A Philosophies behind SAMA & Scalable Meta Learning

Here, we additionally discuss several important design principles and philosophies behind scalable meta learning and SAMA in a Q&A format.

Q. Why do we study scalable meta learning?

A. Richard Sutton points out in his article "The Bitter Lesson" [59] that machine learning algorithms that stand the test of time are ones that continue to *scale gracefully with the increased computation budget* (*i.e*., scalable algorithms). Given that meta learning is an important topic in machine learning with many applications including data optimization [53], hyperparameter optimization [17], few-shot learning [15, 50], and adversarial learning [44], it was a natural call for us to investigate the scalability of meta learning algorithms following the spirit of "The Bitter Lesson". Interestingly, such a focus on the scalability of meta learning algorithms distinguishes our work from most other meta learning works, in which the typical focus is to improve the overall performance of meta learning algorithms *under a limited computation budget* (usually bounded by a single GPU).

Q. What are the design principles behind scalable meta learning?

A. The increased computation budget powered by hardware advancements (*e.g*., Moore's law) has evolved a new ecosystem of large models and datasets in machine learning over time, which involves both systems and algorithms components. For example, to efficiently leverage the increased computation for large-scale learning, diverse systems techniques, such as data/model/pipeline parallelism have been developed [29, 35, 56]. At the same time, researchers have devised various algorithms that are highly effective for large-scale learning, such as backpropagation [54], skip connections [28], Adam optimizer [32], self-attention [61], etc. Accordingly, in addition to guaranteeing memory/compute efficiency for the scalability, our major design principle for scalable meta learning was to *ensure compatibility with existing systems and algorithms* in the large-scale learning ecosystem.

Systems compatibility Given that a great deal of systems support in machine learning, such as communication-computation overlap [35], has been developed for first-order gradient methods, avoiding explicit computations of higher-order gradient information including Hessian-vector products was an important design principle in SAMA. Even though we mostly explored distributed training in this work, SAMA is also compatible with other system features such as half-precision training and activation checkpointing, which could further improve memory efficiency.

Algorithms compatibility While there exist several meta learning algorithms that avoid the computation of higher-order gradient information [34, 37, 65], many of these algorithms either assume the use of a naive SGD update rule or devise specific update rules tailored to their own algorithms at the base level, significantly hampering their algorithm compatibility. In contrast, SAMA allows for the use of arbitrary optimizers at the base level via the algorithmic adaptation.

B Experiment Details

In this section, we discuss various experiment details such as hyperparameters, baselines, and compute resources used for our experiments in Section 4. For reproducible research, we plan to release our experiment codes and SAMA implementation in the future (at the moment, codes are available in the supplementary material).

B.1 Noisy Finetuning of Large Language Models

Hyperparameters We ran training for 1000 iterations on TREC/SemEval/IMDB/ChemProt/Yelp/AGNews datasets from the WRENCH benchmark [67], with a batch size of 32, a weak supervision algorithm of majority voting, and the hyperparameters in Table 4 below.

	model	optimizer	init_lr	lr_scheduler	wdecay	dataset	unroll step	SAMA α
Base	BERT-base	Adam	1e-5	cosine	0	WRENCH train set (with majority voting)	10	1.0
Meta (Reweight)	2-layer MLP	Adam	1e-5	None	0	WRENCH dev set	N/A	N/A
Meta (Correct)	2-layer MLP	Adam	1e-5	None	0	WRENCH dev set	N/A	N/A

Table 4: Hyperparameters for *noisy finetuning of large language models* experiments.

Baselines We adopted naive finetuning and self-training (*i.e.*, COSINE [66]) approaches from the original WRENCH benchmark paper [67] as our baseline.

Compute Resources We used 1 NVIDIA RTX 2080Ti GPU for the main experiment, and 4 NVIDIA Tesla V100 GPUs for the throughput-memory analysis in Table 2 and Figure 1.

