Task Switching Network for Multi-task Learning
Supplementary material

Guolei Sun[1], Thomas Probst[1], Danda Pani Paudel[1], Nikola Popovic[1], Menelaos Kanakis[1],
Jagruti Patel[1], Dengxin Dai[1,2], Luc Van Gool[1]
[1]Computer Vision Laboratory, ETH Zurich, Switzerland
[2]MPI for Informatics, Germany
guolei.sun@vision.ee.ethz.ch

Overview

- **Sec. 1. Experimental Details:** implementation details for all experiments
- **Sec. 2. Additional Baseline:** additional baseline for multi-task learning
- **Sec. 3. Additional Qualitative Results:** comparisons among qualitative results produced by different methods

1. Experimental Details

Here, we report implementation details for our experiments, which are based on the existing works [5, 2, 12].

Common hyperparameters. The common hyperparameters are used throughout our experiments, unless otherwise stated. Specifically, we use Adam optimizer [3] to optimize all models. The initial learning rate (*lr*) for the decoder is $1e^{-4}$ while we set *lr* to be a smaller value of $1e^{-6}$ for the encoder (ResNet-18 pretrained on ImageNet [9]), in order to keep the generalizability of the features output from encoder. The learning rate is reduced by half if the validation loss doesn't reduce for 10 epochs, *i.e.*, ReduceLROnPlateau. A single Tesla GPU is used for all experiments. Tasks are optimized in an epoch-wise manner, *i.e.*, for each epoch one task finishes training and then another task starts training. The sequence of tasks being trained is random for each epoch. For augmentations, we use random horizontal flips, random rotation, and random cropping. For both training and test, images are normalized by subtracting the mean values of [0.485, 0.456, 0.406], and divided by standard derivation of [0.220, 0.224, 0.225]. The input for models are resized to 512×512. For evaluation on all tasks, single test image is used.

Hyperparameters for PASCAL-Context. All models on PASCAL-Context [7] are trained for a maximum of 130 epochs. The batch size of 16 is used for training.

Hyperparameters for NYUD-Context. All models on NYUD [10] are trained for a maximum of 230 epochs. The batch size of 8 is used for training.

Loss functions and weights. As a common practice for existing multi-task methods [5, 2, 11], different tasks use different loss functions, with different weights. In our experiments, we use the same losses and weights as [5, 2]. Specifically, for *edge detection (Edge)*, we use the weighted binary cross-entropy loss, for which edge pixels are scaled with a weight of 0.95 and non-edge pixels are scaled with 0.05 since the number of non-edge pixels is much more than the number for edge pixels. For both *semantic segmentation (SemSeg)* and *parts segmentation (Parts)*, normal cross entropy loss is used. For *saliency detection (Sal)*, the predictions are penalized using normal binary cross-entropy loss. For both *surface normals (Normals)*

and *depth estimation (Depth)*, we use L_1 loss. Before computing the L_1 loss for $Normals$, the predictions from the network are normalized to unit vectors. The loss weights for *Edge*, *SemSeg*, *Parts*, *Sal*, *Normals*, and *Depth* are 50, 1, 2, 5, 10, and 1, respectively.

Evaluation. To evaluate *edge detection*, the maximum allowed mislocalization of the optimal dataset F-measure (odsF) [6] is set to be 0.0075 for both PASCAL-Context and NYUD, following [8]. For evaluation of other tasks, we follow the standard metrics.

Details for Taskonomy. For Taskonomy dataset [12], we customize our method on following tasks: *colorization*, *inpainting*, *autoencoding*, *denoising*, *2D keypoints*, *2D segment*, *2D edges*, *3D keypoints*, *2.5d segment*, *curvature*, *occlusion edges*, *reshading*, *z-depth*, *distance*, *surface normals*, *semantic segmentation*, *object class*, *scene class*, *room layout*, and *vanishing point*. For details of each task, we refer to [12]. The original Taskonomy dataset contains around 4.6 million images (indoor scenes) for 537 buildings. Each image has annotations for all tasks and semantic information is distilled from models trained on ImageNet [9], MS COCO [4], and MIT Places [13]. In our experiments, we use a small subset, which has 9,464 images from 1 building (Cauthron). More details (image-level statistics, point-level statistics, camera-level statistics and model-level statistics) about the dataset can be found from official Taskonomy website. The optimizer and learning rate are the same as those used for PASCAL-Context and NYUD. Tasks are also optimized in an epoch-wise manner. A single Tesla V100 GPU with 32G memory and batch size of 40 are used. The input for our model is resized to 256×256. The training is terminated after 150 epochs.

For choices of loss functions, we follow the work [12]. Specifically, for classification tasks (*semantic segmentation*, *object class*, and *scene class*), we use cross entropy loss. For most regression tasks (*colorization*, *inpainting*, *autoencoding*, *denoising*, *2D keypoints*, *2D edges*, *3D keypoints*, *curvature*, *occlusion edges*, *reshading*, *z-depth*, *distance*, and *surface normals*), we use L_1 loss. For remaining regression tasks (*room layout* and *vanishing point*), we use mean square error (squared L_2) loss. For unsupervised tasks (*2D segment* and *2.5d segment*), we use triple metric loss [1]. All losses have equal weight.

2. Additional Baseline

We report a simple baseline for multi-task learning, which is to use a shared encoder and decoder, but have different heads (last convolutional layer) for different tasks. Hence the baseline is named "Multi-head". The comparisons are shown in Table 1. As expected, the baseline performs badly, with average performance drop of 9.56%, because tasks are interfering each other heavily. Our method largely improves the baseline, with average performance drop of 0.30%, while only increasing the number of parameters by 0.6 million. Since Multi-decoder baseline has task-specific decoder for different tasks, it is much better than Multi-head (4.32% vs 9.97%). However, it uses much more parameters (43.9 million) and the parameters scale with respect to number of tasks.

Table 1: We compare with Multi-head baseline on PASCAL-Context. It shows serious *task interference* when sharing most parameters of the network among tasks. Our proposed method largely outperforms the Multi-head baseline while only increasing the model size by a small margin.

Method	Edge↑	SemSeg↑	Parts↑	Normals↓	Sal↑	$\Delta_m\%\downarrow$	# params
Single-task	71.3	64.3	55.5	16.3	62.9	-	88.7M
Multi-head	71.1	54.9	50.5	19.4	59.6	9.56	**17.7M**
Multi-decoder	72.2	55.4	55.5	16.8	59.1	4.32	43.9M
Ours	70.6	64.2	55.0	16.3	63.3	**0.30**	18.3M

3. Additional Qualitative Results

More qualitative results are shown in Fig. 1. It shows that our method outperforms baseline (Task-specific INs), especially on high-level tasks such as *semantic segmentation*, *human parts segmentation*, and *saliency detection*.

Figure 1: Qualitative results. We show more visual comparisons between our model with baseline (Task-specific INs). From *left* to *right*: input image, edge prediction, semantic segmentation, parts segmentation, surface normals, and saliency.

References

[1] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv*, 2017. 2

[2] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. *ECCV*, 2020. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 1

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 2

[5] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019. 1

[6] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004. 2

[7] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 1

[8] Jordi Pont-Tuset and Ferran Marques. Supervised evaluation of image segmentation and object proposal techniques. *TPAMI*, 38(7):1465–1478, 2015. 2

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 2

[10] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1

[11] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. *ECCV*, 2020. 1

[12] A. Zamir, Alexander Sax, William Bokui Shen, L. Guibas, Jitendra Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. *CVPR*, 2018. 1, 2

[13] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, 2014. 2