MulT: An End-to-End Multitask Learning Transformer
Supplementary

Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk and Mathieu Salzmann
School of Computer and Communication Sciences, EPFL, Switzerland
{deblina.bhattacharjee, tong.zhang, sabine.susstrunk, mathieu.salzmann}@epfl.ch

We present additional discussions and experiments, particularly the ablation study analyzing the effect of the shared attention in our MulT model, the performances of our 4-task and 5-task networks as well as the ablation study of the effect of the network size on the different task combinations. We also analyze the number of parameters required by each model. We show additional qualitative results comparing the performance of the different models on the Taskonomy [12] and Replica [9] benchmarks. Finally, we discuss the environmental impact of training our model and ways to mitigate it. The paper is organized as follows:

1. Effect of shared attention

To account for the task dependencies beyond sharing encoder parameters, we develop a shared attention mechanism that integrates the information contained in the encoded features into the decoding stream. Empirically, we have found that the attention from the surface normal task stream benefits our 6-task MulT model and we thus take this task as reference task r, whose attention is shared across the tasks.

In Table 1, we show the relative performance of our 6-task MulT model with a single-task dedicated Swin transformer baseline [7] under two settings. In the first setting the 6-task MulT model is trained *without* the shared attention across the 6 tasks, whereas in the second setting our MulT model is trained *with* the shared attention. The shared attention mechanism benefits the performance of our MulT model, allowing it to learn task inter-dependencies. The models under both the scenarios comprise an increased size of the network.

Feature fusion method. We further explore the different feature fusion methods such as concatenation and cross-attention. Concatenating all the features does not benefit our network to learn task interdependencies, as observed in our preliminary experiments, and was thus not reported. Our method *is* a learnable fusion strategy, using a learnable shared attention (SA) mechanism. We also tried the cross-attention (CA) mechanism from CrossVit [2], but it did not beat our SA mechanism in the given multitask setting, as seen in Table 2

2. Task combinations

We now show the effect of different task combinations on the relative performance of each task. From our experiments in Table 3, we observe that the performance of 2D keypoints, 2D edges and segmentation benefits from the inclusion of other tasks like surface normal estimation, depth and reshading. In particular, surface normal estimation is the most beneficial task for the other tasks. For instance, any task with the combination of surface normal estimation, leverages the surface statistics to improve its performance.

We also observe that increasing the number of tasks improves the results of our MulT model, e.g., a 6-task network outperforms a 5-task one, which in turn outperforms a 4-task network. Note all the models in Table 3 are trained with shared attention to learn task inter-dependencies.

3. Effect of network size

As more number of tasks are added to our MulT model, we observe, as in [8], that effectively leveraging between 3 and 6 tasks required increasing the size of the network modules. Altogether, reporting results for all possible task combinations requires training $(2^6 - 1)$ models. We see that improving the network size has significant effect on the relative performance of the different tasks. We quantitatively evaluate all the task combinations in the 4-task, 5-task and 6-task settings; with and without an increase in the network size. For the normal network size, we use swin-T as the backbone [7] containing $(2, 2, 6, 2)$ transformer blocks in the respective stages of the encoder, whereas for the in-

	Relative Performance On					
	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{K}$	E	$\mathcal{R}$
6-task MulT w/o shared attention	+15.0%	+8.13%	+6.92%	+42.9%	+81.3%	+14.8%
6-task MulT w/ shared attention	**+19.7%**	**+10.2%**	**+8.72%**	**+94.75%**	**+88.8%**	**+16.4%**

Table 1. **Effect of shared attention on our MulT model.** We show the relative performance of our 6-task MulT model with a single-task dedicated Swin transformer baseline [7] under two settings- *without* and *with* the shared attention mechanism. Note that under both the settings, our MulT model comprises the increase network size. We show, the relative performance percentage for each task evaluated by taking the percentage increase or decrease w.r.t. the single-task dedicated Swin transformer baseline [7]. The shared attention mechanism benefits the performance of our MulT model, allowing it to learn task inter-dependencies. The results here are reported on the Taskonomy test set. Bold and underlined values show the best and second-best results, respectively.

