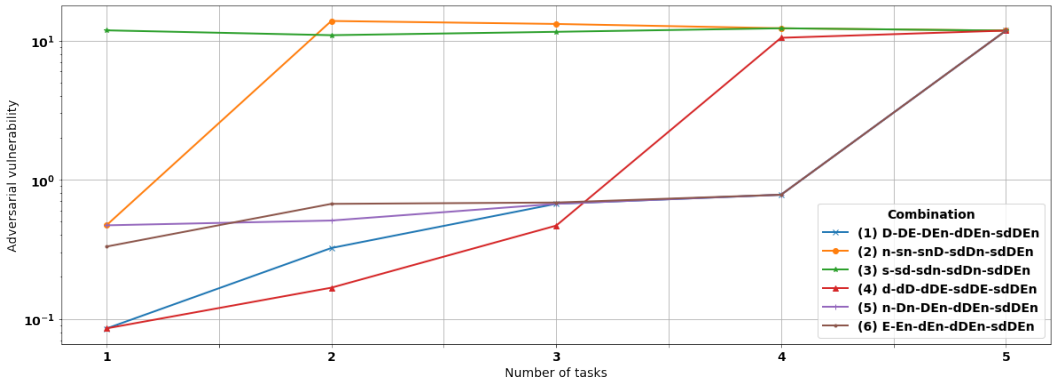


Figure 12: Adversarial Vulnerability when adding consecutive tasks. The tasks are not weighted.

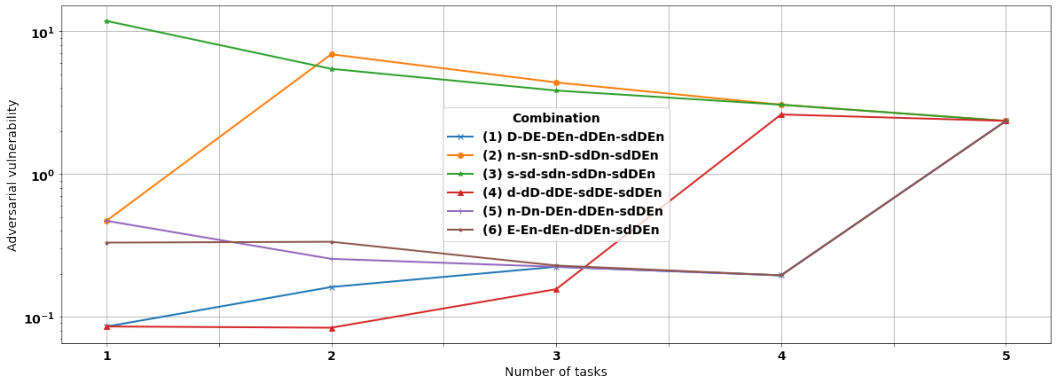


Figure 13: Adversarial Vulnerability when adding consecutive tasks. The tasks are weighted (1/N).

3.4 Impact of attack settings

Impact of number of steps Figures in Fig 15 show the impact of iteration steps on the robustness of the different tasks. The first finding is that while the multi-task models and the mono-task models display similar robustness on one-step and few step attacks, the differences across the models widens as we increase the number of steps (nd attack on a nd model causes a 75% increase in the MSE of z-depth in comparison with a d attack on the same nd model (figure 14c).

Especially, some combinations of tasks are more sensitive to the number of iterations. While Ds attack on a Ds model plateau after 15 epochs, D attack on the same Ds model keeps increasing significantly with the number of steps (figure 14b). Similarly, Es attack on an Es model plateau after 10 steps, while E attack on the same Es model keeps increasing (figure 14d). In general, against the same multi-task model, multi-task attacks tend to plateau much earlier,

Against mono-task attacks, Mono-task models are neither the more robust or the less across the different tasks. For instance training a model with the two tasks d and D makes the model 12% more robust than mono-task D (figure 14b, however training the model on the tasks n and D makes the model 48% less robust than the mono-task model D.

