Supplementary Document for

Dynamic Neural Network for Multi-Task Learning Searching across Diverse Network Topologies

Wonhyeok Choi, Sunghoon Im*
Department of Electrical Engineering & Computer Science, DGIST, Daegu, Korea
{smu06117, sunghoonim}@dgist.ac.kr

1. Implementation Details

Central Network Architecture We set the first 12 hidden states, the same as the VGG-16 [35], except for the max-pooled states as:

State	Shape
v_0 (image state)	B, 3, H, W
v_1	B, 64, H, W
v_2	B, 64, H, W
v_3	B, 128, H//2, W//2
v_4	B, 128, H//2, W//2
v_5	B, 256, H//4, W//4
v_6	B, 256, H//4, W//4
v_7	B, 256, H//4, W//4
v_8	B, 512, H//8, W//8
v_9	B, 512, H//8, W//8
v_{10}	B, 512, H//8, W//8
v_{11}	B, 512, H//16, W//16
v_{12}	B, 512, H//16, W//16
v_{13} (read-out state)	B, 512, H//16, W//16

Table 1. **Shape of all hidden states**

where shapes of states are represented as (batch size, number of channels, height, and width). Then, we link the states with edges as a block that consists of sequential operations as follows:

$e_{ij} : v_i \rightarrow v_j$
conv3x3(C_{v_i}, C_{v_j}, padding = 1, stride = 1), BatchNorm(C_{v_j}), ReLU(), Maxpool(kernel size = $H_{v_j} // H_{v_i}$)

Table 2. **The operation block of e_{ij}**

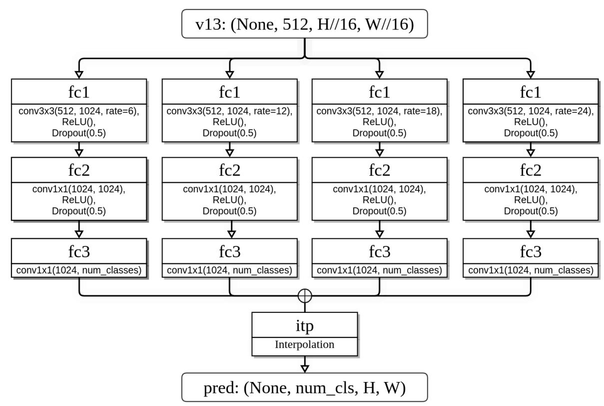

Figure 1. **Task-specific head configuration**

where C_{v_i} is the number of channels of v_i, and H_{v_i} is the height of v_i. We illustrate the overall structure of the central network with $M = 3$ in Fig. 4. The read-in layer embeds the interpolated feature into all hidden states $v_1, v_2, ..., v_{12}$ with $\alpha_i \in \mathcal{A}$. Then, the network sequentially updates the hidden states with task-specific weight $\gamma_{ij} \in \Gamma$ that corresponds to e_{ij}. Lastly, the read-out layer extracts the weighted sum of interpolated hidden states with $\beta_i \in \mathcal{B}$.

Task-specific Head Architecture For NYU-v2 [34], Cityscapes [7], and PASCAL-Context [26], we use the ASPP [5] architecture, a popular architecture for pixel-wise prediction tasks, as our task-specific heads.

Training Details The overall training process of our framework consists of 3 stages: warm-up, search, and fine-tuning. For Omniglot [17], we train the network 2,000, 3,000, and 5,000 iterations for warm-up, search, and fine-tuning stages, respectively. Similarly, for both NYU-v2 [34] and Cityscapes [7], we train the network 5,000, 15,000, and 20,000 iterations for warm-up, search, and fine-tuning stages, respectively. For PASCAL-Context [26], the network is trained for 10,000, 20,000, and 30,000 iterations for the warm-up, search, and fine-tuning stages, respectively.

