Understanding deep meta-learning

Huisman, M.

Bibliography

Alet, F., Doblar, D., Zhou, A., Tenenbaum, J., Kawaguchi, K., and Finn, C. (2021). Noether networks: meta-learning useful conserved quantities. *Advances in Neural Information Processing Systems*, 34:16384–16397.

Alver, S. and Precup, D. (2021). What is going on inside recurrent meta reinforcement learning agents? *arXiv preprint arXiv:2104.14644*.

Anderson, T. (2008). *The Theory and Practice of Online Learning*. AU Press, Athabasca University.

Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29*, pages 3988–3996. Curran Associates Inc.

Antoniou, A., Edwards, H., and Storkey, A. (2019). How to train your MAML. In *International Conference on Learning Representations (ICLR'19)*.

Baik, S., Choi, M., Choi, J., Kim, H., and Lee, K. M. (2020). Meta-learning with adaptive hyperparameters. In *Advances in Neural Information Processing Systems 33*.

Barrett, D. G., Hill, F., Santoro, A., Morcos, A. S., and Lillicrap, T. (2018). Measuring abstract reasoning in neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, pages 4477–4486. JLMR.org.

Bateni, P., Goyal, R., Masrani, V., Wood, F., and Sigal, L. (2020). Improved few-shot visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14493–14502.

Bengio, S., Bengio, Y., Cloutier, J., and Gecsei, J. (1997). On the optimization of a synaptic learning rule. In *Optimality in Artificial and Biological Neural Networks*. Lawrance Erlbaum Associates, Inc.

Bengio, Y., Bengio, S., and Cloutier, J. (1991). Learning a synaptic learning rule. In *International Joint Conference on Neural Networks (IJCNN'91)*, volume 2. IEEE.

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. (2019). Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR'19)*.

Bottou, L. (2004). Stochastic Learning. In *Advanced lectures on machine learning*, pages 146–168. Springer.

Brazdil, P., Carrier, C. G., Soares, C., and Vilalta, R. (2008). *Metalearning: Applications to Data Mining.* Springer-Verlag Berlin Heidelberg.

Brazdil, P., van Rijn, J. N., Soares, C., and Vanschoren, J. (2022). *Metalearning: Applications to Automated Machine Learning and Data Mining.* Springer, 2nd edition.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chan, S. C., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A. K., Richemond, P. H., McClelland, J., and Hill, F. (2022). Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems.*

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. (2019). A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR'19).*

Chen, Y., Liu, Z., Xu, H., Darrell, T., and Wang, X. (2021). Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071.

Clavera, I., Nagabandi, A., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. (2019). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations (ICLR'19).*

Collins, L., Mokhtari, A., and Shakkottai, S. (2020). Why does maml outperform erm? an optimization perspective. *arXiv preprint arXiv:2010.14672.*

Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815.*

Deleu, T., Würfl, T., Samiei, M., Cohen, J. P., and Bengio, Y. (2019). Torchmeta: A Meta-Learning library for PyTorch. Available at: https://github.com/tristandeleu/pytorch-meta.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Ding, B. (2023). Understanding maml through its loss landscape. Master's thesis, Leiden University.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv preprint arXiv:1611.02779.*

Edwards, H. and Storkey, A. (2017). Towards a Neural Statistician. In *International Conference on Learning Representations (ICLR'17).*

Elsken, T., Staffler, B., Metzen, J. H., and Hutter, F. (2020). Meta-learning of neural architectures for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, pages 12365–12375.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2021). A brief review of domain adaptation. *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*, pages 877–894.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 1126–1135. PMLR.

Finn, C. and Levine, S. (2018). Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *International Conference on Learning Representations (ICLR'18)*.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. (2019). Online Meta-Learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, pages 1920–1930. JLMR.org.

Finn, C., Xu, K., and Levine, S. (2018). Probabilistic Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems 31*, pages 9516–9527.

Flennerhag, S., Rusu, A. A., Pascanu, R., Visin, F., Yin, H., and Hadsell, R. (2020). Meta-learning with warped gradient descent. In *International Conference on Learning Representations (ICLR'20)*.

Garcia, V. and Bruna, J. (2017). Few-Shot Learning with Graph Neural Networks. In *International Conference on Learning Representations (ICLR'17)*.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. M. A. (2018). Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80, pages 1704–1713.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. (2018). Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations (ICLR'18)*.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural Turing Machines. *arXiv preprint arXiv:1410.5401*.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. (2018). Meta-Reinforcement Learning of Structured Exploration Strategies. In *Advances in Neural Information Processing Systems 31*, pages 5302–5311. Curran Associates Inc.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30.