	Relative Performance for 6 task MulT vs 1-task SWIN on Taskonomy					
	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{K}$	E	$\mathcal{R}$
MulT w/ CA	+1.06%	+5.11%	-3.33%	+13.3%	+25.9%	+0.06%
MulT w/ SA	**+19.7%**	**+10.2%**	**+8.72%**	**+94.75%**	**+88.8%**	**+16.4%**

Table 2. **Quantitative comparison of training the 6-task MulT model on the Taskonomy benchmark [12] with Cross attention (CA) [2] and our proposed shared attention (SA).** Our shared attention mechanism benefits MulT where it consistently outperforms the MulT with CA method. Bold values show the best results.

creased network we use swin-L [7] as the backbone containing $(2, 2, 18, 2)$ transformer blocks in the respective stages of the encoder. The increase in the number of transformer blocks in the third stage of the encoder in the swin-L backbone helps to accommodate the increased number of tasks. Our best performing MulT model comprises the increased network size and shared attention.

In Table 3, we observe that increasing the number of tasks improves the results of our MulT model, where a 6-task network outperforms a 5-task one, which in turn outperforms a 4-task network. Note all the models in Table 3 are trained with shared attention to learn task inter-dependencies.

4. Parameter comparison

We show the number of parameters learnt by our 6-task MulT model *without* an increased network size and compare it to the number of parameters learnt by the multitasking Resnet50 baseline and the single dedicated Swin-Tiny (Swin-T) baseline. Further, we show the number of parameters learnt by our 6-task MulT model *with* an increased network size and compare it to the number of parameters learnt by the multitasking Resnet152 baseline and the single dedicated Swin-Large (Swin-L) baseline. We see that our MulT model, both without and with an increased network size, is more parameter efficient than the 1-task dedicated Swin-T and Swin-L models, respectively. Note that the number of parameters and the inference time of six 1-task Swin-T models and six 1-task Swin-L models are added to get the total number of parameters and the total inference time for all the six tasks. Our MulT model learns more number of parameters than the multitasking CNN baselines [3] but infers the final predictions across the six tasks in comparable time.

5. Additional qualitative results

We qualitatively compare the results of our MulT model with different CNN-based multitask baselines [4, 8, 11, 12], as well as with the single task dedicated Swin transformer [7]. The results in Figure 1 and Figure 2 show the performance of the different networks across multiple vision tasks on the Taskonomy benchmark [12] and Replica test set [9], respectively. All the multitasking models are jointly trained on the six tasks on the Taskonomy benchmark, and the single task dedicated Swin models are trained on the respective tasks. Our MulT model yields higher-quality predictions than both the single task Swin baselines and the multitask CNN baselines.

6. Environmental impact

Models consume power both during training as well as during inference. However, a bigger source of energy consumption today comes from after the models are deployed, i.e. during the inference stage [1]. Nvidia estimated that in 2019, 80–90% of the cost of a model is in the inference. To worsen this, machine learning practitioners waste a ton of resources on redundant training [1]. By being a multitask framework, our MulT model helps to reduce the power consumption during inference unlike the single task baselines that need to be run multiple times to achieve the predictions on the different tasks. A shown in Table 4, our MulT model requires less inference time than the single task transformer baselines while reporting better performance. Nonetheless, running our MulT model in the cloud takes 21 hours to train on 32 Nvidia V100-SXM2-32GB GPUs, where a single GPU emits 3.11kg of CO_2 with a CO_2 offset of 1.55kg [5]. This is equivalent to 12.5 kilometers driven by an average car [6]. To mitigate, the carbon footprint of training our model we have reputable carbon offsets as well as follow a centralised cloud infrastructure with sustainable power supplies. Furthermore, by employing an efficient shared attention mechanism as [10], that operates in linear time, we can