B.2 Continued Pretraining of Large Language Models

Hyperparameters We ran training for 100 epochs with a batch size of 16, a maximum sequence length of 256, and the hyperparameters in Table 5 below.

	model	optimizer	init_lr	lr_scheduler	wdecay	dataset	unroll step	SAMA α
Base (Downstream)	RoBERTa-base	Adam	2e-5	linear decay + warmup linear (warmup proportion 0.6)	0	train split of ChemProt/HyperPartisan/ ACL-ARC/SciERC	10	0.3
Base (Auxiliary)	RoBERTa-base	Adam	2e-5	linear decay + warmup linear (warmup proportion 0.6)	0	train split of ChemProt/HyperPartisan/ ACL-ARC/SciERC	10	0.3
Meta	2-layer MLP	Adam	1e-5	None	0	train split of ChemProt/HyperPartisan/ ACL-ARC/SciERC	N/A	N/A

Table 5: Hyperparameters for *continued pretraining of large language models* experiments.

Baselines We adopt DAPT [25] and TARTAN-MT [11] as our baselines for this experiment. In detail, DAPT [25] performs additional masked language model pretraining on domain-specific data on top of the pretrained RoBERTa-base model and then finetunes the model on the downstream text classification task. We follow [25] (see Table 14 in the original paper) for setting downstream finetuning hyperparameters. Alternatively, TARTAN-MT [11] performs masked language modeling with task specific data and downstream text classification training simultaneously in a multitask fashion through two different heads.

Compute Resources We used 1 NVIDIA Tesla V100 GPU for the main experiment, and 1 NVIDIA RTX A6000 GPU for the "memory vs model-size analysis" in Figure 1.

B.3 Scale-Agnostic Efficient Data Pruning

Hyperparameters We ran meta learning for 30 epochs with a batch size of 256 and the configuration shown in Table 6 below. After pruning data based on the meta learning result, we ran ImageNet-1k training for 120 epochs with learning rate decayed by 10 at epochs [40, 80] following the setup in DynaMS [62].

	model	optimizer	init_lr	lr_scheduler	wdecay	dataset	unroll step	SAMA α
Base	ResNet-50	SGD	1e-1	None	1e-4	ImageNet-1k train set	2	1.0
Meta	2-layer MLP	Adam	1e-5	None	0	ImageNet-1k train set	N/A	N/A

Table 6: Hyperparameters for *ImageNet-1k data pruning* experiments

For the CIFAR-10 data pruning experiment, we ran meta learning for 50 epochs with the batch size of 128, and configuration in Table 7 below. After pruning the data based on the meta learning result, we ran CIFAR-10 training for 200 epochs with the cosine learning rate decay schedule following the setup in DeepCore [23].

	model	optimizer	init_lr	lr_scheduler	wdecay	dataset	unroll step	SAMA α
Base	ResNet-18	SGD	1e-1	None	5e-4	CIFAR-10 train set	2	1.0
Meta	2-layer MLP	Adam	1e-5	None	0	CIFAR-10 train set	N/A	N/A

Table 7: Hyperparameters for *CIFAR-10 data pruning* experiments

Baselines We adopt EL2N [46], GraNd [46], DynaMS [62] as our baselines for the ImageNet-1k experiments and GraNd [46], forgetting [60], margin [8] for the CIFAR-10 experiments. In detail, EL2N/GraND [46] respectively select samples with large L2-loss/gradient-norm values, forgetting [46] chooses samples that are frequently forgotten during training, and margin [8] chooses samples with least confidence. While these baselines are considered *static pruning*, DynaMS [62] falls under the category of dynamic pruning where data to be pruned change during training. Dynamic pruning may see the whole training data across different epochs, making a fair comparison difficult. Surprisingly, despite being a static pruning algorithm, SAMA-based data pruning still achieves a better performance than DynaMS.

Compute Resources We used 4 NVIDIA Tesla V100 GPUs for Imagenet-1k data pruning meta learning experiments and 1 NVIDIA RTX 2080Ti GPU for CIFAR-10 experiments.

Additional Information We measured the uncertainty $\mathcal{U}$ via the difference between the predictions of the current model and the exponentially-moving-averaged model.