Hannan, J. (1957). Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, 2^{nd} edition.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Heggan, C., Budgett, S., Hospedales, T., and Yaghoobi, M. (2022). Metaaudio: A few-shot audio classification benchmark. In *International Conference on Artificial Neural Networks*, pages 219–230. Springer.

Hinton, G. E. and Plaut, D. C. (1987). Using Fast Weights to Deblur Old Memories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pages 177–186.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hochreiter, S., Younger, A. S., and Conwell, P. R. (2001). Learning to Learn Using Gradient Descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2023). Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*.

Huisman, M., van Rijn, J. N., and Plaat, A. (2021). A preliminary study on the feature representations of transfer learning and gradient-based meta-learning techniques. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*.

Iqbal, M. S., Luo, B., Khan, T., Mehmood, R., and Sadiq, M. (2018). Heterogeneous transfer learning techniques for machine learning. *Iran Journal of Computer Science*, 1(1):31–46.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, (ICLR'17)*.

Jankowski, N., Duch, W., and Grąbczewski, K. (2011). *Meta-Learning in Computational Intelligence*, volume 358. Springer-Verlag Berlin Heidelberg.

Jiang, W., Kwok, J., and Zhang, Y. (2022). Subspace learning for effective meta-learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 10177–10194. PMLR.

Kalai, A. and Vempala, S. (2005). Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307.

Kim, J., Lee, S., Kim, S., Cha, M., Lee, J. K., Choi, Y., Choi, Y., Cho, D.-Y., and Kim, J. (2018). Auto-meta: Automated gradient based meta learner search. *arXiv preprint arXiv:1806.06927*.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic gradient descent. In *International Conference on Learning Representations (ICLR'15)*.

Kirsch, L., Harrison, J., Sohl-Dickstein, J., and Metz, L. (2022). General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*.

Kirsch, L. and Schmidhuber, J. (2021). Meta learning backpropagation and improving it. In *Advances in Neural Information Processing Systems 34*, pages 14122–14134.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese Neural Networks for One-shot Image Recognition. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37. JMLR.org.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. (2011). One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, pages 2568–2573.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. `http://yann.lecun.com/exdb/mnist`. Accessed: 7-10-2020.

Lee, H.-y., Li, S.-W., and Vu, N. T. (2022). Meta learning for natural language processing: A survey. *arXiv preprint arXiv:2205.01500*.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. (2019). Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665.

Lee, Y. and Choi, S. (2018). Gradient-based meta-learning with learned layerwise metric and subspace. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, pages 2927–2936. PMLR.

Li, K. and Malik, J. (2018). Learning to Optimize Neural Nets. *arXiv preprint arXiv:1703.00441*.

Li, Z., Zhou, F., Chen, F., and Li, H. (2017). Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv preprint arXiv:1707.09835*.

Lian, D., Zheng, Y., Xu, Y., Lu, Y., Lin, L., Zhao, P., Huang, J., and Gao, S. (2019). Towards fast adaptation of neural architectures with meta learning. In *International Conference on Learning Representations (ICLR'19)*.

Liu, H., Simonyan, K., and Yang, Y. (2019). DARTS: Differentiable architecture search. In *International Conference on Learning Representations (ICLR'19)*.

Liu, Q. and Wang, D. (2016). Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in neural information processing systems 29*, pages 2378–2386. Curran Associates Inc.

Lu, J., Gong, P., Ye, J., and Zhang, C. (2020). Learning from very few samples: A survey. *arXiv preprint arXiv:2009.02653*.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, (ICLR'17)*.

Mangla, P., Kumari, N., Sinha, A., Singh, M., Krishnamurthy, B., and Balasubramanian, V. N. (2020). Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227.

Martens, J. and Grosse, R. (2015). Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32th International Conference on Machine Learning (ICML'15)*, pages 2408–2417. JMLR.org.

Metz, L., Maheswaranathan, N., Nixon, J., Freeman, D., and Sohl-Dickstein, J. (2019). Understanding and correcting pathologies in the training of learned optimizers. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, pages 4556–4565. PMLR.

Miconi, T., Rawal, A., Clune, J., and Stanley, K. O. (2019). Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In *International Conference on Learning Representations (ICLR'19)*.

Miconi, T., Stanley, K., and Clune, J. (2018). Differentiable plasticity: training plastic neural networks with backpropagation. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, pages 3559–3568. JLMR.org.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. (2018). A simple neural attentive meta-learner. In *International Conference on Learning Representations (ICLR'18)*.

