Reparameterizing Convolutions for Incremental Multi-Task Learning without Task Interference (Supplementary Material)

Menelaos Kanakis[1], David Bruggemann[1], Suman Saha[1], Stamatios Georgoulis[1], Anton Obukhov[1], and Luc Van Gool[1,2]

[1] ETH Zurich [2] KU Leuven

A Implementation Details

We based our implementation details on the work of [8], listed below for completeness.

Generic hyperparamaters. All models are optimized using SGD with a learning rate 0.005, momentum 0.9, weight decay 0.0001, and the "poly" learning rate schedule [1]. We use a single GPU with a minibatch of 8 images. The input images during training are augmented with random horizontal flips and random scaling in the range [0.5, 2.0] in 0.25 increments. The validity of these hyperparameters has already been tested in [8], and hence they are used in all our experiments too, in order to ensure fair comparisons amongst different methods.

Dataset specific hyperparameters. PASCAL-Context [10] models are trained for 60 epochs. The spatial size of the input images is 512×512. NYUD [13] models are trained for 200 epochs. The spatial size of the input images is 425×560. Images of insufficient size are padded with the mean color.

Task weighting and loss functions. As is common in multi-task learning (MTL), losses require careful loss weighting [8, 14, 4, 12], where each loss is task-dependent. For edge detection, we optimize the binary cross-entropy (BCE) loss, scaled by 50. Due to the class imbalance between the edge and non-edge pixels, edge pixels are penalized with a weight 0.95, while non-edge pixels with a scale of 0.05, accommodating [5, 7]. For evaluation, we set the maximum allowed mis-localization of the optimal dataset F-measure (odsF) [9] to 0.0075 and 0.011 for PASCAL-Context and NYUD, respectively, using the package of [11]. Semantic segmentation and human parts segmentation are optimized with cross-entropy loss, weighted by the factors of 1 and 2, respectively. Predictions of surface normals (normalized to unit vectors) and depth modalities are penalized using the $\mathcal{L}_1$ loss, scaled by 10 and 1, respectively. Saliency is optimized using the BCE loss, weighted by a factor of 5.

Table 1. Single-task baseline comparison. We report the single-task performance of the baseline implementations of [8, 14] for similar architectures on PASCAL-Context. The arrow indicates the direction for better performance.

Method	Edge ↑	SemSeg ↑	Parts ↑	Normals ↓	Sal ↑
ASTMT [8]	70.30	63.90	55.90	15.10	63.90
MTI-Net [14]	68.20	64.49	57.43	14.77	66.38
Ours	**71.88**	**66.22**	**59.69**	**13.64**	**66.62**

B Reparameterization Details

In Section 3.3 of the main text (Response initialization, RI), we introduced the methodology for the generation of a better filter bank W_s when compared to that directly learned by pre-training W_s on ImageNet, and demonstrated improved performance when utilizing RI in Section 4. In this section, we present additional detail.

Recall that we defined $\boldsymbol{y} = f(\boldsymbol{x}; W^m) = W^m \boldsymbol{x}$ the responses of a convolutional layer for an input tensor $\boldsymbol{x}$, where $W^m \in \mathbb{R}^{c_{out} \times k^2 c_{in}}$ are the pre-trained ImageNet weights. We specify $Y \in \mathbb{R}^{c_{out} \times n}$ as a matrix containing n responses of $\boldsymbol{y}$ with the mean vector $\overline{\boldsymbol{y}}$ subtracted. To generate the new filter bank, we first compute the eigen-decomposition of the covariance matrix $YY^T = USU^T$ (using Singular Value Decomposition, SVD), where $U \in \mathbb{R}^{c_{out} \times c_{out}}$ is an orthogonal matrix with the eigenvectors on the columns, and S is a diagonal matrix of the corresponding eigenvalues. We can now utilize UU^T which acts as a method to project to (U^T) and from (U) a latent space. Thus, we can rewrite $\boldsymbol{y} = UU^T(\boldsymbol{y} - \overline{\boldsymbol{y}}) + \overline{\boldsymbol{y}}$, with the centering operation being of importance due to the space UU^T being generated from centred responses. This gives rise to

