Supplementary Material: Achievement-based Training Progress Balancing for Multi-Task Learning

Hayoung Yun and Hanjoo Cho*
Samsung Research
{hayoung.yun, hanjoo.cho}@samsung.com

A. Training Details for the NYU v2 dataset

A.1. Network Architecture

We employed ResNet50 [7] with dilated convolutions [1] as backbone. Dilated convolutions were applied for stages 3 and 4 to replace with stride convolutions to keep the feature size. As a result, the feature size of backbone output enlarges from 15 x 20 to 60 x 80. We performed prediction using the DeepLabV3 [2] predictor on the output features. Due to the significant computational burden of the ASPP (atrous spatial pyramid pooling) module in DeepLab, we designed our architecture so that all tasks share a single ASPP, rather than each task having its own ASPP as in previous approaches [9, 8, 16]. The numbers of GMAC (Giga Multiply-Accumulate Operations) for different single-task and multi-task models are described in Table A. Our multi-task models reduced the number of computations to 33.00% by sharing ASPP, whereas the multi-task models with individual ASPP for each task reduced to 60.58% (Table B).

	ResNet50	ASPP	Prediction	Total
segmentation	116.800	72.038	2.880	191.718
depth estimation	116.800	72.038	2.832	191.670
surface normal	116.800	72.038	2.834	191.672
multi-task	116.800	72.038	8.536	197.385

Table A. GMAC comparison for single-task and multi-task models

A.2. Training configurations

We trained single-task and multi-task models for 100 epochs with a batch size of 8. We adopted an ADAM optimizer with a momentum of 0.9 and a weight decay of 5e-4. We tried 10 times for each of the learning rates of 8e-4, 4e-4, 2e-4, 1e-4, and 8e-5, scheduled by cosine decay without warmup. We selected the multi-task model with the best Acc_{MTL} of each trial and computed their average accuracy metrics for each learning rate, excluding the maximum and minimum accuracy (hence, the average of eight).

*Corresponding author

B. Experimental Results for the NYU dataset

B.1. Detailed Results for Various Prediction Heads

In this subsection, we will demonstrate the detailed results for various prediction heads, depicted in Table 2, which were shared DeepLabV3, shared DeepLabV3+, individual DeepLabV3, and individual DeepLabV3+. The numbers of GMAC for the architectures were shown in Table B. The results of DeepLabV3 with shared ASPP were presented in Table 1. Then, we provided the results for DeepLabV3+ with shared ASPP (Table C), and individual ASPP with DeepLabV3 (Table D) and DeepLabV3+ (Table E).

ASPP	Head	ResNet50	ASPP	Prediction	Total
Shared	DeepLabV3 [2]	116.80	72.04	8.54	197.39
	DeepLabV3+ [3]	116.80	72.04	17.67	206.51
Individual	DeepLabV3 [2]	116.80	216.11	8.54	341.46
	DeepLabV3+ [3]	116.80	216.11	18.93	351.84

Table B. GMAC comparison for different network architectures

B.2. Results for the MobileNetV2 Backbone

The detailed results for the MobileNetV2 [12] backbone were described in Table F. Because the output feature size of the MobileNet backbone was significantly reduced by 32, we employed the shared DeepLabV3+ prediction head to exploit high resolution features.

B.3. Results for the EfficientNetV2 Backbone

The detailed results for the EfficientNetV2-S [12] backbone were described in Table G. Like MobileNetV2, the output feature size was reduced by 32, so the DeepLabV3+ prediction head was employed.