*Corresponding author

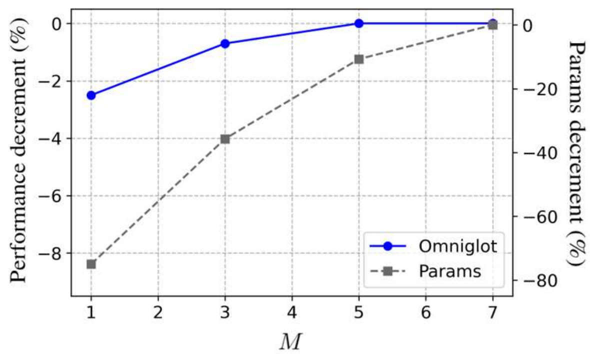

Figure 2. **Model performance with respect to the proposed flow-restriction (Omniglot)**

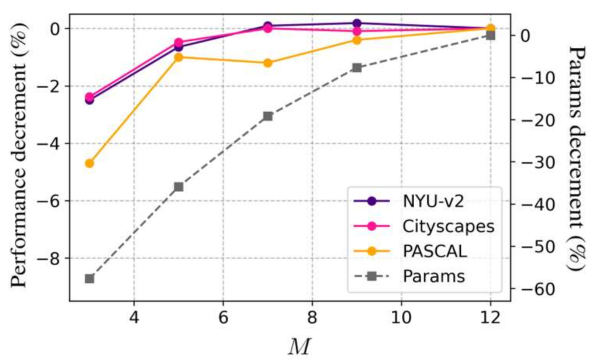

Figure 3. **Model performance with respect to the proposed flow-restriction (NYU-v2, Cityscapes, PASCAL-Context)**

We train all baselines [1, 11, 12, 20, 25, 29, 32, 38] with the same number of fine-tuning iterations for a fair comparison. Before the fine-tuning stage, we rewind the model parameters to the parameters at the end of the warm-up stage. We also report the learning rates of model weights parameters and upper-level parameters, and the balancing hyperparameter of squeeze loss $\mathcal{L}_{sq}$ in the Tab. 3.

2. Full Results of All Metrics

In addition to the relative performance of all datasets (in the main paper), we report all the absolute task performance of NYU-v2, Cityscapes, and PASCAL-Context dataset with baseline in Tab. 5-7.

3. Trade-off Curves of All Datasets

Similar to Sec. 4.4 in the main paper, we analyze performance and computational complexity with respect to the flow constant M for all datasets. We plot the degradation ratio of the performance (left y-axis) and parameter (right y-axis) by changing the flow constant M in Fig. 2-3. The final task performance degradation of each dataset, including Omniglot, NYU-v2, Cityscapes, and PASCAL-Context, is marked by blue, purple, pink, and orange, respectively. Additionally, the number of parameters of search space for Omniglot, and other datasets are marked by a gray dashed line.

Dataset	weight lr	upper lr	λ_{sq}
Omniglot [17]	0.0001	0.01	0.05
NYU-v2 [34]	0.0001	0.01	0.05
Cityscapes [7]	0.0001	0.05	0.01
PASCAL-Context [26]	0.0001	0.01	0.005

Table 3. **Hyperparameters for each dataset** We report the learning rates of model weights parameters (weight lr), and upper-level parameters (upper lr). and balancing weight λ_{sq} for squeeze loss $\mathcal{L}_{sq}$. **Our framework does not use task-balancing parameters.**

4. Ablation Studies

4.1. Three-stage learning scheme

We follow the learning scheme as traditional Nas-style MTL three-stage learning. To show that the three-stage learning scheme boosts the overall performance on multi-task learning scenarios, we report the relative task performance of each stage in Tab. 4.

Method ($M=5$)	$\Delta_{\mathcal{T}_{sem}} \uparrow$	$\Delta_{\mathcal{T}_{dep}} \uparrow$	$\Delta_{\mathcal{T}_{norm}} \uparrow$	$\Delta_{\mathcal{T}} \uparrow$	# of Param $\downarrow$
with three-stages	**0.0**	**0.0**	**0.0**	**0.0**	**1.04**
w/o warm-up	-7.4	-3.7	-3.0	-4.3	**1.04**
w/o search + FBR	-14.8	-0.1	-3.3	-6.1	6.50
w/o fine-tune	-13.6	-9.7	-3.3	-8.9	**1.04**

Table 4. **Ablation studies of three-stages on NYU-v2 dataset**

4.2. Ablation studies on key components

Lastly, we provide the absolute task performance of all metrics for ablation studies of four key components; flow restriction, read-in/out layers, flow-based reduction, and squeeze loss in Tab. 8.

