Supplementary Materials: Multi-task Curriculum Learning Based on Gradient Similarity

Hiroaki Igarashi[1]
hiroaki.igarashi.j3m@jp.denso.com
Kenichi Yoneji[1]
kenichi.yoneji.j3v@jp.denso.com
Kohta Ishikawa[2]
ishikawa.kohta@core.d-itlab.co.jp
Rei Kawakami[3]*
reikawa@sc.e.titech.ac.jp
Teppei Suzuki[2]
suzuki.teppei@core.d-itlab.co.jp
Shingo Yashima[2]
yashima.shingo@core.d-itlab.co.jp
Ikuro Sato[2,3]
sato.ikuro@core.d-itlab.co.jp

[1] DENSO CORPORATION
Tokyo, Japan
[2] DENSO IT Laboratory
Tokyo, Japan
[3] Tokyo Institute of Technology
Tokyo, Japan

In this supplementary document, we report follows:

A. Additional experimental results of the NYUv2 dataset [6]

B. Visualization results of the cosine similarity and weight

C. The problem of gradient manipulation

A Additional Experimental Results

In the main paper, we reported the experimental results of the NYUv2 dataset [6], where the optimizer is changed from Adam to SGD, because the adaptive learning rate in the Adam optimizer is not suitable with MCLGS. However, this change in the evaluation setup [3] could be unfair for the other existing methods. Thus, we also evaluated each methods with the Adam optimizer following the setting in [3]. Additionally, Δm is recalculated using the result of the STL baseline with the Adam optimizer.

We present the full results on the NYUv2 dataset in Table 1. The Δm of MCLGS with CAGrad using SGD reaches -5.01% and outperforms all the other methods, while that of

* Work done while at DENSO IT Laboratory.

#P.	Method	Optim.	Weighting	Segmentation Accuracy↑ mIoU	Segmentation Accuracy↑ Pix Acc	Depth Error↓ Abs Err	Depth Error↓ Rel Err	Surface Normal Angle Distance↓ Mean	Surface Normal Angle Distance↓ Median	Surface Normal Within t° ↑ 11.25	Surface Normal Within t° ↑ 22.5	Surface Normal Within t° ↑ 30	$\Delta m\%$↓ (mean ± stderr)
3	STL Baseline	Adam	-	37.88	63.43	0.6287	0.2532	**24.96**	**19.12**	**30.14**	**57.46**	**69.38**	-
		SGD	-	**39.33**	**64.55**	**0.5785**	**0.2339**	26.12	20.44	28.04	54.56	66.84	-1.72 ± 1.21
1.77	MTL Baseline (MTAN [18])	Adam	equal	38.49	64.68	0.5532	0.2273	28.00	23.73	22.23	47.90	61.48	1.73 ± 1.67
			uncert.	37.51	64.35	**0.5351**	0.2215	**26.51**	21.69	25.56	52.12	**65.20**	-1.31 ± 0.53
		SGD	equal	**40.88**	**66.14**	0.5489	0.2290	27.83	23.23	23.63	48.93	62.15	-0.42 ± 0.77
			uncert.	38.95	64.76	0.5423	**0.2185**	26.55	**21.67**	**25.69**	**52.12**	65.20	**-2.08 ± 0.99**
1.77	MGDA [18]	Adam	equal	30.17	60.56	0.6116	0.2419	**24.66**	**18.88**	**30.69**	**58.10**	**70.06**	2.52 ± 0.96
			uncert.	**39.25**	**65.17**	**0.5644**	**0.2217**	24.75	19.44	29.54	56.90	69.34	**-4.58 ± 1.16**
		SGD	equal	20.52	53.16	0.6635	0.2532	26.00	20.45	28.07	54.58	66.96	13.03 ± 1.13
			uncert.	36.42	63.54	0.5912	0.2286	25.51	19.95	28.89	55.70	68.06	-0.96 ± 0.42
1.77	PCGrad [18]	Adam	equal	39.07	64.91	0.5542	0.2229	27.25	22.74	24.00	49.88	63.22	-0.18 ± 1.65
			uncert.	38.03	64.48	**0.5353**	0.2243	**26.00**	**20.97**	**26.91**	**53.56**	**66.41**	-2.35 ± 0.97
		SGD	equal	**40.73**	**66.24**	0.5558	0.2275	27.70	23.07	23.70	49.25	62.49	-0.47 ± 0.85
			uncert.	39.05	65.10	0.5366	**0.2163**	26.36	21.35	26.38	52.81	65.70	**-2.95 ± 0.64**
1.77	GradDrop [18]	Adam	equal	**40.57**	65.82	0.5413	0.2242	27.99	23.80	22.08	47.78	61.44	0.06 ± 0.82
			uncert.	38.37	64.56	**0.5369**	0.2270	**26.37**	21.57	25.62	52.36	**65.53**	-1.48 ± 0.45
		SGD	equal	40.56	**66.13**	0.5538	0.2251	27.90	23.35	23.26	48.70	61.99	-0.23 ± 1.14
			uncert.	39.18	64.87	0.5397	**0.2201**	26.42	**21.52**	**26.04**	**52.42**	65.44	**-2.39 ± 1.14**
1.77	CAGrad [18]	Adam	equal	**40.06**	**65.87**	**0.5325**	**0.2155**	25.69	20.78	26.94	54.02	67.05	**-4.53 ± 0.63**
			uncert.	39.31	65.43	0.5394	0.2283	**25.57**	20.71	27.19	54.22	**67.30**	-3.22 ± 1.19
		SGD	equal	39.39	65.27	0.5578	0.2270	25.88	20.61	27.69	54.34	66.98	-2.85 ± 1.33
			uncert.	37.51	64.05	0.5722	0.2339	25.93	**20.57**	**27.77**	**54.40**	66.99	-0.90 ± 1.02
1.77	MCLGS (ours)	Adam	equal	37.89	63.79	0.5646	0.2288	27.97	23.68	22.74	48.06	61.46	2.47 ± 1.72
			uncert.	37.28	63.52	0.5478	0.2273	26.35	21.30	26.53	52.85	65.64	-0.78 ± 0.34
		SGD	equal	**40.98**	**66.43**	0.5729	0.2358	27.51	22.96	23.96	49.44	62.65	0.19 ± 0.83
			uncert.	40.20	65.63	**0.5429**	**0.2174**	**26.03**	**21.11**	**26.68**	**53.32**	**66.26**	**-3.71 ± 0.76**
1.77	CAGrad + MCLGS (ours)	Adam	equal	38.82	64.94	0.5436	**0.2151**	26.24	21.56	25.80	52.37	65.59	-2.47 ± 1.48
			uncert.	38.25	64.73	**0.5364**	0.2198	25.80	20.93	26.99	53.65	66.66	-2.90 ± 0.64
		SGD	equal	**41.37**	**66.47**	0.5513	0.2230	**25.50**	20.22	28.26	55.16	67.75	**-5.01 ± 0.57**
			uncert.	39.72	65.65	0.5470	0.2222	25.51	**20.15**	**28.47**	**55.33**	**67.82**	-4.33 ± 0.96