B.4. Segmentation and Depth Estimation

The experimental results for two tasks, semantic segmentation and depth estimation, were described in Table H.

	methods	segmentation	depth estimation		surface normal			total		
		$mIoU \uparrow$	$\delta_1 \uparrow$	$rmse \downarrow$	$mean \downarrow$	$median \downarrow$	$11.25 \uparrow$	$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.4449	0.8054	0.5801	19.4138	13.2616	0.4536	0.3986	0.00%	
Constant	Uniform	0.4447	0.8098	0.5756	22.7259	17.4917	0.3377	0.3683	-8.07%	30.98
Scale -based	RLW [8]	0.4436	0.8084	0.5765	22.8454	17.6649	0.3326	0.3665	-8.55%	31.55
	DWA [11]	0.4429	0.8108	0.5760	22.7235	17.5215	0.3369	0.3677	-8.25%	30.77
	GLS [5]	0.4313	0.8238	0.5606	20.8125	15.1440	0.3954	0.3835	-3.88%	31.27
Gradient -based	MGDA [13]	0.2896	0.7670	0.6231	19.2335	13.1608	0.4562	0.3394	-13.41%	76.78
	PCGrad [17]	0.4439	0.8019	0.5841	23.9044	18.8431	0.3097	0.3581	-11.04%	58.02
	CAGrad [9]	0.4440	0.8021	0.5824	23.9114	18.8163	0.3103	0.3584	-10.94%	58.99
	GradNorm [4]	0.4429	0.7850	0.5972	22.3589	16.8694	0.3542	0.3677	-8.21%	36.19
	IMTL-G [10]	0.4346	0.8039	0.5799	20.5369	14.7189	0.4082	0.3838	-3.78%	35.33
	IMTL [10]	0.4200	0.7897	0.5935	20.9657	14.9806	0.4017	0.3746	-6.18%	57.85
Accuracy -based	DTP [6]	0.4422	0.7513	0.6225	22.4029	16.8250	0.3557	0.3625	-9.63%	31.46
	AMTL	0.4344	0.8211	0.5670	20.8688	15.1885	0.3943	0.3831	-3.98%	30.65

Table C. Comparison to the benchmark and proposed multi-task losses for the DeepLabV3+ prediction head with shared ASPP on the NYU v2 dataset. $mIoU$, δ_1, 11.25, Acc_{MTL}, and Δ_{MTL} are better when higher while $rmse$, $mean$, and $median$ are better when lower. $time$ denotes the average training time for epoch in seconds.

	methods	segmentation	depth estimation		surface normal			total		
		$mIoU \uparrow$	$\delta_1 \uparrow$	$rmse \downarrow$	$mean \downarrow$	$median \downarrow$	$11.25 \uparrow$	$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.4437	0.8087	0.5814	19.3462	13.2045	0.4553	0.3989	-	
Constant	Uniform	0.4443	0.8150	0.5766	22.3212	16.3137	0.3738	0.3763	-6.00%	43.31
Scale -based	RLW [8]	0.4471	0.8130	0.5725	22.3235	16.3094	0.3741	0.3774	-5.71%	43.22
	DWA [11]	0.4466	0.8141	0.5730	22.3439	16.3962	0.3721	0.3768	-5.87%	43.01
	GLS [5]	0.4397	0.8220	0.5681	20.5555	14.5141	0.4156	0.3895	-2.41%	43.28
Gradient -based	MGDA [13]	0.3863	0.8008	0.5840	19.6941	13.5648	0.4437	0.3770	-5.34%	88.97
	PCGrad [17]	0.4468	0.8111	0.5763	23.4195	17.4143	0.3532	0.3697	-7.95%	63.91
	CAGrad [9]	0.4467	0.8106	0.5793	23.4502	17.4594	0.3513	0.3689	-8.15%	63.01
	GradNorm [4]	0.4455	0.8167	0.5715	22.3188	16.3126	0.3734	0.3773	-5.74%	57.90
	IMTL-G [10]	0.4268	0.8199	0.5665	19.8490	13.7477	0.4377	0.3917	-1.79%	57.89
	IMTL [10]	0.4218	0.8074	0.5783	20.6877	14.5346	0.4161	0.3815	-4.43%	99.27
Accuracy -based	DTP [6]	0.4436	0.7958	0.5951	22.1462	16.1081	0.3773	0.3739	-6.63%	43.43
	AMTL	0.4414	0.8207	0.5674	20.5774	14.5073	0.4160	0.3899	-2.29%	44.08