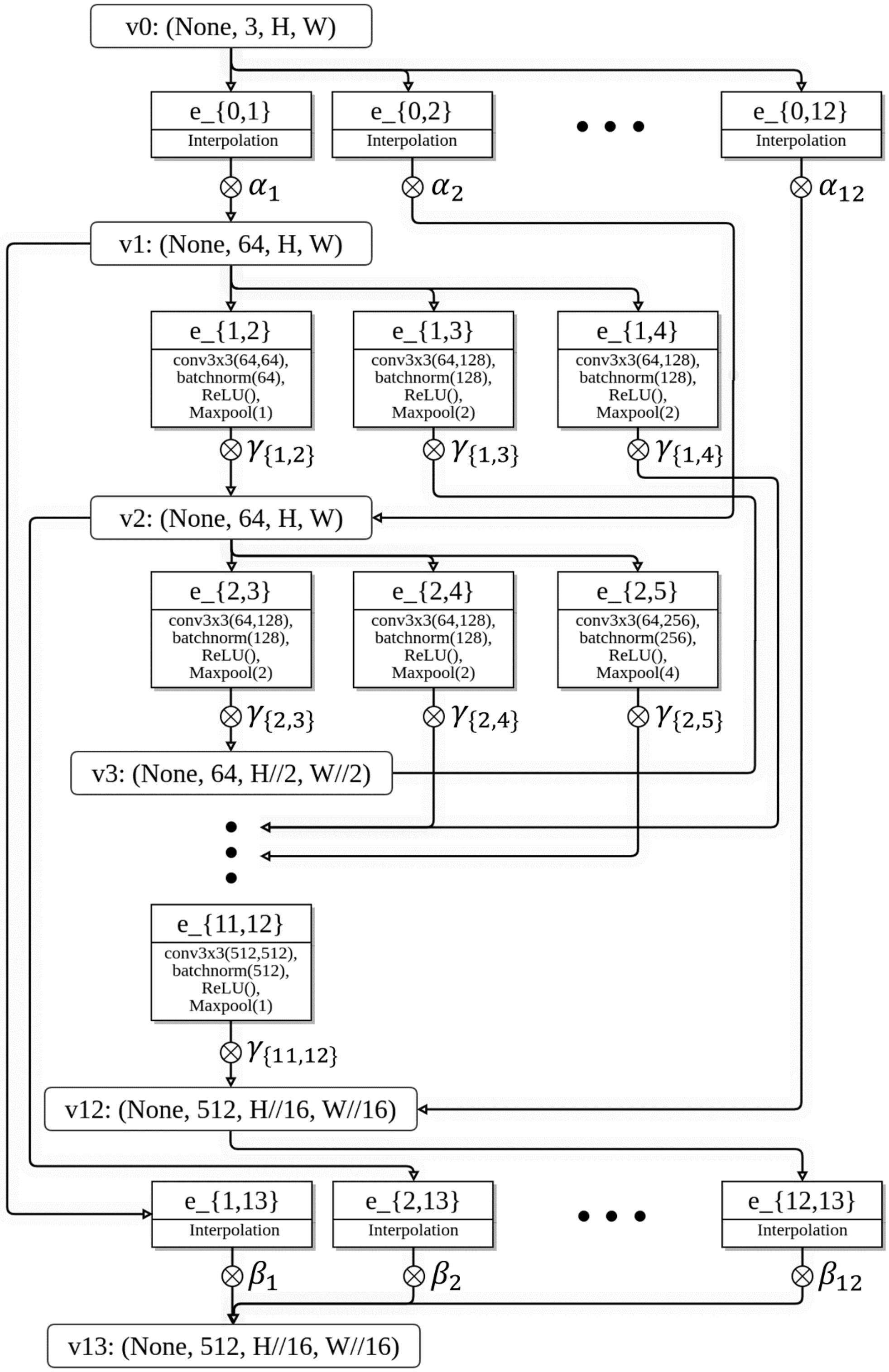

Figure 4. **Central network configuration**

Method	# Params ↓	Semantic Seg.		Depth Prediction					Surface Normal Prediction				
		mIoU ↑	Pixel Acc ↑	Error ↓		θ, within ↑			Error ↓		δ, within ↑		
				Abs	Rel	1.25	1.25^2	1.25^3	Mean	Median	11.25°	22.5°	30°
Single-Task	3	27.5	58.9	0.62	0.25	57.9	85.8	95.7	17.5	15.2	34.9	73.3	85.7
Shared Bottom	**1**	24.1	57.2	0.58	0.23	62.4	88.2	96.5	16.6	13.4	42.5	73.2	84.6
Cross-Stitch	3	25.4	57.6	0.58	0.23	61.4	88.4	95.5	17.2	14.0	41.4	70.5	82.9
Sluice	3	23.8	56.9	0.58	0.24	61.9	88.1	96.3	17.2	14.4	38.9	71.8	83.9
NDDR-CNN	3.15	21.6	53.9	0.66	0.26	55.7	83.7	94.8	17.1	14.5	37.4	73.7	85.6
MTAN	3.11	26.0	57.2	0.57	0.25	62.7	87.7	95.9	16.6	13.0	43.7	73.3	84.4
DEN	1.12	23.9	54.9	0.97	0.31	22.8	62.4	88.2	17.1	14.8	36.0	73.4	85.9
AdaShare	**1**	30.2	62.4	0.55	**0.20**	64.5	90.5	97.8	16.6	**12.9**	**45.0**	71.7	83.0
Ours ($M = 5$)	1.04	31.8	63.7	0.56	0.21	64.3	90.2	97.7	16.5	13.2	43.9	71.7	82.9
Ours ($M = 7$)	1.31	**32.3**	64.3	**0.54**	**0.20**	**64.7**	90.5	98.1	**16.4**	**12.9**	43.1	**73.8**	**86.1**
Ours ($M = 9$)	1.63	32.1	**64.6**	**0.54**	**0.20**	**64.7**	**91.1**	**99.1**	**16.4**	13.1	43.4	**73.8**	86.0

Table 5. **NYU v2 full results**

Model	# Params ↓	Semantic Seg.		Depth Prediction				
		mIoU ↑	Pixel Acc ↑	Error ↓		δ, within ↑		
				Abs	Rel	1.25	1.25^2	1.25^3