Effect of the network size on different task combinations for Taskonomy test [12]													
		w/ increased network size						**w/o increased network size**					
No. of Tasks	Trained on	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{K}$	E	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{K}$	E	$\mathcal{R}$
4-task MulT	$\mathcal{S}+\mathcal{D}+\mathcal{N}+\mathcal{K}$	+13.8%	+8.36%	+6.91%	+82.2%	-	-	+7.84%	+6.95%	+5.07%	+75.4%	-	-
	$\mathcal{S}+\mathcal{D}+\mathcal{N}+E$	+14.0%	+8.38%	+7.05%	-	+74.9%	-	+8.08%	+7.11%	+5.10%	-	+63.3%	-
	$\mathcal{S}+\mathcal{D}+\mathcal{N}+\mathcal{R}$	+14.2%	+8.55%	+7.17%	-	-	+9.13%	+8.11%	+7.20%	+5.33%	-	-	+6.77%
	$\mathcal{S}+\mathcal{D}+\mathcal{K}+E$	+13.5%	+8.08%	-	+73.0%	+74.6%	-	+7.41%	+6.84%	-	+67.7%	+62.7%	-
	$\mathcal{S}+\mathcal{D}+\mathcal{K}+\mathcal{R}$	+14.0%	+8.22%	-	+72.4%	-	+8.91%	+8.03%	+6.95%	-	+66.2%	-	+6.39%
	$\mathcal{S}+\mathcal{D}+E+\mathcal{R}$	+14.3%	+8.30%	-	-	+73.1%	+9.04%	+8.22%	+6.98%	-	-	+61.5%	+6.73%
	$\mathcal{S}+\mathcal{N}+E+\mathcal{R}$	+15.0%	-	+7.13%	-	+73.9%	+9.17%	+8.80%	-	+5.28%	-	+61.8%	+6.80%
	$\mathcal{S}+\mathcal{N}+\mathcal{K}+\mathcal{R}$	+14.9%	-	+7.01%	+87.5%	-	+8.99%	+8.61%	-	+5.12%	+79.0%	-	+6.45%
	$\mathcal{S}+\mathcal{N}+\mathcal{K}+E$	+14.7%	-	+6.89%	+88.4%	+75.4%	-	+8.55%	-	+5.05%	+79.7%	+66.9%	-
	$\mathcal{S}+\mathcal{K}+E+\mathcal{R}$	+13.7%	-	-	+73.5%	+74.5%	+8.97%	+7.72%	-	-	+68.9%	+62.5%	+6.42%
	$\mathcal{D}+\mathcal{K}+E+\mathcal{R}$	-	+7.91%	-	+73.3%	+74.8%	+9.88%	-	+6.63%	-	+68.4%	+63.0%	+7.00%
	$\mathcal{D}+\mathcal{N}+\mathcal{K}+\mathcal{R}$	-	+8.44%	+7.20%	+87.0%	-	+10.3%	-	+7.15%	+5.40%	+78.8%	-	+7.33%
	$\mathcal{D}+\mathcal{N}+E+\mathcal{R}$	-	+8.63%	+7.25%	-	+75.5%	+11.1%	-	+7.29%	+5.49%	-	+66.8%	+8.12%
	$\mathcal{D}+\mathcal{N}+\mathcal{K}+E$	-	+8.40%	+7.10%	+87.2%	+75.8%	-	-	+7.12%	+5.20%	+79.2%	+67.0%	-
	$\mathcal{N}+\mathcal{K}+E+\mathcal{R}$	-	-	+7.12%	+88.8%	+75.0%	+10.6%	-	-	+5.27%	+80.1%	+66.1%	+7.74%
5-task MulT	$\mathcal{S}+\mathcal{D}+\mathcal{N}+\mathcal{K}+E$	+17.2%	+9.07%	+8.11 %	+92.5 %	+82.6%	-	+11.6%	+7.75%	+6.16%	+89.9%	+72.5%	-
	$\mathcal{S}+\mathcal{D}+\mathcal{N}+\mathcal{K}+\mathcal{R}$	+17.7%	+9.10%	+7.59%	+92.0%	-	+12.9%	+12.0%	+7.91%	+5.94%	+89.5%	-	+10.0 %
	$\mathcal{S}+\mathcal{D}+\mathcal{N}+E+\mathcal{R}$	+16.9%	+9.22%	$\underline{+8.26\%}$	-	$\underline{+82.9\%}$	+12.7%	+10.8%	+8.08%	$\underline{+6.47\%}$	-	$\underline{+72.9\%}$	+9.71%
	$\mathcal{S}+\mathcal{D}+\mathcal{K}+E+\mathcal{R}$	+15.1%	+8.86%	-	+75.0%	+78.8%	+10.2%	+9.10%	+7.47%	-	+70.7%	+67.7%	+7.80%
	$\mathcal{S}+\mathcal{N}+\mathcal{K}+E+\mathcal{R}$	$\underline{+18.3\%}$	-	+7.33%	$\underline{+94.1\%}$	+82.2%	+13.0%	$\underline{+12.5\%}$	-	+5.55%	$\underline{+91.9\%}$	+72.2%	+10.3%
	$\mathcal{D}+\mathcal{N}+\mathcal{K}+E+\mathcal{R}$	-	+$\underline{9.77\%}$	+8.06%	+93.9%	+82.6%	$\underline{+13.8\%}$	-	$\underline{+8.33\%}$	+6.11%	+91.6%	+72.5%	$\underline{+10.7\%}$
6-task MulT	$\mathcal{S}+\mathcal{D}+\mathcal{N}+\mathcal{K}+E+\mathcal{R}$	**+19.7%**	**+10.2%**	**+8.72%**	**+94.7%**	**+88.8%**	**+16.4%**	**+13.8%**	**+9.11%**	**+6.99%**	**+92.5%**	**+78.3%**	**+12.9%**

