ACCELERATING NEURAL SELF-IMPROVEMENT VIA BOOTSTRAPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot learning with sequence-processing neural networks (NNs) has recently attracted a new wave of attention in the context of large language models. In the standard N-way K-shot learning setting, an NN is explicitly optimised to learn to classify unlabelled inputs by observing a sequence of NK labelled examples. This pressures the NN to learn a learning algorithm that achieves maximum performance, given the limited number of training examples. Here we study an auxiliary loss that encourages *further* acceleration of few-shot learning, by applying recently proposed bootstrapped meta-learning to NN few-shot learners: we optimise the K-shot learner to match its own performance achievable by observing *more* than NK examples, using *only* NK examples. Promising results are obtained on the standard Mini-ImageNet dataset.

1 INTRODUCTION & MOTIVATION

Since the seminal works by Hochreiter et al. (2001); Younger et al. (1999); Cotter & Conwell (1991; 1990), many sequence processing neural networks (NNs)—including recurrent NNs (RNNs) e.g., in Santoro et al. (2016) but also Transformer variants (Vaswani et al., 2017) as in Mishra et al. (2018)—have been trained as a meta-learner (Schmidhuber, 1987; 1992) that learns to learn to solve a task by observing sequences of training examples (i.e., pairs of inputs and their labels). A popular application of such an approach is to build a sample efficient *few-shot learner* capable of adaption to a new (but related) task from few training examples or demonstrations (Finn et al., 2017), which typically achieves respectable performance in the standard benchmarks for few-shot learning (e.g., image classification), see e.g., Mishra et al. (2018); Munkhdalai & Yu (2017). More recently, such "on-the-fly" learning capability of sequence processing NNs has attracted broader interests in the context of large language models (LLMs; Radford et al. (2019)). In fact, the task of language modelling itself has a form of *sequence processing with error feedback* (essential for meta-learning (Schmidhuber, 1990)): the correct label to be predicted is fed to the model with a delay of one time step in an auto-regressive manner. Trained on a large amount of text covering a wide variety of credit assignment paths, LLMs exhibit certain sequential few-shot learning capability in practice (Brown et al., 2020). Such a task or capability, rebranded as *in-context learning*, has been the subject of numerous recent studies (e.g., Xie et al. (2022); Chan et al. (2022b;a); Kirsch et al. (2022); Akyürek et al. (2022); von Oswald et al. (2022); Dai et al. (2022)).

Here our focus is the standard *sequential few-shot classification*. Let d, N, K, L be positive integers. In sequential N-way K-shot classification settings, a sequence processing NN with a parameter set $\theta \in \mathbb{R}^L$ observes a pair $(\boldsymbol{x}_t, y_t)$ where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the input and $y_t \in \{1, ..., N\}$ is its label at each step $t \in \{1, ..., N \cdot K\}$, corresponding to K examples for each one of N classes. After the presentation of these $N \cdot K$ examples (often called the *support set*), one extra input $\boldsymbol{x} \in \mathbb{R}^d$ (often called the *query*) is fed to the model without its label—in practice, an unknown label token $\varnothing$ is fed instead, e.g., $\varnothing = 0$—and the task of the model is to predict its label. During training, the parameters of the model θ are optimised to maximise the probability $p(y|(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_{N \cdot K}, y_{N \cdot K}), (\boldsymbol{x}, \varnothing); \theta)$ of the correct label $y \in \{1, ..., N\}$ of the input query $\boldsymbol{x}$. Since class-to-label associations are randomised and unique to each sequence $((\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_{N \cdot K}, y_{N \cdot K}), (\boldsymbol{x}, \varnothing))$, each such a sequence can be used as a new (few-shot) learning experience to train the model.