Similar behaviour happens against multi-task attacks. While it is easier to attack a single task model (task D) than attacking 2 tasks together, other combination of tasks are easier to attack than attacking one single task (E and s together are easier to attack than s alone in figure 14a). **Conclusion: In general, when given sufficient steps multi-task models perform as poorly as a mono-tasks models and multi-task attacks plateau earlier than their multi-task counterparts.**

Impact of Epsilon We evaluate our different tasks combinations under different strength of attacks. We present the results of 4 tasks in figure 16, for each of the main tasks (s,D,E,n), the boxplots reflects the relative error across various auxiliary tasks,both in the mono-task attack context and in the multi-task context. Our results show that strength of attacks impact differently the different tasks. While, the image segmentation task (s) and the Normal prediction (n) task display a linear error with the increased epsilon, the Edge (E) and Depth tasks (D) show an exponential vulnerability to the strength of the attack. This different behaviour reflects both in multi-task attacks and mono-task attacks. It is worth noticing that this different behaviour cannot be explained by the nature of metric used (task S uses mIOu error and cross entropy loss while task n uses MSE and L1 loss) nor the amplitude of error (n and E have closer range of values than E and D).

Our results also hint that under highs attack budgets, mono-task attacks and the multi-task attacks achieve close performance, while the variance of robustness provided by the different auxiliary tasks widens.

Impact of Norm We evaluate in this context the impact of using norm L2 in our attacks, under low perturbation amount (epsilon=4), and limited attack steps (25).

Figure 17 shows that the relative task vulerability introduced by an L2 attack is very limited in comparison with what we can achieve with an L-infinite under the same configuration. **Conclusion: While improving the robustness against L2 attacks, multi-task learning provides little defense against L-infinite attacks.**

4 Appendix D: Source code

We provide in the review package a folder *Code* with 3 components:
Our source code is under the MIT licence.

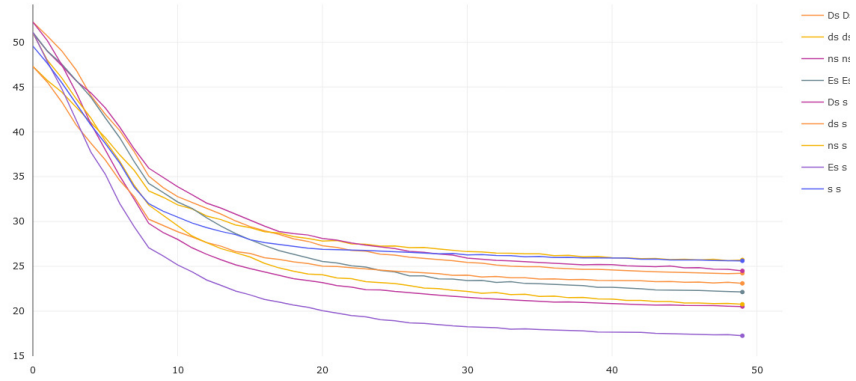
- **MTRobust**: cloned from `https://github.com/columbia/MTRobust/` and extended with additional models (Xception, WideResnet). This repository is used to train the models following the same setting as the original paper of MTRobust. Read their documentation for more instructions about how to train models. Or use our scripts in *MTRobust/jobs/*.

  You will need to download the Taskonomy dataset as explained in the original repository and update the configuration files in *MTRobust/jobs/*.

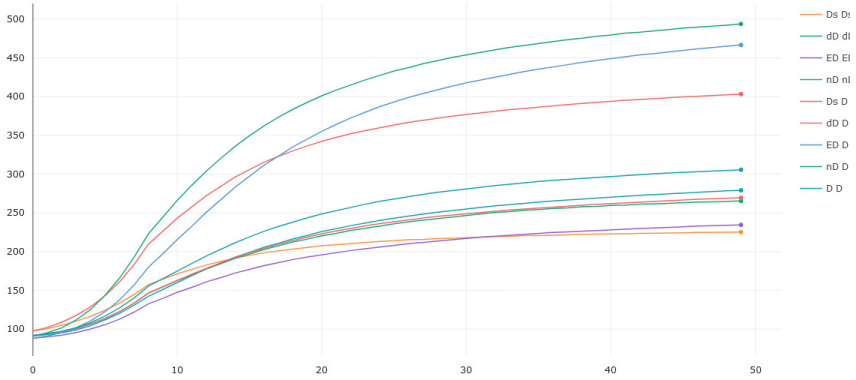
- **MTVulnerability**: Our package to attack and evaluate the vulnerability of the models. The folder *MTVulnerability/jobs* contains the script you can run directly. Please read the specific *README* file of our package for more details & instructions.