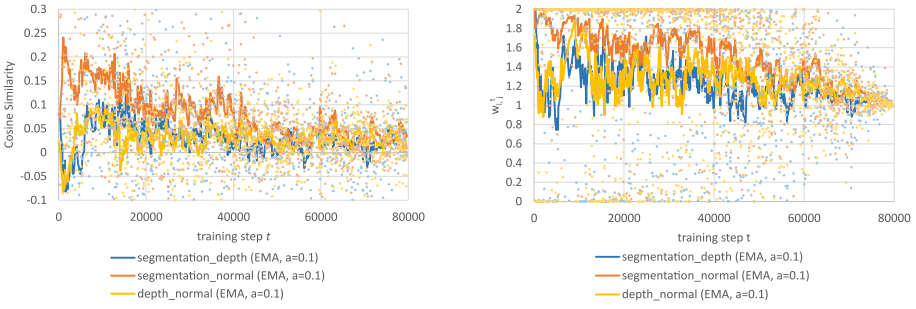
Table 1: Multi-task learning resuluts of the NYUv2 dataset: MCLGS with CAGrad outperforms all the other methods. MCLGS without CAGrad is better than PCGrad and GradDrop. For the loss weighting scheme, "equal" represents no loss balancing, and "uncert" denotes the uncertainty weigh loss [20]. #P denotes the relative model size compared to the vanilla SegNet. The best average result for each method is marked in bold. The best average result among all multi-task methods is annotated with boxes.

MCLGS using the SGD optimizer is -3.71% which is better than PCGrad and GradDrop. Additionally, since the adaptive learning rate of Adam is incompatible with MCLGS, MCLGS is more compatible with SGD than Adam.

Additionally, we found that MGDA with uncertainty weigh loss performs effectively. While MGDA with equal weighting is biased against surface normal prediction, the uncertainty weigh loss improves the performance of semantic segmentation and depth estimation. This may be because MGDA breaks the balance of loss functions to maintain the Pareto-optimal, while uncertainty weigh loss restores the balance. This is reasonable because CAGrad, which introduces a balance constraint of loss function into MGDA, also performs well.