Table D. Comparison to the benchmark and proposed multi-task losses for the DeepLabV3 prediction head with individual ASPP on the NYU v2 dataset.

methods		segmentation	depth estimation		surface normal			total		
		$mIoU\uparrow$	$\delta_1\uparrow$	$rmse\downarrow$	$mean\downarrow$	$median\downarrow$	11.25 $\uparrow$	$Acc_{MTL}\uparrow$	$\Delta_{MTL}\uparrow$	$time$
Single-Task		0.4449	0.8054	0.5801	19.4138	13.2616	0.4536	0.3986	-	
Constant	Uniform	0.4437	0.8142	0.5726	22.2965	16.3707	0.3716	0.3762	-5.96%	45.61
Scale -based	RLW [8]	0.4457	0.8132	0.5769	22.2547	16.3355	0.3723	0.3764	-5.88%	43.37
	DWA [11]	0.4445	0.8140	0.5757	22.2624	16.3621	0.3715	0.3761	-5.97%	43.83
	GLS [5]	0.4386	0.8211	0.5657	20.6328	14.6300	0.4128	0.3885	-2.58%	44.12
Gradient -based	MGDA [13]	0.3950	0.8022	0.5833	19.7439	13.6664	0.4406	0.3793	-4.75%	90.15
	PCGrad [17]	0.4454	0.8114	0.5768	23.0143	17.1312	0.3562	0.3710	-7.44%	64.25
	CAGrad [9]	0.4467	0.8104	0.5789	23.0637	17.1736	0.3552	0.3708	-7.51%	63.75
	GradNorm [4]	0.4458	0.8136	0.5767	22.3320	16.4119	0.3705	0.3760	-6.02%	59.13
	IMTL-G [10]	0.4396	0.8161	0.5680	20.3185	14.2563	0.4230	0.3910	-1.93%	58.67
	IMTL [10]	0.4240	0.8055	0.5837	20.7662	14.6261	0.4134	0.3807	-4.57%	103.50
Accuracy -based	DTP [6]	0.4463	0.8012	0.5874	22.2058	16.2499	0.3743	0.3751	-6.24%	43.10
	AMTL	0.4251	0.8249	0.5636	20.2782	14.1928	0.4247	0.3883	-2.59%	43.39

Table E. Comparison to the benchmark and proposed multi-task losses for the DeepLabV3+ prediction head with individual ASPP on the NYU v2 dataset.

methods		segmentation	depth estimation		surface normal			total		
		$mIoU\uparrow$	$\delta_1\uparrow$	$rmse\downarrow$	$mean\downarrow$	$median\downarrow$	11.25 $\uparrow$	$Acc_{MTL}\uparrow$	$\Delta_{MTL}\uparrow$	$time$
Single-Task		0.3798	0.7685	0.6323	21.2734	14.8062	0.4151	0.3581	-	
Constant	Uniform	0.3850	0.7761	0.6180	25.0320	19.8745	0.2976	0.3313	-7.91%	16.04
Scale -based	RLW [8]	0.3825	0.7706	0.6235	25.2629	20.1929	0.2914	0.3280	-8.92%	19.14
	DWA [11]	0.3836	0.7767	0.6185	24.9665	19.8221	0.2982	0.3311	-7.94%	19.08
	GLS [5]	0.3662	0.7930	0.6048	22.4388	16.6333	0.3633	0.3464	-3.30%	17.26
Gradient -based	MGDA [13]	0.2578	0.7307	0.6646	20.8591	14.6106	0.4183	0.3109	-11.93%	32.31
	PCGrad [17]	0.3855	0.7653	0.6284	26.5895	21.7927	0.2674	0.3204	-11.44%	23.71
	CAGrad [9]	0.3843	0.7670	0.6270	26.5724	21.8092	0.2670	0.3202	-11.48%	23.61
	GradNorm [4]	0.3841	0.7570	0.6339	24.1916	18.6917	0.3213	0.3346	-6.87%	20.96
	IMTL-G [10]	0.3646	0.7763	0.6181	21.5271	15.3898	0.3958	0.3513	-1.88%	21.04
	IMTL [10]	0.3619	0.7658	0.6298	22.0750	16.0011	0.3802	0.3445	-3.82%	35.06
Accuracy -based	DTP [6]	0.3844	0.7310	0.6566	24.5180	19.0092	0.3150	0.3289	-8.58%	16.04
	AMTL	0.3696	0.7927	0.6070	22.3961	16.5574	0.3651	0.3476	-2.95%	16.25