While K is a pre-defined constant in the setting above, ideally, we want to obtain a learning algorithm (encoded by the NN with parameters θ) that achieves the "maximum performance" with the minimum

number of few-shot learning examples i.e., here $N \cdot K$. We disregard mathematical rigour for now, in favour of an intuitive description that illustrates the motivation of our work (Sec. 2). We informally assume that there exists a convenient error function $\mathcal{L} : (\mathbb{R}^L, \mathbb{N}) \to \mathbb{R}$ that, given a number of learning steps $t \in \mathbb{N}$, evaluates the performance $\mathcal{L}(\theta, t)$ (e.g., the lower the better) of $\theta \in \mathbb{R}^L$ as a learning algorithm on a certain task which we assume to be fixed. We also assume a certain performance value $\mathcal{L}^* \in \mathbb{R}$ which we consider as "optimal" for this task. We further assume that there exists another convenient function $\tau : (\mathbb{R}^L, \mathbb{R}) \to \mathbb{N}$ that tells us the number of learning steps $\tau(\theta, \mathcal{L}^*)$ required by the learning algorithm parameterised by θ to achieve the performance $\mathcal{L}^*$. Then, the learning algorithm we want to find is the solution of:

$$\underset{\theta}{\text{minimise}}\, \tau(\theta, \mathcal{L}^*) \tag{1}$$

In general, we do not have access to such a function τ. But if we assume that, given θ, performance is monotonic w.r.t. the number of learning steps, i.e., for $t_1, t_2 \in \mathbb{N}$ such that $t_1 < t_2$, we have $\mathcal{L}(\theta, t_2) \leq \mathcal{L}(\theta, t_1)$, i.e., $\mathcal{L}(\theta, t_2)$ is better than $\mathcal{L}(\theta, t_1)$, and if we further assume that we have access to the "models" $\boldsymbol{W}^{\theta}_{t_1}$ and $\boldsymbol{W}^{\theta}_{t_2}$ trained on the fly by the learning algorithm θ at the corresponding steps t_1 and t_2, there are several ways of explicitly optimising θ such that the performance of $\boldsymbol{W}^{\theta}_{t_1}$ "matches" that of $\boldsymbol{W}^{\theta}_{t_2}$, e.g., via knowledge distillation. Such an auxiliary task encourages θ to achieve performance $\mathcal{L}(\theta, t_2)$ within the number of steps t_1 which is less than t_2 required by the current θ.

This idea directly relates to that of *bootstrapped meta-learning* by Flennerhag et al. (2022) but applied to few-shot learning with sequence processing NNs. Here we explore this approach in self-modifying NNs based on a modern variant of self-referential weight matrices (Schmidhuber, 1992; 1993) that has several appealing properties including its close connection to Transformers (Sec. 2.1).

2 Method

The main idea of this work is to apply *bootstrapped training* (Flennerhag et al., 2022) to improve few-shot learning with sequence processing NNs. In what follows, we specify the details of our sequence processor (Sec. 2.1), and describe the bootstrapped training objective (Sec. 2.2).

2.1 Background: Linear Transformers and Self-Referential Weight Matrices

Bootstrapped training (Sec. 2.2) can be applied to any sequence-processing NNs trained for few-shot learning. An application to NNs with a self-modifying weight matrix (WM) yields a particularly intuitive setting (illustrated in Figure 1; commented on later). Here we use the *modern self-referential weight matrix* (SRWM; Irie et al. (2022b)) as a generic self-modifying WM. An SRWM is a WM that sequentially modifies itself as a response to a stream of input observations (Schmidhuber, 1992; 1993). The modern SRWM belongs to the family of linear Transformers a.k.a. Fast Weight Programmers (FWPs; Schmidhuber (1991b); Katharopoulos et al. (2020); Choromanski et al. (2021); Peng et al. (2021); Schlag et al. (2021); Irie et al. (2021)). Linear Transformers and FWPs are an important class of the now popular Transformers (Vaswani et al., 2017): unlike the standard Transformers whose state size linearly grows with the context length, the state size of FWPs is constant w.r.t. sequence length (like in the standard RNNs). This is an attractive property, since several open problems with Transformers relate to its explicitly limited context window size (see e.g., Laskin et al. (2022); Reed et al. (2022)). Such a property is important also because potential extensions of the current in-context learning setting to further *in-context continual learning* etc. would require handling context lengths much larger than those supported by today's Transformers. Moreover, the duality between linear attention and FWPs (Schlag et al., 2021)—and likewise, between linear attention and linear layers trained by the gradient descent learning algorithm (Irie et al., 2022a; Aizerman et al., 1964)—have played a key role in certain theoretical analyses of few-shot learning capabilities of Transformers (von Oswald et al., 2022; Dai et al., 2022).