Mitchell, T. M. (1980). The need for biases in learning generalizations. Technical Report CBM-TR-117, Rutgers University.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Munkhdalai, T. and Yu, H. (2017). Meta networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 2554–2563. JLMR.org.

Naik, D. K. and Mammone, R. J. (1992). Meta-neural networks that learn by learning. In *International Joint Conference on Neural Networks (IJCNN'92)*, volume 1, pages 437–442. IEEE.

Nichol, A., Achiam, J., and Schulman, J. (2018). On First-Order Meta-Learning Algorithms. *arXiv preprint arXiv:1803.02999*.

Olah, C. (2015). Understanding LSTM Networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 23-01-2023.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*.

Oreshkin, B., López, P. R., and Lacoste, A. (2018). Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems 31*, pages 721–731. Curran Associates Inc.

Pan, S. J. and Yang, Q. (2009). A Survey on Transfer Learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Park, E. and Oliva, J. B. (2019a). Meta-curvature. In *Advances in Neural Information Processing Systems 32*, NIPS'19, pages 3314–3324.

Park, E. and Oliva, J. B. (2019b). Meta-curvature. In *Advances in Neural Information Processing Systems 32*, pages 3309–3319.

Peng, Y., Flach, P. A., Soares, C., and Brazdil, P. (2002). Improved Dataset Characterisation for Meta-learning. In *International Conference on Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 141–152. Springer.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. (2020). Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International Conference on Learning Representations (ICLR'20).*

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-Learning with Implicit Gradients. In *Advances in Neural Information Processing Systems 32*, pages 113–124.

Ravi, S. and Larochelle, H. (2017). Optimization as a Model for Few-Shot Learning. In *International Conference on Learning Representations (ICLR'17).*

Ren, M., Ravi, S., Triantafillou, E., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. (2018). Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR'18).*

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. (2019). Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems 32*, pages 7957–7968.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. (2019). Meta-learning with latent embedding optimization. In *International Conference on Learning Representations (ICLR'19).*

Salakhutdinov, R., Tenenbaum, J., and Torralba, A. (2012). One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 195–206. JMLR Workshop and Conference Proceedings.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with Memory-augmented Neural Networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML'16)*, pages 1842–1850.

Schmidhuber, J. (1987). Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. Master's thesis, Technische Universität München.

Schmidhuber, J. (1993). A neural network that embeds its own meta-levels. In *IEEE International Conference on Neural Networks*, pages 407–412. IEEE.

Schmidhuber, J., Zhao, J., and Wiering, M. (1997). Shifting Inductive Bias with Success-Story Algorithm, Adaptive Levin Search, and Incremental Self-Improvement. *Machine Learning*, 28(1):105–130.

Schmidt, L. A. (2009). *Meaning and compositionality as statistical induction of categories and constraints*. PhD thesis, Massachusetts Institute of Technology.

Shi, B., Sun, M., Puvvada, K. C., Kao, C.-C., Matsoukas, S., and Wang, C. (2020). Few-shot acoustic event detection via meta learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80. IEEE.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems 28*.

Shyam, P., Gupta, S., and Dukkipati, A. (2017). Attentive Recurrent Comparators. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 3173–3181. JLMR.org.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Simon, C., Koniusz, P., Nock, R., and Harandi, M. (2020). On modulating the gradient for meta-learning. In *European Conference on Computer Vision*, pages 556–572. Springer.

Simons, T. (2022). The training of neural networks that can train neural networks. Master's thesis, Leiden University.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates Inc.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852.

Sun, P., Ouyang, Y., Zhang, W., and Dai, X. (2021). Meda: Meta-learning with data augmentation for few-shot text classification. In *IJCAI*, pages 3929–3935.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. (2019). Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. (2018). Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208. IEEE.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* MIT press, 2^{nd} edition.

Taylor, M. E. and Stone, P. (2009). Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research*, 10(7).

Thrun, S. (1998). Lifelong Learning Algorithms. In *Learning to learn*, pages 181–209. Springer.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., and Isola, P. (2020). Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539.*

Tieleman, T. and Hinton, G. (2017). Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical Report.*

Tokmakov, P., Wang, Y.-X., and Hebert, M. (2019). Learning Compositional Representations for Few-Shot Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6372–6381.

Triantafillou, E., Larochelle, H., Zemel, R., and Dumoulin, V. (2021). Learning a universal template for few-shot dataset generalization. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 10424–10433. PMLR.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. (2020). Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations (ICLR'20)*.

Ullah, I., Carrión-Ojeda, D., Escalera, S., Guyon, I., Huisman, M., Mohr, F., van Rijn, J. N., Sun, H., Vanschoren, J., and Vu, P. A. (2022). Meta-album: Multi-domain meta-dataset for few-shot image classification. *Advances in Neural Information Processing Systems*, 35:3232–3247.