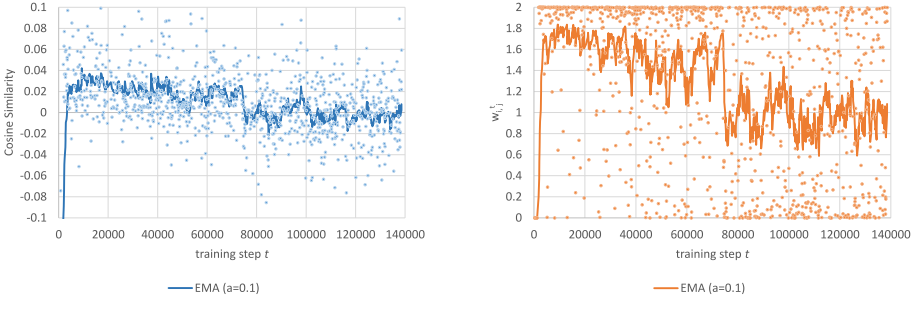
B Visualization Results of the Cosine Similarity and Weight

To confirm that the weight $w_{i,j}^t$ is generated depending on the cosine similarity, we visualized the cosine similarity and the weight generated by MCLGS in Figures 1 and 2. As designed, $w_{i,j}^t$ fluctuates a lot depending on the cosine similarity but converges around 1.0 as learning



(a) Cosine similarity (b) $w_{i,j}^t$

Figure 1: Visualization results of cosine similarity and $w_{i,j}^t$ with increasing training step t of the NYUv2 dataset. The weight is generated depending on the cosine similarity. Segmentation, depth, and normal represent semantic segmentation, depth estimation, and surface normal prediction, respectively. Raw values are plotted as points, while lines represent the exponential mean average (EMA) of each value.

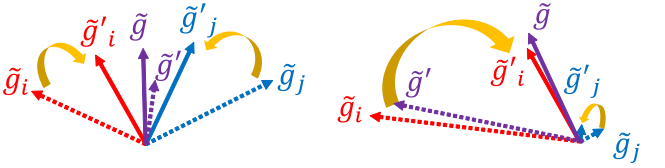


(a) Cosine similarity (b) $w_{i,j}^t$

Figure 2: Visualization results of cosine similarity and $w_{i,j}^t$ with increasing training step t of the BDD100K dataset. The weight is generated depending on the cosine similarity. Raw values are plotted as points while lines represent the exponential mean average (EMA) of each value.

progresses because MCLGS includes samples generating gradient conflicts at the end of the training. Additionally, we observed that the cosine similarity also tends to converge to 0, which means that the direction of task-wise gradients is orthogonal.

As shown in Figure 1, the cosine similarity between semantic segmentation and surface normal estimation is higher than that at the beginning of the NYUv2 dataset. This means that the gradient of depth estimation highlights a different direction from that of the others. One possible reason could be that semantic segmentation and surface normal prediction are prone to using object boundary information, while depth estimation is not. Furthermore, the model might be trained to take this information initially.



(a) Successful example (b) Failed example

Figure 3: Examples that PCGrad [7] works well and not. (a) Projected average gradient $\tilde{g}$ is still similar to the original average gradient $\tilde{g}'$, while gradient conflict is removed. (b) Projected gradient $\tilde{g}$ is far from the original average gradient $\tilde{g}'$ even though gradient conflict is removed. In this case, the conflicted component of the task gradient $\tilde{g}_i$ is dominant on the average gradient because the norm of $\tilde{g}_i$ is much larger than that of $\tilde{g}_j$. However, this component is eliminated by PCGrad.

C The Problem of Gradient Manipulation

In the main paper, we described that gradient manipulation leads to non-optimal solution for the original objectives. In this section, we will give the details of gradient manipulation and the case updating parameters into non-optimal direction.

For example, PCGrad [7] manipulated gradients such that the conflicting components were removed, and only the orthogonal components of each gradient were extracted and used for the update. Gradient manipulation of PCGrad is formulated as follows:

$$\tilde{g}_i = \tilde{g}_i - \frac{\tilde{g}_i \cdot \tilde{g}_j}{\|\tilde{g}_j\|^2} \tilde{g}_j, \tag{1}$$

where $\tilde{g}_i$ and $\tilde{g}_j$ denote batch gradients of the i-th and j-th task, respectively. Note that Eq. 1 represents the manipulation for $\tilde{g}_i$, but PCGrad also applied this manipulation for $\tilde{g}_j$ as well. Figure 3 shows examples of gradient projection by PCGrad. As shown in Figure 3 (a), if the magnitude of task-wise gradients is similar, the projected average gradient $\tilde{g}$ is also similar to the original average gradient $\tilde{g}'$. Therefore, the retained solution is around the original objectives in this case. However, as shown in Figure 3 (b), if the magnitude of task-wise gradient is much different, the retained solution is far from the original objectives. In this case, although conflicted component of $\tilde{g}_i$ is dominant on the average gradient $\tilde{g}$, PCGrad eliminated this component to remove gradient conflict.

References

[1] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33: 2039–2050, 2020.

[2] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[3] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

[5] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[6] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[7] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.