The dynamics of an SRWM (Irie et al., 2022b) are as follows. Let d_{in}, d_{out}, t be positive integers, and $\otimes$ denote outer product. At each time step t, an SRWM $\boldsymbol{W}_{t-1} \in \mathbb{R}^{(d_{\text{out}}+2*d_{\text{in}}+1)\times d_{\text{in}}}$ observes an input $\boldsymbol{x}_t \in \mathbb{R}^{d_{\text{in}}}$, and outputs $\boldsymbol{y}_t \in \mathbb{R}^{d_{\text{out}}}$, while also updating itself to $\boldsymbol{W}_t$ as follows:

$$\boldsymbol{y}_t, \boldsymbol{k}_t, \boldsymbol{q}_t, \beta_t = \boldsymbol{W}_{t-1}\phi(\boldsymbol{x}_t) \tag{2}$$

$$\boldsymbol{v}_t = \boldsymbol{W}_{t-1}\phi(\boldsymbol{q}_t);\ \bar{\boldsymbol{v}}_t = \boldsymbol{W}_{t-1}\phi(\boldsymbol{k}_t) \tag{3}$$

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} + \sigma(\beta_t)(\boldsymbol{v}_t - \bar{\boldsymbol{v}}_t) \otimes \phi(\boldsymbol{k}_t) \tag{4}$$

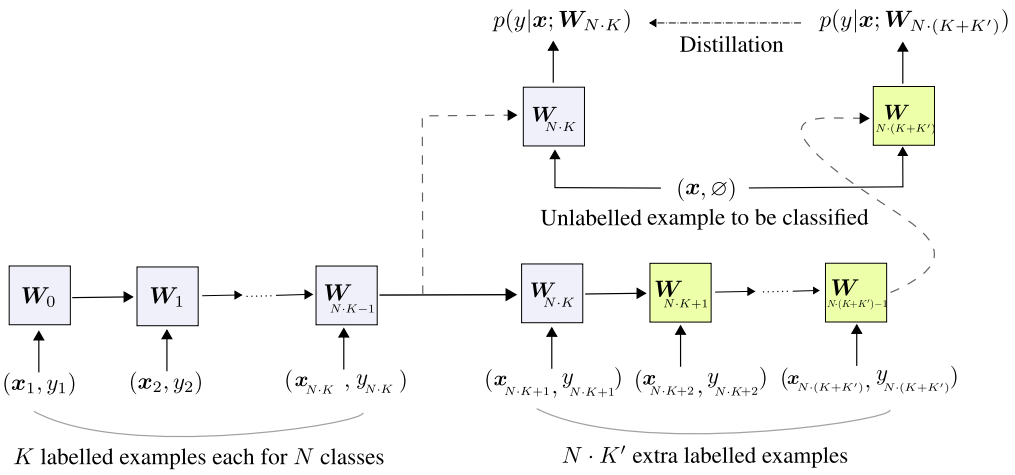

Figure 1: An illustration of bootstrapped training of a self-modifying WM $\boldsymbol{W}_0$. In N-way K-shot learning (in *grey*), $\boldsymbol{W}_0$ observes a sequence of $N \cdot K$ labelled examples, and yields $\boldsymbol{W}_{N\cdot K}$ which is used to classify an unlabelled input $\boldsymbol{x}$. In bootstrapped training (in *light green*), we feed $N \cdot K'$ extra inputs to $\boldsymbol{W}_{N\cdot K}$ to obtain $\boldsymbol{W}_{N\cdot(K+K')}$. We evaluate $\boldsymbol{W}_{N\cdot(K+K')}$ using the same test example $\boldsymbol{x}$, and its output is used as a teaching signal to train $\boldsymbol{W}_{N\cdot K}$ via knowledge distillation.

where $\boldsymbol{v}_t, \bar{\boldsymbol{v}}_t \in \mathbb{R}^{(d_{\text{out}}+2*d_{\text{in}}+1)}$ are value vectors, $\boldsymbol{q}_t \in \mathbb{R}^{d_{\text{in}}}$ and $\boldsymbol{k}_t \in \mathbb{R}^{d_{\text{in}}}$ are query and key vectors, and $\beta_t \in \mathbb{R}$ is the learning rate. σ and ϕ denote sigmoid and softmax functions respectively. $\boldsymbol{W}_0$ is the only trainable parameters of this layer, that encodes the initial self-modification algorithm.