Table F. Comparison to recent multi-task losses for the MobileNetv2 backbone and shared DeepLabV3+ prediction head on the NYU v2 dataset.

methods		segmentation	depth estimation		surface normal			total		
		$mIoU \uparrow$	$\delta_1 \uparrow$	$rmse \downarrow$	$mean \downarrow$	$median \downarrow$	$11.25 \uparrow$	$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.4622	0.8300	0.5540	21.3341	16.0052	0.3730	0.3877	-	
Constant	Uniform	0.4861	0.8335	0.5500	22.3076	17.1379	0.3458	0.3868	-0.20%	23.67
Scale -based	RLW [8]	0.4839	0.8284	0.5535	22.6282	17.4877	0.3377	0.3830	-1.20%	24.24
	DWA [11]	0.4856	0.8327	0.5479	22.2851	17.1112	0.3462	0.3870	-0.14%	24.29
	GLS [5]	0.4607	0.8427	0.5418	20.6434	15.1945	0.3942	0.3958	2.06%	24.65
Gradient -based	MGDA [13]	0.2915	0.7195	0.6470	20.0508	14.3504	0.4211	0.3268	-14.08%	58.70
	PCGrad [17]	0.4865	0.8238	0.5554	23.7069	18.8369	0.3094	0.3743	-3.51%	40.83
	CAGrad [9]	0.4881	0.8226	0.5580	23.7155	18.8571	0.3094	0.3742	-3.52%	42.19
	GradNorm [4]	0.4889	0.8091	0.5639	21.9632	16.6797	0.3573	0.3873	-0.06%	31.59
	IMTL-G [10]	0.4710	0.8113	0.5639	20.1275	14.4924	0.4155	0.3991	2.90%	29.22
	IMTL [10]	0.4725	0.8019	0.5699	20.6113	15.0365	0.3995	0.3936	1.54%	43.88
Accuracy -based	DTP [6]	0.4876	0.7733	0.5911	21.6803	16.2800	0.3672	0.3837	-0.97%	24.36
	AMTL	0.4710	0.8426	0.5396	20.6636	15.2123	0.3941	0.3989	2.85%	24.40

Table G. Comparison to recent multi-task losses for the EfficientNetV2-S backbone and the DeepLabV3+ prediction head with shared ASPP on the NYU v2 dataset.

methods		segmentation	depth estimation		total		
		$mIoU \uparrow$	$\delta_1 \uparrow$	$rmse \downarrow$	$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.4437	0.8087	0.5814	0.7234	0.00%	-
Constant	Uniform	0.4464	0.7994	0.5809	0.7236	0.03%	29.49
Scale -based	RLW [8]	0.4458	0.7996	0.5816	0.7230	-0.06%	33.30
	DWA [11]	0.4484	0.7995	0.5853	0.7239	0.08%	29.29
	GLS [5]	0.4397	0.8091	0.5755	0.7221	-0.19%	29.01
Gradient -based	MGDA [13]	0.4439	0.8098	0.5770	0.7252	0.25%	58.78
	PCGrad [17]	0.4429	0.7936	0.5903	0.7166	-0.94%	43.40
	CAGrad [9]	0.4441	0.7925	0.5944	0.7161	-1.02%	43.05
	GradNorm [4]	0.4477	0.7806	0.6013	0.7142	-1.28%	32.65
	IMTL-G [10]	0.4414	0.8021	0.5810	0.7201	-0.46%	33.12
	IMTL [10]	0.4323	0.7832	0.5976	0.7035	-2.78%	53.60
Accuracy -based	DTP [6]	0.4452	0.7590	0.6124	0.7040	-2.70%	29.69
	AMTL	0.4439	0.8083	0.5771	0.7248	0.20%	29.97

Table H. Comparison to recent multi-task losses for semantic segmentation and depth estimation on the NYU v2 dataset.