Vanschoren, J. (2018). Meta-Learning: A Survey. *arXiv preprint arXiv:1810.03548*.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2014). OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, 15(2):49–60.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates Inc.

Vinyals, O. (2017). Talk: Model vs optimization meta learning. `http://metalearning-symposium.ml/files/vinyals.pdf`. Neural Information Processing Systems (NIPS'17); accessed 06-06-2020.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016). Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, pages 3637–3645.

Vuorio, R., Cho, D.-Y., Kim, D., and Kim, J. (2018). Meta Continual Learning. *arXiv preprint arXiv:1806.06928*.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, G., Luo, C., Sun, X., Xiong, Z., and Zeng, W. (2020a). Tracking by instance detection: A meta-learning approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6288–6297.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020b). Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys*, 53(3):1–34.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural

Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint arXiv:1609.08144*.

Yang, S., Liu, L., and Xu, M. (2021). Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations (ICLR'21)*.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. (2020). Meta-learning without memorization. In *International Conference on Learning Representations (ICLR'20)*.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. (2018). Bayesian Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems 31*, pages 7332–7342. Curran Associates Inc.

Younger, A. S., Hochreiter, S., and Conwell, P. R. (2001). Meta-learning with backpropagation. In *International Joint Conference on Neural Networks (IJCNN'01)*, volume 3. IEEE.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. (2019). Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *arXiv preprint arXiv:1910.10897*.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. (2019). Fast context adaptation via meta-learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, pages 7693–7702. PMLR.

Appendix A

Hyperparameter details for TURTLE

This is the appendix for Chapter 3, where we describe additional hyperparameter details.

A.1 Used hyperparameters

For all techniques mentioned below, we performed meta-validation after every 2,500 training tasks. The best-resulting configuration was evaluated at meta-test time.

For sine wave regression, we use the same base-learner network as Finn et al. (2017), i.e., a fully-connected feed-forward network consisting of a single input node followed by two hidden layers with 40 ReLU nodes each and a final single-node output layer.

For few-shot image classification problems, we use the same base-learner network as used by Snell et al. (2017) and Chen et al. (2019). This network is a stack of four identical convolutional blocks. Each block consists of 64 convolutions of size 3×3, batch normalization, a ReLU nonlinearity, and a 2D max-pooling layer with a kernel size of 2. The resulting embeddings of the $84 \times 84 \times 3$ input images are flattened and fed into a dense layer with N nodes (one for every class in a task). The base-learner is trained to minimize the cross-entropy loss on the query set, conditioned on the support set.

Transfer learning baselines Note that these models (TrainFromScratch, finetuning, baseline++) pre-trained on minibatches of size 16 sampled from the joint data obtained by merging all meta-training tasks. At test time, they were trained for 100 steps on mini-batches of size 4 sampled from new tasks following Chen et al. (2019). Every 25 steps, we evaluated their performance on the entire support set to select the best configuration to test on the query set.

LSTM meta-learner For selecting the hyperparameters of the LSTM meta-learner[1], we followed Ravi and Larochelle (2017). That is, we use a 2-layer architecture, and Adam as meta-optimizer with a learning rate of 0.001. The batch size was set equal to the size of the task. Meta-gradients were

[1]Used code: `https://github.com/markdtw/meta-learning-lstm-pytorch`.

clipped to have a norm of at most 0.25, following. The meta-network receives four inputs obtained by preprocessing the loss and gradients using in similar fashion to Andrychowicz et al. (2016) and Ravi and Larochelle (2017). On miniImageNet and CUB, the LSTM optimizer is set to perform 12 updates per task when the number of examples per class is $k = 1$ and 5 updates when $k = 5$.

MAML Again, we follow Finn et al. (2017) for selecting the hyperparameters, except for the meta-batch size on sine wave regression as we found it not to help performance. This means that the inner learning rate was set to 0.01 and the outer learning rate to 0.001, with Adam as meta-optimizer. These settings hold for both sine wave regression and image classification. When $T > 1$, we use gradient value clipping with a threshold of 10. On image classification, MAML was set to optimize the initial parameters based on $T = 5$ update steps, but an additional 5 steps were made afterwards to further increase the performance. Moreover, we used a meta-batch size of 4 and 2 for 1- and 5-shot image classification respectively.