Table 3. **Quantitative comparison of training different task combinations in our MulT model on the Taskonomy benchmark [12].** Increasing the number of tasks improves the results of our MulT models, where a 6-task network outperforms a 5-task one, which in turn outperforms a 4-task network. Note all the models are trained with shared attention to learn task inter-dependencies. The relative performance percentage for each task is evaluated by taking the percentage increase or decrease w.r.t. the single-task swin [7] baseline. Bold and underlined values show the best and second-best results, respectively.

Parameter Comparison		
Model	No. of Params (M)	Inference time (ms)
Multitasking Resnet50 [3]	153.6	12
six 1-task Swin-T [7]	344	90
MulT w/o increased network	231	13
Multitasking Resnet152 [3]	361.2	27
six 1-task Swin-L [7]	728	198
MulT w/ increased network	545	29

Table 4. **Parameter comparison of our 6-task MulT model with the baselines.** We see that our MulT model, both without and with an increased network size, is more parameter efficient than the 1-task dedicated Swin-T and Swin-L models, respectively. Note that the number of parameters and the inference time of six 1-task Swin-T models and six 1-task Swin-L models are added to get the total number of parameters and the total inference time for all the six tasks. Further, our MulT model learns more number of parameters than the multitasking CNN baselines [3] but infers the final predictions across the six tasks in comparable time.

extend our mitigation efforts and reduce the overall hours of GPU computation.