$$\begin{aligned}
\boldsymbol{y} &= W^m \boldsymbol{x} = UU^T(W^m \boldsymbol{x} - \overline{\boldsymbol{y}}) + \overline{\boldsymbol{y}} \\
\boldsymbol{y} &= UU^T W^m \boldsymbol{x} + (\overline{\boldsymbol{y}} - UU^T \overline{\boldsymbol{y}}) \\
\boldsymbol{y} &= W_t^i W_s \boldsymbol{x} + b
\end{aligned} \tag{1}$$

where W_t^i, initialized by U, represents the task-specific parameters optimized independently for each task i, and is implemented as a 1×1 convolution. The non-trainable shared parameters are defined as $W_s = U^T W^m$ and implemented as a $k \times k$ convolution, with k being the filter size of W^m. The bias b can be added to the running mean of the batchnorm following the convolution [3].

C Baseline

To ensure our re-implementation provides a stable baseline, Table 1 compares the single-task performance of our implementation using a ResNet-18 based DeepLabv3+, the results from [14] using a ResNet-18 based FPN [6], and the results from [8] who utilized a ResNet-26 based DeepLabv3+. We demonstrate

Table 2. Comparison with the single-task baseline on PASCAL-Context for a DeepLabv3+ with an R-34 backbone.

Method	Edge ↑	SemSeg ↑	Parts ↑	Normals ↓	Sal ↑	$\Delta_m\%$ ↓
Single-task	73.63	69.34	62.96	13.39	67.49	-
RCM (ours)	72.87	69.11	61.41	13.71	67.69	**1.18**

Table 3. Comparison with the single-task baseline on NYUD for a DeepLabv3+ with an R-34 backbone.

Method	Edge ↑	SemSeg ↑	Normals ↓	Depth ↓	$\Delta_m\%$ ↓
Single-task	70.13	37.39	21.47	0.54	-
RCM (ours)	69.50	36.19	21.70	0.55	**1.77**

that our single-task baseline outperforms both works on every task, and even though the numbers are not directly comparable due to minor implementation differences, it provides a verification of a strong baseline.

D Additional Backbone Experiments

We additionally compare the proposed RCM (Reparameterized Convolutions for Multi-task learning) with respect to the single-task performance on the DeepLabv3+ with the deeper ResNet34 (R-34) [2] backbone. Results for PASCAL-Context [10] and NYUD [13] can be seen in Table 2 and Table 3, respectively. As seen, the percentage drops of 1.18% and 1.77% for PASCAL-Context and NYUD respectively are comparable to that of the ResNet18 backbone reported in the main paper.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
4. Kendall, A., Gal, Y., Cipolla, R.: 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. pp. 7482–7491 (2018)
5. Kokkinos, I.: Pushing the boundaries of boundary detection using deep learning. arXiv preprint arXiv:1511.07386 (2015)
6. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
7. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Convolutional oriented boundaries: From image segmentation to high-level tasks. IEEE transactions on pattern analysis and machine intelligence **40**(4), 819–833 (2017)
8. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1851–1860 (2019)
9. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE transactions on pattern analysis and machine intelligence **26**(5), 530–549 (2004)
10. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 891–898 (2014)
11. Pont-Tuset, J., Marques, F.: Supervised evaluation of image segmentation and object proposal techniques. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1465–1478 (2015)
12. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Advances in Neural Information Processing Systems. pp. 527–538 (2018)
13. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European conference on computer vision. pp. 746–760. Springer (2012)
14. Vandenhende, S., Georgoulis, S., Van Gool, L.: Mti-net: Multi-scale task interaction networks for multi-task learning. arXiv preprint arXiv:2001.06902 (2020)