TURTLE We performed many experiments with the hyperparameters of TURTLE on sine wave regression. Here, we only report the settings that were found to give the best performance, which were also used on the image classification problems. That is, the meta-network consists of 5 hidden layers of 20 nodes each. Every hidden node is followed by a ReLU nonlinearity. The input consists of a raw gradient, a historical real-valued number indicating the moving average of the previous input gradients with a (with a beta decay of 0.9), and a time step integer $t \in \{0, ..., T-1\}$. The output layer consists of a single node which corresponds to the proposed weight update. For training, we used meta-batches of size 2. Additionally, TURTLE maintains a separate learning rate for all weights in the base-learner network. Lastly, TURTLE uses second-order gradients and Adam as meta-optimizer with a learning rate of 0.001.

Appendix B

Additional experimental results for OP-LSTM

This is the appendix for Chapter 5, where we present additional experimental results.

B.1 Sine wave regression: additional results

We also performed an experiment to investigate the effect of the input representation on the performance of the plain LSTM approach (proposed by Younger et al. (2001); Hochreiter et al. (2001)) on the 5-shot sine wave regression performance. The experimental setting follows the setup described in Section 5.5.1. For every input format, we performed hyperparameter tuning with the same randomly sampled hyperparameter configurations using Table B.2. The performances of the best validated models per input format are displayed in Table B.1. The best performance is obtained by feeding the current input, previous target, and the previous prediction into the LSTM, although the differences with other inputs are small.

Table B.1: The influence of different input information on the performance of the LSTM on 5-shot sine wave regression. 95% confidnce intervals are displayed as $\pm x$.

Input $\mathbf{x}_t$	Prev target y_{t-1}	Prev pred $\hat{y}_{t-1}$	Prev error e_{t-1}	5-shot MSE
✓	✓			0.04 ± 0.002
✓	✓	✓		**0.03** ± 0.002
✓	✓		✓	0.05 ± 0.004
✓	✓	✓	✓	0.06 ± 0.011

B.2 Hyperparameter tuning

B.2.1 Permutation invariance experiments

For the permutation invariance experiments on few-shot sine wave regression, we sampled 20 random configurations for the plain LSTM from the distributions displayed in Table B.2 and validated their performance on 5-shot ($k = 5$) sine-wave regression. We selected the best configuration and evaluated it on the meta-test tasks,

Table B.2: The used ranges and distributions for tuning the hyperparameters with random search for sine wave regression.

Hyperparameter	Range
Number of layers	Uniform({1,2,3,4})
Hidden dimensions	Uniform({1,3,8,20,40})
Meta-batch size	Uniform({1,2,3,4})
Learning rate	LogUniform(1e-5, 4e-2)
Unroll steps	Uniform({1,2,...,14})

For Omniglot, we performed random search with a function evaluation budget of 100, with a fixed learning rate of 0.001. The architecture of the plain LSTM with sequential data processing was sampled uniformly at random from {1024-512-256-128-64,2048-1024-512-128-64,2048-1024-512-256-128,1024-600-400-200-92,1024-512-512-256-128-64,1024-512-512-256-256-128-64, 612-400-256-128-64,1024-1024-1024-512-256-128-64,2048-1024-512-180-100,1024-580-280-160-80,256-128-64, 512-256-128-64,128-64-64-64,256-128-64,512-256-64,256-128-100,128-64-64-64-64,64-64-64-64,50-50}, the number of passes over the support data T was sampled uniformly at random from $\{1, 2, \ldots, 10\}$, and the meta-batch size from $\{1, 2, \ldots, 32\}$. We used the best hyperparameter configuration of the sequential plain LSTM for the plain LSTM with batching to compare the differences in performance.

B.2.2 Omniglot

For the **plain LSTM** approach, we used the best hyperparameter configuration found for the permutation invariance experiments.

For **OP-LSTM**, we performed a grid search, varying the meta-batch size within {1,4,8,16,32}, the architecture of the coordinate-wise LSTM within {20-1, 10-10-1, 40-5, 40-20-1, 20-20-20-5} (note that the last element is always 1 because it operates per coordinate), and the number of passes over the support set within {1,3,5,10}.

Detailed learning curves for the plain LSTM on Omniglot Here, we show the validation learning curves of the sequential LSTM and the LSTM which uses batching to complement the results displayed in Section 5.5.1. Figure B.1 displays the validation learning curves of the LSTM with batch data ingestion (top row) and the LSTM with sequential data processing (bottom row). As we can see, batching increases the stability of the training process and makes the LSTM less sensitive to the random initialization, as every run succeeds to reach convergence in contrast to the sequential LSTM.