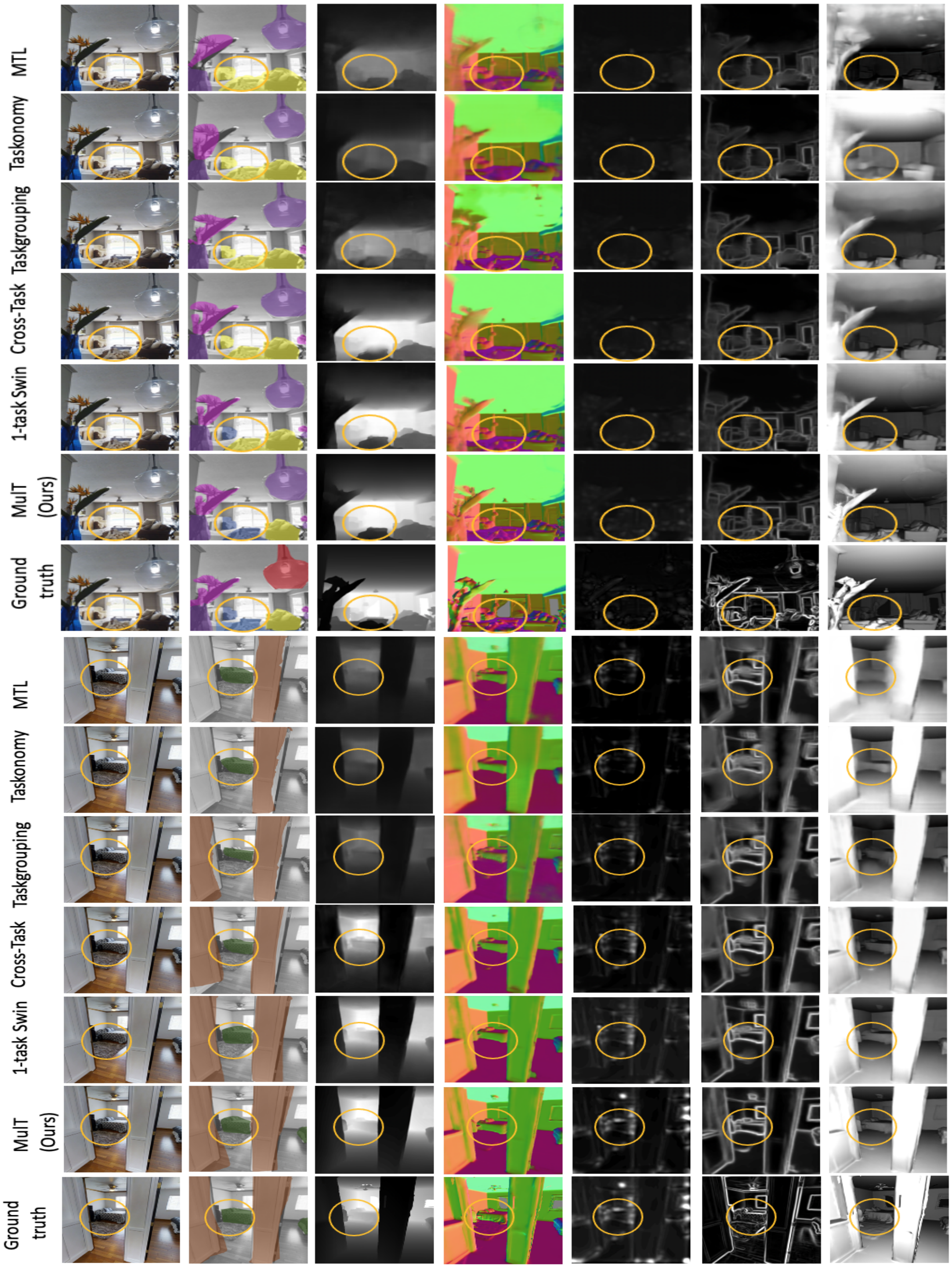

Figure 1. **Qualitative comparison on the six vision tasks** of the Taskonomy benchmark [12]. From top to bottom, we show qualitative results using MTL [4], Taskonomy [12], Taskgrouping [8], Cross-task consistency [11], the single-task dedicated Swin transformer [7] and our six-task **MulT** model. We show, from left to right, the input image, the semantic segmentation results, the depth predictions, the surface normal estimations, the 2D keypoint detections, the 2D edge detections and the reshading results for all the models. All models are jointly trained on the six vision tasks, except for the Swin transformer baseline, which is trained on the independent single tasks. Our MulT model outperforms both the single task Swin baselines and the multitask CNN based baselines. Best seen on screen and zoomed within the yellow circled regions.