C Algorithmic Adaptation for Adam Optimizer

Since the Adam optimizer [32] has been the most popular optimizer to train large models, exemplified by Transformers [68], here we provide the adaptation matrix for Adam. We denote first and second moments of the gradient in Adam as m and v respectively, and the learning rate as γ.

$$\begin{aligned}
\frac{\partial u_{adam}}{\partial g} &= \frac{\partial u}{\partial g}\left(\gamma\frac{\beta_1 m + (1-\beta_1)g}{\sqrt{\beta_1 v + (1-\beta_1)g^2} + \epsilon}\right) \\
&= \gamma\frac{(1-\beta_1)\beta_2 v - (1-\beta_1)\beta_2 mg + (1-\beta_1)\epsilon\sqrt{\beta_1 v + (1-\beta_1)g^2}}{\sqrt{\beta_1 v + (1-\beta_1)g^2}\left(\sqrt{\beta_1 v + (1-\beta_1)g^2} + e\right)^2} \\
&\approx \gamma\frac{(1-\beta_1)\beta_2 v - (1-\beta_1)\beta_2 mg}{\sqrt{\beta_1 v + (1-\beta_1)g^2}\left(\sqrt{\beta_1 v + (1-\beta_1)g^2} + e\right)^2} \qquad (\text{because } \epsilon \ll 1)
\end{aligned}$$

Adaptation matrices can be similarly derived for other adaptive optimizers.

D The Effect of Scaling in Model-Agnostic Meta Learning

Since the inception of MAML [15], a myriad of algorithms have been proposed to improve few-shot image classification while assuming a fixed network architecture. In contrast, here we shift our focus from the algorithm to the scale, and propose to study the following question: "Leveraging the compute/memory efficiency of SAMA, can we improve the few-shot generalization capability by scaling up the network size?". Since SAMA is a variant of implicit differentiation, we closely follow the experiment setup in iMAML [50], where proximity to the initialization weights is explicitly enforced by L_2-regularization. The major difference is that iMAML uses a conjugate-gradient-based method, which requires second-order gradient information to compute meta gradients, while we adopt SAMA to achieve improved scaling to larger networks with its superior memory/compute efficiency. We conduct preliminary experiments on the Omniglot 20-way 1-/5-shot tasks with the basic 4-layer CNN architecture, while varying the width (hidden size) of the networks to study the effect of the model size on the few-shot classification accuracy. The experiment results are provided in Figure 4 below.

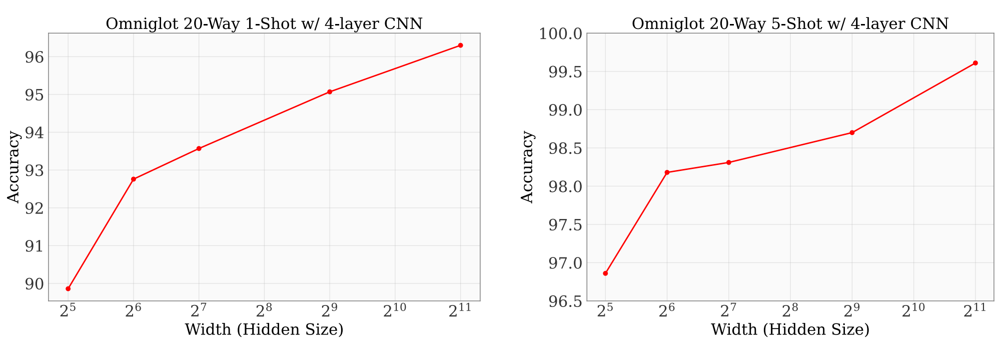

Figure 4: Few-shot image classification accuracy on Omniglot 20-way 1-/5-shot tasks with varying network sizes.

Interestingly, we observe that the increased model size leads to consistent improvements in few-shot classification accuracy. The important question following this observation is "can we apply scaling laws [31] from other tasks (*e.g.*, language modeling) to general meta learning beyond few-shot image classification?" Since meta learning involves two optimization problems (meta and base) unlike traditional machine learning problems, it is as of now unclear how to define the general concept of "scale" in terms of both model and dataset sizes. We expect that further research in this direction would be critical in systematically studying scalable meta learning.

References

[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019.

[3] Sébastien MR Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for meta-learning research. *arXiv preprint arXiv:2008.12284*, 2020.

[4] Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Sang Keun Choe, Willie Neiswanger, Pengtao Xie, and Eric Xing. Betty: An automatic differentiation library for multilevel optimization. *arXiv preprint arXiv:2207.02849*, 2022.