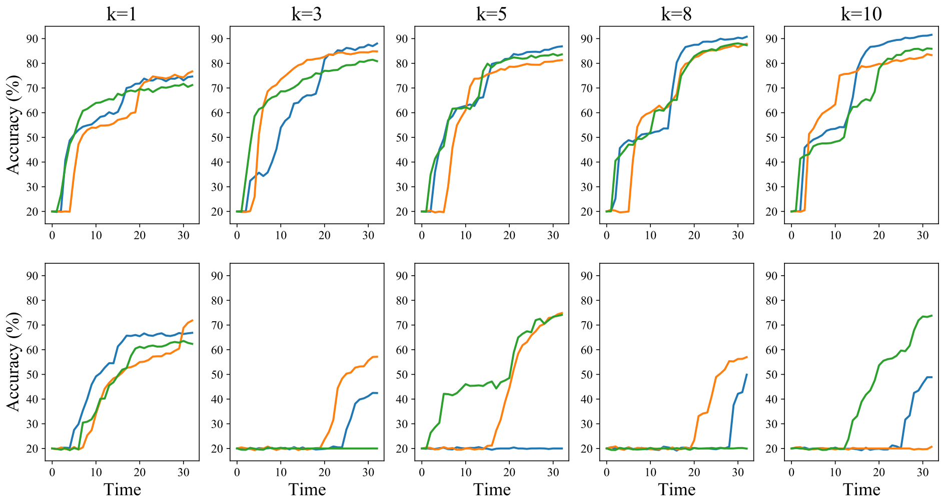

Figure B.1: The mean validation accuracy of the LSTM over time on Omniglot for every of the three different runs, for different numbers of examples per class k. **Top row:** LSTM with batching (mean-pooling). **Bottom row:** LSTM with sequential data ingestion. As we can see, batching improves the stability of the training process.

B.2.3 miniImageNet and CUB

For **plain LSTM**, we used random search with a budget of 130 function evaluations, the meta-batch size was sampled uniformly between 1 and 48, the number of layers between 1 and 4, the hidden size log-uniformly between 32 and 3200, and the number of passes T over the support dataset uniformly between 2 and 9.

For **OP-LSTM**, we performed the same grid search as for Omniglot. We use the best found hyperparameters for both methods on miniImageNet also on CUB.

We also measured the running times of the techniques on miniImageNet and CUB, as shown in Table B.3. We note that the running times may be affected by the server's load and thus can only give a rough estimation of the required amount of compute time. As we can see, the plain LSTM is the slowest method, despite achieving random performance on miniImageNet. OP-LSTM, in contrast, is more efficient.

B.3 Robustness to random seeds

Here, we investigate the robustness of the investigated methods to the random seed for the few-shot image classification experiments performed in Section 5.5.3. We perform th Instead of computing the confidence intervals over the performances of all test tasks for all seeds, we now compute the confidence interval over the mean test performance per run. As we perform three runs per method, we compute the confidence intervals over three observations per method. Note that the mean performance does not change as taking the mean of the three means will be equivalent (as the means

Table B.3: Mean running times on 5-way miniImageNet and CUB classification over 3 runs. All methods used a Conv-4 backbone as a feature extractor. "xh$y$min" means x hours and y minutes. The "-" indicates that the method did not finish within 2 days of running time.

		miniImageNet		CUB	
Technique	params	1-shot	5-shot	1-shot	5-shot
MAML	121 093	13h9min	12h1min	26h57min	17h39min
Warp-MAML	231 877	12h25min	12h30min	13h6min	12h48min
SAP	412 852	5h40min	11h14min	7h11min	11h17min
ProtoNet	121 093	4h14min	5h6min	31h18min	38h46min
LSTM	55 879 349	40h14min	46h47min	-	-
OP-LSTM (ours)	141 187	4h50min	5h31min	31h58min	40h8min

are based on an equal number of task performances).

B.4 Within-domain

Here, we present additional results for the conducted within-domain image classification experiments.

Omniglot The mean test performance and confidence intervals over the random seeds for Omniglot image classification are shown in Table B.4. As we can see, the confidence intervals are higher than in previous experiments because the intervals are computed over 3 observations instead of 1800 individual test task performances (600 per run). As we can see, the LSTM is unstable, supporting the hypothesis that the optimization problem is difficult. OP-LSTM, on the other hand, is less sensitive to the chosen random seed and has a stability that is comparable to that of MAML.

Table B.4: The mean test accuracy (%) on 5-way Omniglot classification across 3 different runs. The 95% confidence intervals, computed over the mean performances of the 3 different random seeds, are displayed as $\pm x$. The plain LSTM is outperformed by MAML. All methods (except LSTM) used a fully-connected feed-forward classifier.