- **models**: This folder provides one pretrained Taskonomy model for task combination s (semantic segmentation) and d (Z-depth) to use as a quick test. You can use this folder for the variable **MODEL** in the scripts located in *MTVulnerability/jobs*.

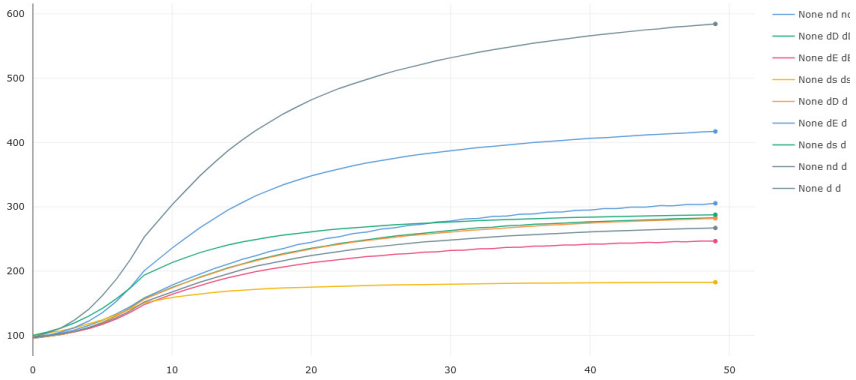
Our experiments use CometML to track and record the results of our experiments. You will need a valid (free) account from `https://www.comet.ml/` and a personal API Key to send the results of the experiments.