Single-Task	2	40.2	74.7	0.017	0.33	70.3	86.3	93.3
Shared Bottom	1	37.7	73.8	0.018	0.34	72.4	88.3	94.2
Cross-Stitch [25]	2	40.3	74.3	**0.015**	**0.30**	74.2	89.3	94.9
Sluice [32]	2	39.8	74.2	0.016	0.31	73.0	88.8	94.6
NDDR-CNN [11]	2.07	41.5	74.2	0.017	0.31	74.0	89.3	94.8
MTAN [20]	2.41	40.8	74.3	**0.015**	0.32	75.1	89.3	94.6
DEN [1]	1.12	38.0	74.2	0.017	0.37	72.3	87.1	93.4
AdaShare [38]	1	41.5	74.9	0.016	0.33	**75.5**	89.8	94.9
Ours ($M = 5$)	**0.96**	42.8	75.1	0.016	0.32	74.8	89.1	94.2
Ours ($M = 7$)	1.16	46.4	**75.6**	0.016	0.33	74.0	89.3	94.0
Ours ($M = 9$)	1.31	**46.5**	75.4	0.016	0.32	75.4	**90.4**	**96.1**

Table 6. **Cityscapes full results**

Method	# Params ↓	Semantic Seg.	Part Seg.	Saliency	Surface Normal	Edge
		mIoU ↑	mIoU ↑	mIoU ↑	Mean ↓	Mean ↓
Single-Task	5	**63.9**	57.6	65.2	14.0	**0.018**
Shared Bottom	**1**	59.7	57.2	63.0	16.0	**0.018**
Cross-Stitch [25]	5	63.1	59.7	65.1	14.2	**0.018**
Sluice [32]	5	62.9	56.9	64.9	14.4	0.019
NDDR-CNN [11]	5.61	63.2	56.1	65.2	14.7	**0.018**
MTAN [20]	5.21	61.6	57.2	65.0	14.7	0.019
AdaShare [38]	**1**	63.1	**59.9**	64.9	14.1	**0.018**
LTB [12]	3.19	59.5	56.5	65.3	14.2	**0.018**
PHN [29]	2.51	59.7	56.7	64.6	14.0	**0.018**
Ours ($M = 5$)	1.93	63.7	59.6	65.8	14.0	**0.018**
Ours ($M = 7$)	1.91	**63.9**	57.5	66.3	**13.8**	**0.018**
Ours ($M = 9$)	2.31	**63.9**	59.7	**66.4**	**13.8**	**0.018**