2.2 Bootstrapped Training

Bootstrapped training of few-shot learning NNs is directly inspired by Flennerhag et al. (2022) (and also relates to Tack et al. (2022)). The overall process is depicted in Figure 1, and works as follows.

Let N, K, K', d be positive integers. In N-way K-shot classification (see Sec. 1), an SRWM (see Sec. 2.1) $\boldsymbol{W}_0$ observes a sequence of $N \cdot K$ labelled examples, and the resulting WM, $\boldsymbol{W}_{N\cdot K}$, is used to classify an unlabelled query input $\boldsymbol{x} \in \mathbb{R}^d$. $\boldsymbol{W}_0$ is trained to produce a sequence of self-modifications that result in $\boldsymbol{W}_{N\cdot K}$ capable of such classification. *Bootstrapped training* augments the setting above by continuing to feed $N \cdot K'$ extra labelled inputs to $\boldsymbol{W}_{N\cdot K}$ to obtain $\boldsymbol{W}_{N\cdot(K+K')}$. Then, we feed the same unlabelled example $\boldsymbol{x}$ to $\boldsymbol{W}_{N\cdot(K+K')}$, and use its output distribution as a teaching signal to train $\boldsymbol{W}_{N\cdot K}$ via knowledge distillation (Ba & Caruana (2014); Hinton et al. (2014); which Schmidhuber (1991a) calls *collapsing*). For this to work, we assume $\boldsymbol{W}_{N\cdot(K+K')}$ to perform better than $\boldsymbol{W}_{N\cdot K}$ thanks to extra learning examples (which is effectively the case in standard few-shot learning scenarios). By denoting the output distribution of the model with weights $\boldsymbol{W}_{N\cdot K}$ as $\boldsymbol{p}_{N\cdot K}(\cdot|\boldsymbol{x}) = \boldsymbol{p}(\cdot|\boldsymbol{x}; \boldsymbol{W}_{N\cdot K})$, the training objective function is:

$$\beta_1 \text{CE}(\boldsymbol{q}, \boldsymbol{p}_{N\cdot K}; \boldsymbol{x}) + \beta_2 \text{CE}(\text{sg}(\boldsymbol{p}_{N\cdot(K+K')}), \boldsymbol{p}_{N\cdot K}; \boldsymbol{x}) + \beta_3 \text{CE}(\boldsymbol{q}, \boldsymbol{p}_{N\cdot(K+K')}; \boldsymbol{x}) \tag{5}$$

where $\beta_1, \beta_2, \beta_3 \in \mathbb{R}_{>0}$ are scaling factors, CE denotes the standard cross-entropy loss:

$$\text{CE}(\boldsymbol{q}, \boldsymbol{p}; \boldsymbol{x}) = -\sum_{1 \leq y \leq N} \boldsymbol{q}(y|\boldsymbol{x}) \log(\boldsymbol{p}(y|\boldsymbol{x})) \tag{6}$$

and $\boldsymbol{q}(\cdot|\boldsymbol{x}) \in \{0, 1\}^N$ denotes the one-hot vector encoding the correct label of $\boldsymbol{x} \in \mathbb{R}^d$. In Eq. 5, the first and third terms correspond to the standard N-way K-shot and $(K + K')$-shot learning losses respectively, while the second term is the distillation loss (where sg denotes the "stop-gradient" operation; we want $\boldsymbol{W}_{N\cdot K}$ to match the performance of $\boldsymbol{W}_{N\cdot(K+K')}$, but not the opposite).

3 Experiments

Main Results. We conduct experiments on the standard Mini-ImageNet dataset (Vinyals et al., 2016; Ravi & Larochelle, 2017) for few-shot image classification. Our base architecture is similar to the

Table 1: Few-shot image classification accuracies (%) on Mini-ImageNet with bootstrapped training using $K' \cdot N$ extra examples for various values of K'. The row "K' = None" corresponds to the pure N-way K-shot learning without any auxiliary loss, i.e., $\beta_1 = 1$, $\beta_2 = 0$ and $\beta_3 = 0$ in Eq. 5.