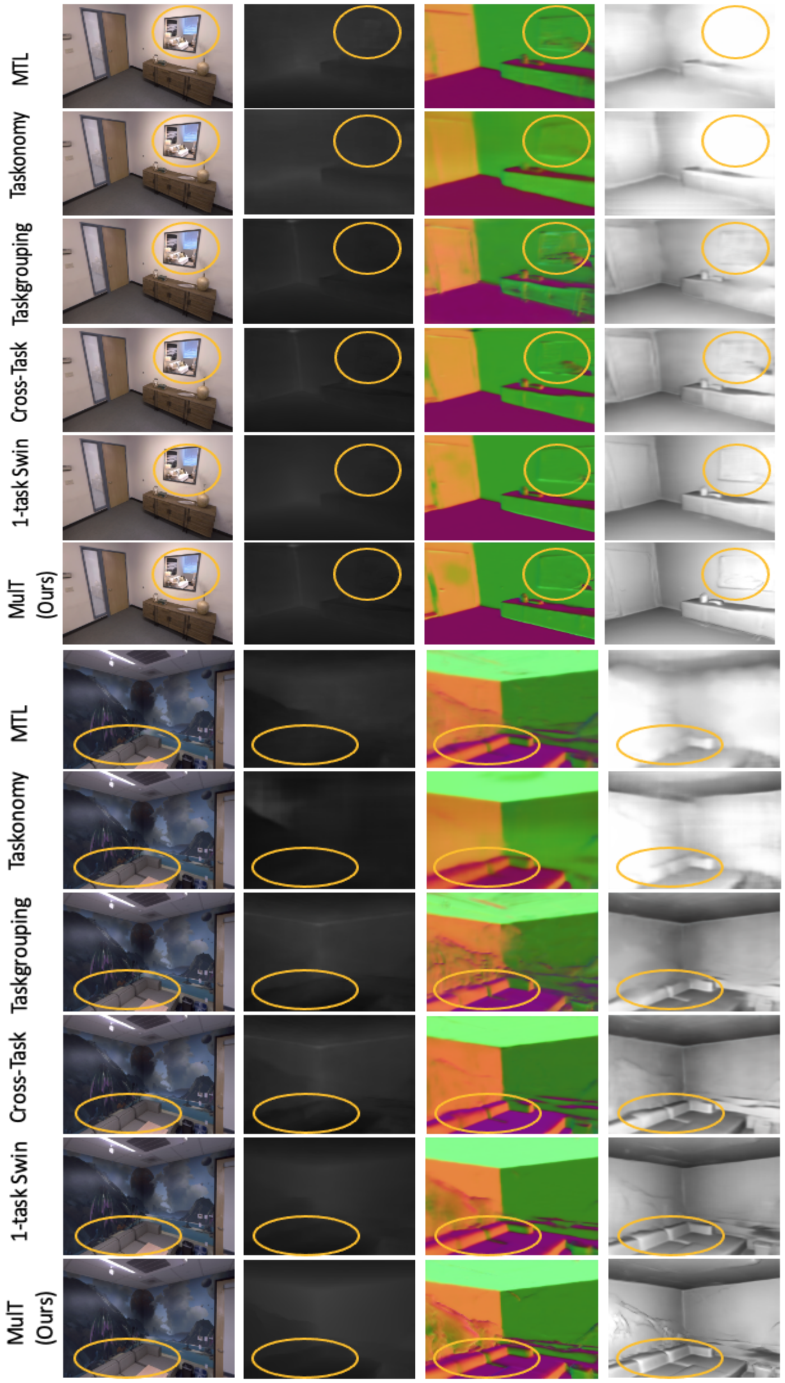

Figure 2. **Qualitative comparison on the three vision tasks** of the Replica benchmark [9]. From top to bottom, we show qualitative results using MTL [4], Taskonomy [12], Taskgrouping [8], Cross-task consistency [11], the single-task dedicated Swin transformer [7] and our six-task **MulT** model. We show, from left to right, the input image, the depth predictions, the surface normal estimations and the reshading results for all the models. All models are jointly trained on the *six* vision tasks of the Taskonomy benchmark and are then fine-tuned to the Replica official training set, except for the Swin transformer baseline, which is trained on the independent *single* tasks. Our MulT model outperforms both the single task Swin baselines and the multitask CNN based baselines. Best seen on screen and zoomed within the yellow circled regions.

References

[1] Lucas Biewald. Deep learning and carbon emissions https://towardsdatascience.com/deep-learning-and-carbon-emissions-79723d5bc86e, accessed november 2021. 2

[2] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *CoRR*, abs/2103.14899, 2021. 1, 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015. 2, 3

[4] Iasonas Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *arXiv: 1609.02132, cs.CV*, 2016. 2, 4, 5

[5] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Ml co2 impact, https://mlco2.github.io/impact, accessed november 2021. 2

[6] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv:1910.09700, cs.CV*, 2019. 2

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv: 2103.14030, cs.CV*, 2021. 1, 2, 3, 4, 5

[8] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv: 1905.07553, cs.CV*, 2019. 1, 2, 4, 5

[9] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, and Erik Wijmans et. al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797, CC-BY 4.0*, 2019. 1, 2, 5

[10] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv: 2107.00641, cs.CV*, 2021. 2

[11] Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task consistency. *arXiv*, 2020. 2, 4, 5

[12] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (MIT License Copyright (c) 2017 Stanford Vision and Learning Group), 2018. 1, 2, 3, 4, 5