Table 7. **PASCAL-Context full results**

Method	Semantic Seg.		Depth Prediction					Surface Normal Prediction				
	mIoU ↑	Pixel Acc ↑	Error ↓		θ, within ↑			Error ↓		δ, within ↑		
			Abs	Rel	1.25	1.25^2	1.25^3	Mean	Median	11.25°	22.5°	30°
Ours ($M = 7$)	32.3	64.3	0.54	**0.20**	64.7	90.5	98.1	**16.4**	**12.9**	**43.1**	**73.8**	86.1
w/o flow-restriction	32.1	64.6	0.54	**0.20**	64.2	**90.7**	98.1	16.5	**12.9**	42.9	73.7	**87.2**
w/o read-in/out	31.3	64.5	0.54	**0.20**	64.5	90.3	98.0	16.6	13.0	42.5	73.0	86.3
w/o flow-based reduction	**32.5**	**64.9**	**0.53**	**0.20**	**64.8**	**90.7**	**98.3**	**16.4**	**12.9**	**43.1**	**73.8**	86.3
w/o $\mathcal{L}_{sq}$	32.1	64.6	0.54	**0.20**	64.7	90.5	98.1	16.5	13.0	42.5	73.6	87.0

Table 8. **Ablation Studies in NYU-v2**

References

[1] Chanho Ahn, Eunwoo Kim, and Songhwai Oh. Deep elastic networks with model selection for multi-task learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6529–6538, 2019. 2, 4

[2] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016.

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. *Advances in neural information processing systems*, 29, 2016.

[4] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 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. 1

[6] Ying Chen, Jiong Yu, Yutong Zhao, Jiaying Chen, and Xusheng Du. Task's choice: Pruning-based feature sharing (pbfs) for multi-task learning. *Entropy*, 24(3):432, 2022.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[8] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[9] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[10] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11543–11552, 2020.

[11] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3205–3214, 2019. 2, 4

[12] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, pages 3854–3863. PMLR, 2020. 2, 4

[13] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision*, pages 1062–1070, 2015.

[14] Brendan Jou and Shih-Fu Chang. Deep cross residual learning for multitask visual recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 998–1007, 2016.

[15] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 1, 2

[18] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–473, 2018.

[19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[20] 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, 4

[21] Jiaqi Ma, Zhe Zhao, Jilin Chen, Ang Li, Lichan Hong, and Ed H Chi. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 216–223, 2019.

[22] Krzysztof Maziarz, Efi Kokiopoulou, Andrea Gesmundo, Luciano Sbaiz, Gabor Bartok, and Jesse Berent. Flexible multi-task networks by learning parameter allocation. *arXiv preprint arXiv:1910.04915*, 2019.

[23] Elliot Meyerson and Risto Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. *arXiv preprint arXiv:1711.00108*, 2017.

[24] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing*, pages 293–312. Elsevier, 2019.

[25] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016. 2, 4

[26] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 1, 2

[27] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018.

[28] Prajit Ramachandran and Quoc V Le. Diversity and depth in per-example routing models. In *International Conference on Learning Representations*, 2018.

[29] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2022. 2, 4

[30] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning*, pages 2902–2911. PMLR, 2017.

[31] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[32] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*, 2, 2017. 2, 4

[33] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[34] 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. 1, 2

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[36] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.

[37] Masanori Suganuma, Shinichi Shirakawa, and Tomoharu Nagao. A genetic programming approach to designing convolutional neural network architectures. In *Proceedings of the genetic and evolutionary computation conference*, pages 497–504, 2017.

[38] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020. 2, 4

[39] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.

[40] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[41] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.