K'	$K = 5$, $K_{\text{test}} = 5$	$K = 10$, $K_{\text{test}} = 10$
None	60.6 ± 0.4	64.1 ± 0.4
1	61.4 ± 0.5	66.9 ± 0.3
5	62.6 ± 0.3	$\mathbf{68.3} \pm 0.3$
10	$\mathbf{63.3} \pm 0.4$	67.6 ± 0.2

Table 2: Few-shot image classification accuracies (%) on Mini-ImageNet of models trained with or without bootstrapped training ("Bootstrap" Yes/No) for various K_{test}. $\beta_1 = 1$ and $\beta_3 = 1$ in all cases. If bootstrapping is used, $\beta_2 = 10.0$ for the $K = 10$ case, and $\beta_2 = 5.0$ for the $K = 5$ case.

	$K = 5$, $K' = 10$		$K = 10$, $K' = 5$	
Bootstrap	No	Yes	No	Yes
K_{test} = 1	44.8 ± 0.2	$\mathbf{45.9} \pm 0.5$	38.7 ± 0.2	44.0 ± 0.3
5	60.1 ± 0.4	63.3 ± 0.4	59.0 ± 0.4	$\mathbf{63.7} \pm 0.2$
10	64.0 ± 0.4	66.6 ± 0.4	64.6 ± 0.4	$\mathbf{68.3} \pm 0.3$
15	65.2 ± 0.4	67.3 ± 0.4	66.2 ± 0.2	$\mathbf{69.5} \pm 0.4$

one of Irie et al. (2022b) (but unlike these authors we use a single architecture for all K). We first evaluate various values of K' (number of extra "shots" used for bootstrapping) in the 5-way 5-shot and 10-shot settings. Table 1 shows the result. Here models are tested in the K_{test}-shot setting where $K_{\text{test}} = K$ (as in training). In all cases, we observe improvements by bootstrapped training over the baselines without any auxiliary losses (slight improvements in the 5-shot case). Now in Table 2, we fix K and K', and show how bootstrapped training affects the performance of the model evaluated with different numbers of examples at test time (i.e., for different K_{test}). Here, bootstrapping is disabled in the baselines ("Bootstrap No") but the auxiliary $(K + K')$-shot loss is used ($\beta_1 = \beta_3 = 1$) for fair comparison. We observe that bootstrapping not only improves K_{test}-shot learning where $K_{test} = K$ (i.e., where the auxiliary bootstrapping loss is applied in training), it also yields a chain of improvements at any K_{test}, resulting in an overall acceleration of few-shot learning.

Remarks & Current Limitations. In Table 2, we observe that the model trained for $K = 10$ underperforms the one trained with $K = 5$ on 1-shot learning task ($K_{\text{test}} = 1$)—in fact, both of them underperform the model specifically tuned for 1-shot learning ($\approx 47\%$ accuracy is reported in Irie et al. (2022b)). This trend may change by training the model in the "delayed label" setting (where correct labels are fed to the model one step later; see, Irie et al. (2022b)). Also, we have not managed to obtain any improvement in the 1-shot learning case (i.e., the same experiments as in Table 1 but with $K = 1$): we speculate that this case is difficult, since with $N = 5$ and $K = 1$, our SRWM can only generate rank-5 updates. Optimally tuning the model for 1-shot learning may leave little leeway for further improvements. Also importantly, so far we have not succeeded at confirming the improvements on another dataset, tiered-ImageNet (Ren et al., 2018): we ran the configuration used for Mini-ImageNet (without changing any hyper-parameters, including those for the model architecture), without observing out-of-the-box improvements. Further empirical investigations are still needed to determine under which conditions exactly bootstrapped training can help.

4 Conclusion

We study bootstrapped training of sequence-processing neural networks (NNs) for few-shot learning (FSL). Bootstrapped training encourages an FSL NN to accelerate FSL by letting it chase its own performance achievable with *more* training examples, using *less* examples. Our preliminary experiments on the standard Mini-ImageNet yield promising results. Further investigations using other datasets and tasks, e.g, imitation learning (Xu et al., 2022), are needed to confirm these trends.