B.5. Effect of Dropout for Gradient-based Multi-task Losses

Following TorchVision's implementation, our ASPP incorporated dropout with a rate of 0.5. Dropout is an effective regularization technique. However, unfortunately, it perturbs the gradients of trainable parameters, thereby affecting gradient-based multi-task losses. We compared the multi-task accuracy of gradient-based multi-task losses and the proposed one, both with and without dropout (Table I).

Due to employing task gradients of all the parameters in an iterative optimization process to determine task weights, MGDA [13] demonstrated the most notable improvement in accuracy upon excluding dropout. PCGrad [17] and CAGrad [9] also utilize all gradients to resolve conflict (but they use only once), leading to significant improvement. In contrast, as GradNorm [4] and IMTL-G [10] use only the task gradients of the last shared layer, their accuracy improvements were more modest. The proposed loss provided a slight increase in accuracy.

methods	w/ dropout	w/o dropout	Improv.
MGDA [13]	0.3229	0.3576	10.75%
PCGrad [17]	0.3558	0.3664	2.98%
CAGrad [9]	0.3556	0.3663	3.01%
GradNorm [4]	0.3690	0.3708	0.49%
IMTL-G [10]	0.3846	0.3876	0.78%
AMTL	0.3847	0.3861	0.36%

Table I. Comparison of Acc_{MTL} to the gradient-based and proposed multi-task losses on the NYU v2 dataset.

C. Training Details for the VOC+NYU dataset

C.1. Preprocessing

Images by the PASCAL VOC dataset were resized to 640x640 while images by the NYU datasets were 480x640. Hence, we applied zero padding to expand NYU images without geometric distortion. Then, geometric and photometric augmentations were conducted.

C.2. Network Architecture

We employed EfficientDet [15] as the baseline architecture and EfficientNetV2-S [14] as the backbone. We extracted 3-level features from the backbone, with sizes of 1/16, 1/32, and 1/64. The extracted features passed through a two-stage bi-directional Feature Pyramid Network (bi-FPN) [15] with a channel size of 64 before being supplied to the task-specific prediction heads. The prediction heads were composed of two inverted residual blocks [12] with a channel size of 64. The detection head generated predictions using multi-level features, while pixel-level predictions (segmentation and depth) were made by resizing the features to the largest size (1/16) and concatenating them before making predictions.

C.3. Training configurations

We trained single-task and multi-task models for 200 epochs with a batch size of 32. We adopted an ADAMW optimizer with a momentum of 0.9 and a weight decay of 5e-6. The learning rate was warmed up during two epochs and scheduled by ReduceOnPleatue so that it was reduced by 1/10 whenever Acc_{MTL} was not improved during 20 epochs. We used learning rates of 4e-4, 2e-4, 1e-4, 8e-5, and 4e-5. Then, we presented the results with the best Acc_{MTL} in Table 7.

References

[1] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *Advances in neural information processing systems*, 30, 2017. 1

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018. 2, 3, 4, 5

[5] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A Rawashdeh. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) Workshops*, pages 0–0, 2019. 2, 3, 4

[6] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 270–287, 2018. 2, 3, 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Baijiong Lin, YE Feiyang, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. 1, 2, 3, 4

[9] 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*, 34:18878–18890, 2021. 1, 2, 3, 4, 5

[10] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021. 2, 3, 4, 5

[11] 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. 2, 3, 4

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 5

[13] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31, 2018. 2, 3, 4, 5

[14] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 5

[15] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 5

[16] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning supplementary materials. 1

[17] 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. 2, 3, 4, 5