Technique	parameters	1-shot	5-shot
MAML	247 621	84.1 $\pm$ 3.10	**93.5** $\pm$ 0.70
ProtoNet	247 621	83.6 $\pm$ 0.52	93.4 $\pm$ 1.48
LSTM	13 530 097	72.6 $\pm$ 3.87	84.8 $\pm$ 6.12
OP-LSTM (ours)	249 167	**84.3** $\pm$ 3.18	91.8 $\pm$ 0.70

MiniImageNet and CUB The mean test performance and confidence intervals over the random seeds for miniImageNet and CUB image classification are shown in Table B.5. In contrast to what we observed on Omniglot, the LSTM is now more stable. This is caused by the fact that it consistently fails to learn a learning algorithm that performs better than random guessing, and thus performs stably at chance level.

Table B.5: Meta-test accuracy scores on 5-way miniImageNet and CUB classification over 3 runs. The 95% confidence intervals, computed over the mean performances of the 3 different random seeds, are displayed as ± x. All methods used a Conv-4 backbone as a feature extractor. The "-" indicates that the method did not finish within 2 days of running time.

		miniImageNet		CUB	
Technique	params	1-shot	5-shot	1-shot	5-shot
MAML	121 093	48.6 ± 4.00	63.0 ± 0.33	57.5 ± 0.83	**74.8** ± 2.10
Warp-MAML	231 877	50.4 ± 2.58	65.6 ± 0.98	59.6 ± 2.15	74.2 ± 2.51
SAP	412 852	**53.0** ± 3.71	67.6 ± 0.47	**63.5** ± 6.24	73.9 ± 1.57
ProtoNet	121 093	50.1 ± 4.06	65.4 ± 2.84	50.9 ± 2.35	63.7 ± 0.47
LSTM	55 879 349	20.2 ± 0.60	19.4 ± 0.47	-	-
OP-LSTM (ours)	141 187	51.9 ± 2.52	**67.9** ± 2.40	60.2 ± 1.58	73.1 ± 1.57

Table B.6: Average cross-domain meta-test accuracy scores over 5 runs using a Conv-4 backbone. Techniques trained on tasks from one data set and were evaluated on tasks from another data set. The 95% confidence intervals, computed over the mean performances of the 3 different random seeds, are displayed as ± x. The "-" indicates that the method did not finish within 2 days of running time.

	MIN → CUB		CUB → MIN	
	1-shot	5-shot	1-shot	5-shot
MAML	37.9 ± 2.22	53.6 ± 0.67	31.1 ± 1.19	45.8 ± 2.06
Warp-MAML	42.0 ± 0.85	56.9 ± 4.16	31.1 ± 1.59	41.3 ± 1.37
SAP	41.5 ± 3.72	58.0 ± 1.79	33.3 ± 2.33	47.1 ± 1.28
ProtoNet	39.7 ± 4.11	56.0 ± 4.89	31.7 ± 0.20	45.3 ± 1.84
LSTM	20.1 ± 0.77	20.0 ± 0.40	-	-
OP-LSTM (ours)	**42.3** ± 1.90	**58.5** ± 1.49	**35.8** ± 2.98	**49.0** ± 0.80

B.5 Cross-domain

Lastly, we compute the confidence intervals in cross-domain settings and display the results in Table B.6. Again, the LSTM is a stable random guesser. The other algorithms are less stable, but do yield a better performance. We cannot observe a general pattern of stability in the sense that one algorithm is consistently more stable than others.

Appendix C

Additional experimental results for SAP

In this appendix for Chapter 6, we show additional experimental results on few-shot image classification.

C.1 Validation of re-implementation

	1-shot		5-shot	
	Reported	Local Repr	Reported	Local repr
MAML	48.7 ± 1.8	48.0 ± 0.8	63.2 ± 0.9	64.4 ± 0.4
T-Net	50.9 ± 1.8	48.9 ± 0.8	-	65.3 ± 0.4
MT-Net	51.7 ± 1.8	48.5 ± 0.8	-	63.0 ± 0.4
Warp-MAML*	-	49.5 ± 0.8	-	63.9 ± 0.4
SAP (ours)	-	**51.6** ± 0.8	-	**65.9** ± 0.4

Table C.1: Mean meta-test accuracy scores on 5-way miniImageNet classification over 5 runs using a Conv-4 backbone with 32 channels. The 95% confidence intervals are displayed as ± x. * Flennerhag et al. (2020) only reported the performance of Warp-MAML with 128 feature maps per convolutional block instead of 32, as displayed in the table.