REFERENCES

Mark A. Aizerman, Emmanuil M. Braverman, and Lev I. Rozonoer. Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25(6):917–936, 1964.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *Preprint arXiv:2211.15661*, 2022.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 27, pp. 2654–2662, Quebec, Canada, December 2014.

Tom B Brown et al. Language models are few-shot learners. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Virtual only, December 2020.

Stephanie CY Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K Lampinen, and Felix Hill. Transformers generalize differently from information stored in context vs in weights. In *NeurIPS Workshop on Memory in Artificial and Real Intelligence (MemARI)*, New Orleans, LA, USA, November 2022a.

Stephanie CY Chan, Adam Santoro, Andrew Kyle Lampinen, Jane X Wang, Aaditya K Singh, Pierre Harvey Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, November 2022b.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, 2021.

Neil E Cotter and Peter R Conwell. Fixed-weight networks can learn. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, pp. 553–559, San Diego, CA, USA, June 1990.

Neil E Cotter and Peter R Conwell. Learning algorithms and fixed dynamics. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, pp. 799–801, Seattle, WA, USA, July 1991.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *Preprint arXiv:2212.10559*, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 1126–1135, Sydney, Australia, August 2017.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. In *Int. Conf. on Learning Representations (ICLR)*, Virtual Only, April 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, Montreal, Canada, December 2014.

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, volume 2130, pp. 87–94, Vienna, Austria, August 2001.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Virtual only, December 2021.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *Proc. Int. Conf. on Machine Learning (ICML)*, Baltimore, MD, USA, July 2022a.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. A modern self-referential weight matrix that learns to modify itself. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 9660–9677, Baltimore, MA, USA, July 2022b.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only, July 2020.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. In *NeurIPS Workshop on Memory in Artificial and Real Intelligence (MemARI)*, New Orleans, LA, USA, November 2022.

Michael Laskin, Luyu Wang, et al. In-context reinforcement learning with algorithm distillation. *Preprint arXiv:2210.14215*, 2022.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *Int. Conf. on Learning Representations (ICLR)*, Vancouver, Cananda, 2018.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 2554–2563, Sydney, Australia, August 2017.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Available Online: https://blog.openai.com/better-language-models, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Int. Conf. on Learning Representations (ICLR)*, Toulon, France, April 2017.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research (TMLR)*, 2022.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Int. Conf. on Learning Representations (ICLR)*, Vancouver, Canada, April 2018.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. Meta-learning with memory-augmented neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 1842–1850, New York City, NY, USA, June 2016.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers are secretly fast weight programmers. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only, July 2021.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Jürgen Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Institut für Informatik, Technische Universität München. Technical Report FKI-126*, 90, 1990.

Jürgen Schmidhuber. Neural sequence chunkers. Technical Report FKI-148-91, Institut für Informatik, Technische Universität München, April 1991a.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to recurrent nets. Technical Report FKI-147-91, Institut für Informatik, Technische Universität München, March 1991b.

Jürgen Schmidhuber. Steps towards "self-referential" learning. Technical Report CU-CS-627-92, Dept. of Comp. Sci., University of Colorado at Boulder, November 1992.

Jürgen Schmidhuber. A self-referential weight matrix. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, pp. 446–451, Amsterdam, Netherlands, September 1993.

Jihoon Tack, Jongjin Park, Hankook Lee, Jaeho Lee, and Jinwoo Shin. Meta-learning with self-improving momentum target. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, December 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 3630–3638, Barcelona, Spain, December 2016.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *Preprint arXiv:2212.07677*, 2022.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, April 2022.

Mengdi Xu, Yuchen Lu, Yikang Shen, Shun Zhang, Ding Zhao, and Chuang Gan. Hyper-decision transformer for efficient online policy adaptation. In *NeurIPS Workshop on Foundation Models for Decision Making*, New Orleans, LA, USA, December 2022.

A Steven Younger, Peter R Conwell, and Neil E Cotter. Fixed-weight on-line learning. *IEEE Transactions on Neural Networks*, 10(2):272–283, 1999.