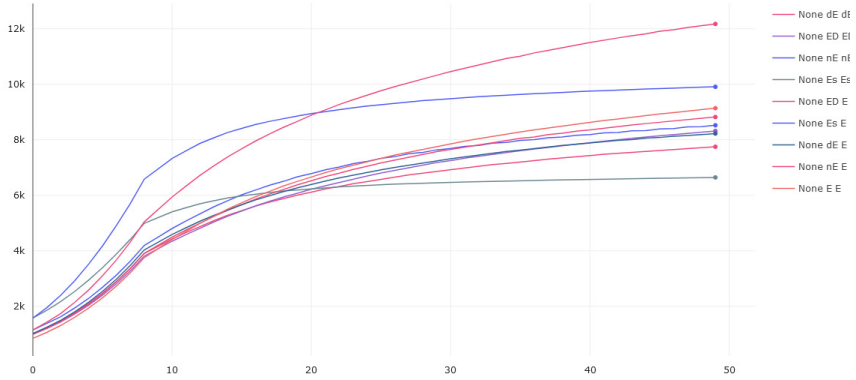
(a) mIoU of Semantic segmentation task



(b) MSE of Euclidian Depth task

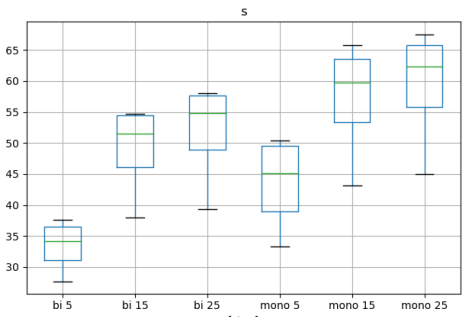


(c) MSE of Z-Depth task

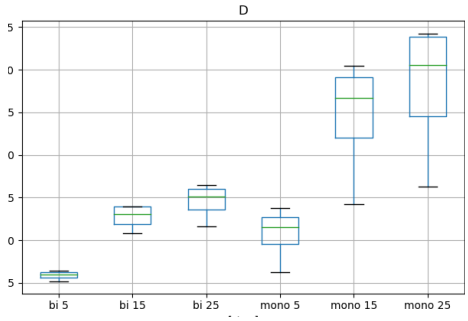


(d) MSE of Edge occlusion task

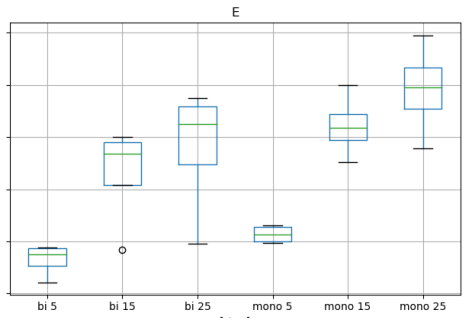
Figure 14: Impact of attack steps on the performance of different tasks .Legend: The first letters are the tasks the model has been trained on. The second letters are the tasks that are attacked



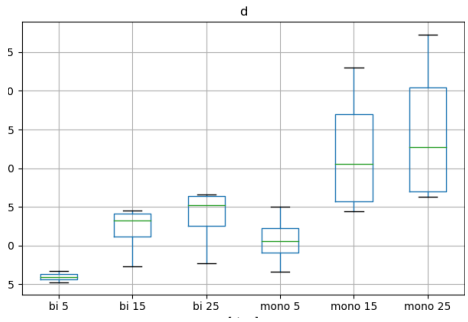
(a) mIoU of Semantic Segmentation



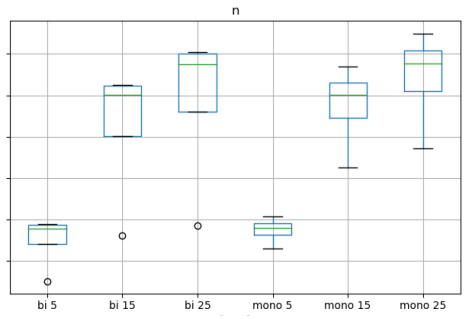
(b) MSE of Euclidian Depth



(c) MSE of Edge Occlusion

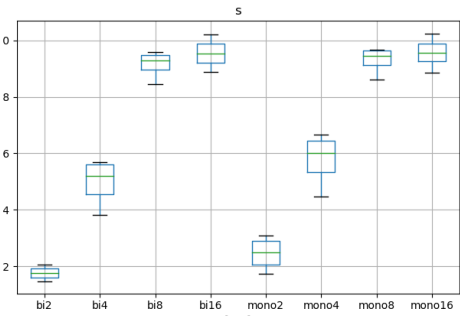


(d) MSE of Z-Depth

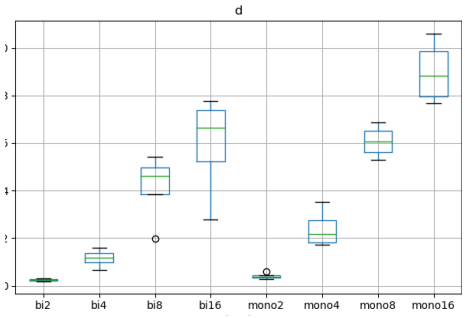


(e) MSE of Normal

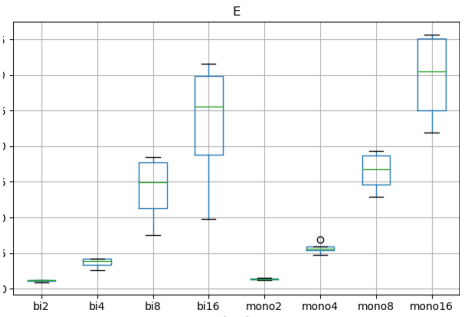
Figure 15: Impact of attack steps on the performance of different tasks: We evaluate the relative task robustness of models for 3 different attack steps: 5, 15 and 25; for adversarial attacks against the main task only (mono) or both tasks (multi)