We re-implemented the baselines to ensure a fair comparison in the used setting, and because the code of Warp-MAML has not been made available for other researchers. To verify our re-implementations of the baselines (T-Net, MT-Net, and Warp-MAML), we compare the reported performances to the ones that we obtain. The results of the image classification experiments are displayed in Table C.1. As we can see, there are minor differences between the reported performances and our local reproduction of their results. Also with the original code of T-Net and MT-Net, we were unable to reproduce their results. Other people have encountered similar issues reproducing the reported numbers of meta-learning techniques, including MAML, T-Net, and MT-Net.[1]

[1]There is an open issue on the GitHub repository of MT-Net about the inability to reproduce their reported results

C.2 Cross-domain few-shot image classification

In Table C.2, we show the cross-domain few-shot learning classification results when using 64 channels with the Conv-4 backbone. Also in this case, SAP outperforms other tested baselines. We also note that the performance of SAP is improved when using 64 channels compared with 32 (see Section 6.5.5).

	MIN → CUB		Tiered → CUB	
	1-shot	5-shot	1-shot	5-shot
MAML	37.1 ± 0.3	53.7 ± 0.3	38.8 ± 0.3	56.8 ± 0.3
T-Net	38.3 ± 0.3	OOM	39.9 ± 0.3	OOM
MT-Net	37.3 ± 0.3	OOM	39.1 ± 0.3	OOM
Warp-MAML	40.7 ± 0.3	56.2 ± 0.3	42.5 ± 0.3	58.9 ± 0.3
SAP (ours)	41.6 ± 0.3	57.8 ± 0.3	43.3 ± 0.3	64.3 ± 0.3

Table C.2: Average cross-domain meta-test accuracy scores over 5 runs using a 64-channel Conv-4 backbone. Techniques trained on tasks from one data set were evaluated on tasks from another data set. The 95% confidence intervals are displayed as ± x.

C.3 The effect of hard pruning

Table C.3 displays the effect of hard pruning when using 64 channels instead of 32. As we can see, hard pruning is slightly beneficial, but again, not significantly.

	miniImageNet		tieredImageNet	
	1-shot	5-shot	1-shot	5-shot
No pruning	52.8 ± 0.8	67.4 ± 0.4	54.5 ± 0.8	71.3 ± 0.4
Top-1	52.8 ± 0.8	**67.6** ± 0.4	**55.1** ± 0.8	**72.7** ± 0.4
Top-2	**52.9** ± 0.8	**67.6** ± 0.4	54.1 ± 0.8	**72.7** ± 0.4
Top-3	52.6 ± 0.8	67.4 ± 0.4	55.0 ± 0.8	72.4 ± 0.4

Table C.3: Mean meta-test accuracy scores on 5-way miniImageNet and tieredImageNet classification with 95% confidence intervals computed over 5 different runs. We used a Conv-4 backbone with 64 channels for these results.

C.4 The learned subspaces for image classification

Figure C.1 displays the learned activation strengths of SAP on 5-way 1-shot miniImageNet using Conv-4 with 64 channels. Similar patterns are observed for the 32-channel case.

---

on miniImageNet. See `https://github.com/yoonholee/MT-net/issues/5`. Other researchers such as Antoniou et al. (2019) have also reported issues reproducing MAML.

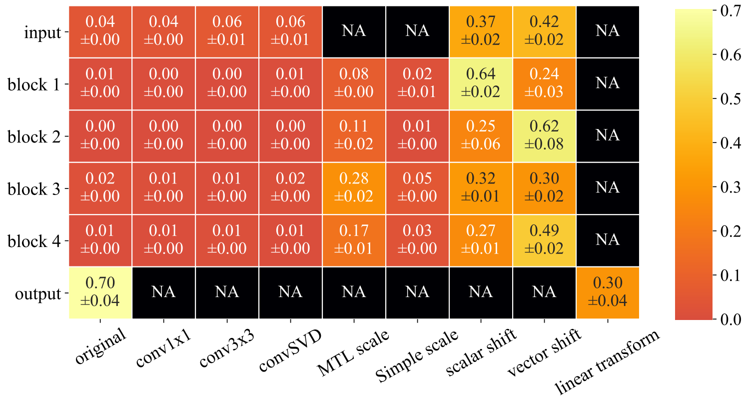

Figure C.1: The importance of the different subspaces/operations in SAP on 5-way 1-shot miniImageNet using Conv-4 with 64 channels. The results are averaged across 5 runs with different random seeds and the standard deviations are shown as $\pm x$. NA entries indicate that these operations were not in the candidate pool for that layer. Simple scalar shift and vector shift operations obtain the highest activation strengths throughout the convolutional network.