[7] Ross M Clarke, Elre T Oldewage, and José Miguel Hernández-Lobato. Scalable one-pass optimisation of high-dimensional weight-update hyperparameters by implicit differentiation. *arXiv preprint arXiv:2110.10461*, 2021.

[8] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

[9] Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A Meta-Learning library for PyTorch, 2019. Available at: https://github.com/tristandeleu/pytorch-meta.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Lucio M Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. Should we be pre-training? an argument for end-task aware training as an alternative. *arXiv preprint arXiv:2109.07437*, 2021.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[16] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017.

[17] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[18] Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. Jfb: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6648–6656, 2022.

[19] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023.

[20] Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR, 2020.

[21] Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

[22] Denis Gudovskiy, Luca Rigazio, Shun Ishizaka, Kazuki Kozuka, and Sotaro Tsukizawa. Autodo: Robust autoaugment for biased data with label noise via scalable probabilistic implicit differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16601–16610, 2021.

[23] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I*, pages 181–195. Springer, 2022.

[24] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *Advances in neural information processing systems*, 31, 2018.

[25] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[26] Ryuichiro Hataya and Makoto Yamada. Nyström method for accurate and scalable implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 4643–4654. PMLR, 2023.

[27] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Meta approach to data augmentation optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2574–2583, 2022.

[28] 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.

[29] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

[30] Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

[31] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[34] Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert Nowak. A fully first-order method for stochastic bilevel optimization. *arXiv preprint arXiv:2301.10945*, 2023.

[35] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

[36] Renjie Liao, Yuwen Xiong, Ethan Fetaya, Lisa Zhang, KiJung Yoon, Xaq Pitkow, Raquel Urtasun, and Richard Zemel. Reviving and improving recurrent back-propagation. In *International Conference on Machine Learning*, pages 3082–3091. PMLR, 2018.

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

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[39] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

[40] Jelena Luketina, Mathias Berglund, Klaus Greff, and Tapani Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*, pages 2952–2960. PMLR, 2016.

[41] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

[42] Luke Metz, James Harrison, C Daniel Freeman, Amil Merchant, Lucas Beyer, James Bradbury, Naman Agrawal, Ben Poole, Igor Mordatch, Adam Roberts, et al. Velo: Training versatile learned optimizers by scaling up. *arXiv preprint arXiv:2211.09760*, 2022.

[43] Luke Metz, Niru Maheswaranathan, Jeremy Nixon, Daniel Freeman, and Jascha Sohl-Dickstein. Understanding and correcting pathologies in the training of learned optimizers. In *International Conference on Machine Learning*, pages 4556–4565. PMLR, 2019.

[44] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[46] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

[47] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

[48] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pages 737–746. PMLR, 2016.

[49] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[50] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[51] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[52] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.

[53] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[54] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[55] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[56] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[57] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

[58] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[59] Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1), 2019.

[60] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[62] Jiaxing Wang, Yong Li, Jingwei Zhuo, Xupeng Shi, WEIZHONG ZHANG, Lixing Gong, Tong Tao, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. DynaMS: Dyanmic margin selection for efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[63] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11293–11302, 2019.

[64] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *arXiv preprint arXiv:2302.03169*, 2023.

[65] Mao Ye, Bo Liu, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. *arXiv preprint arXiv:2209.08709*, 2022.

[66] Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *arXiv preprint arXiv:2010.07835*, 2020.

[67] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. Wrench: A comprehensive benchmark for weak supervision. *arXiv preprint arXiv:2109.11377*, 2021.

[68] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

[69] Miao Zhang, Steven W Su, Shirui Pan, Xiaojun Chang, Ehsan M Abbasnejad, and Reza Haffari. idarts: Differentiable architecture search with stochastic implicit gradients. In *International Conference on Machine Learning*, pages 12557–12566. PMLR, 2021.

[70] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11053–11061, 2021.

[71] Nicolas Zucchet, Simon Schug, Johannes Von Oswald, Dominic Zhao, and João Sacramento. A contrastive rule for meta-learning. *Advances in Neural Information Processing Systems*, 35:25921–25936, 2022.