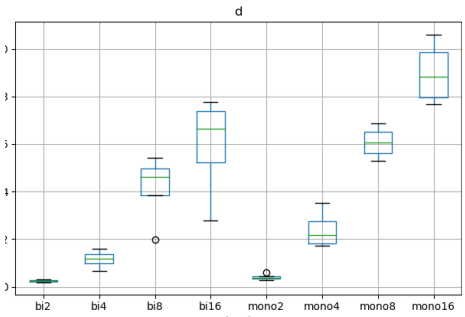
(a) mIoU of Semantic segmentation task



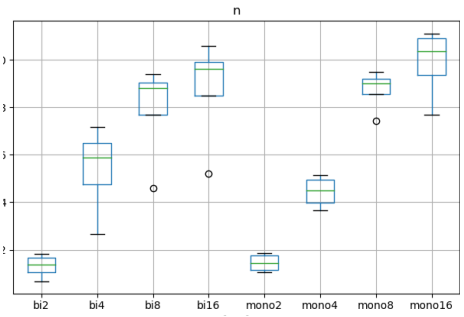
(b) MSE of Euclidian Depth task



(c) MSE of Edge occlusion task

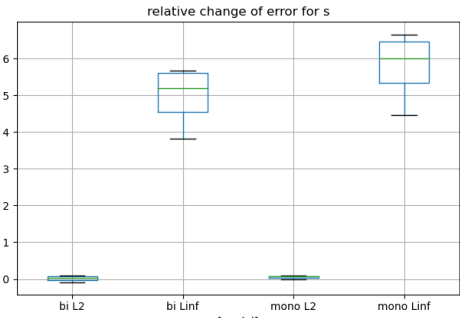


(d) MSE of Z-Depth task

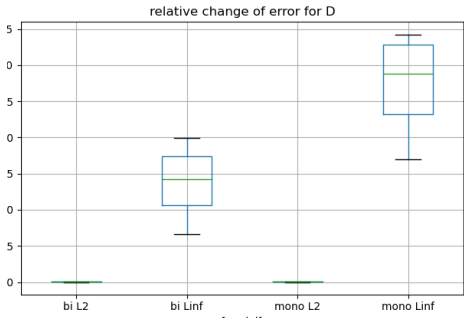


(e) MSE of Normal

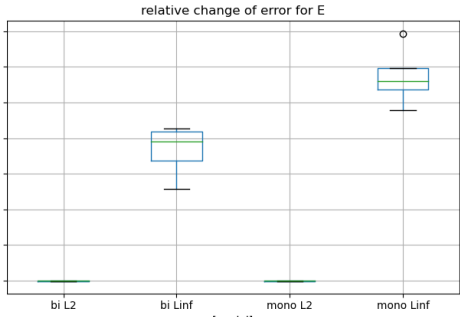
Figure 16: Impact of attack strength on the performance of different tasks: We evaluate the relative task robustness of models for different attack budgets ϵ: 2/255; 4/255; 8/255 and 16/255.



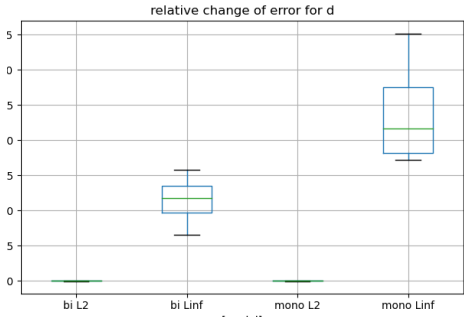
(a) mIoU of Semantic segmentation task



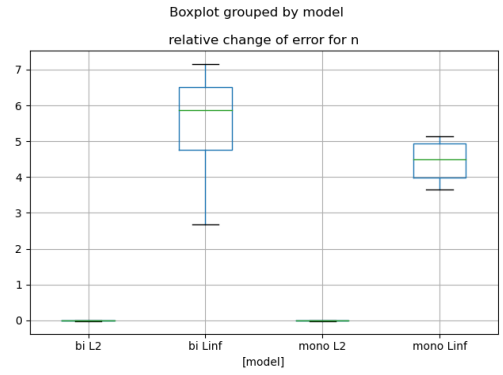
(b) MSE of Euclidian Depth task



(c) MSE of Edge occlusion task



(d) MSE of Z-Depth task



(e) MSE of Normal

Figure 17: Impact of attack norm on the performance of different tasks