Towards General-Purpose In-Context Learning Agents

Louis Kirsch[1], **James Harrison**[2], **C. Daniel Freeman**[2], **Jascha Sohl-Dickstein**[2], **Jürgen Schmidhuber**[1 3]
[1]The Swiss AI Lab IDSIA, USI, SUPSI [2]Google DeepMind
[3]King Abdullah University of Science and Technology (KAUST)
mail@louiskirsch.com

Abstract

Reinforcement Learning (RL) algorithms are usually hand-crafted, driven by the research and engineering of humans. An alternative approach is to automate this research process via meta-learning. A particularly ambitious objective is to automatically discover new RL algorithms from scratch that use in-context learning to learn-how-to-learn entirely from data while also generalizing to a wide range of environments. Those RL algorithms are implemented entirely in neural networks, by conditioning on previous experience from the environment, without any explicit optimization-based routine at meta-test time. To achieve generalization, this requires a broad task distribution of diverse and challenging environments. Our Transformer-based Generally Learning Agents (GLAs) are an important first step in this direction. Our GLAs are meta-trained using supervised learning techniques on an offline dataset with experiences from RL environments that is augmented with random projections to generate task diversity. During meta-testing our agents perform in-context meta-RL on entirely different robotic control problems such as Reacher, Cartpole, or HalfCheetah that were not in the meta-training distribution.

1 Introduction

Improvements in Reinforcement Learning (RL) algorithms are mainly driven by the research and engineering of humans. Meta-learning instead automates this process (Schmidhuber, 1987; Parker-Holder et al., 2022) to discover novel RL algorithms with little human intervention. A key property of human-engineered learning algorithms is their applicability to a wide range of RL problems. To replace such algorithms with automatically discovered ones, those need to be equally general-purpose (Kirsch et al., 2020; Oh et al., 2020; Team et al., 2023).

A flexible approach to this problem is to embed the entire learning algorithm into a neural network such that the network learns-to-learn by in-context learning (Schmidhuber, 1993b; Hochreiter et al., 2001; Duan

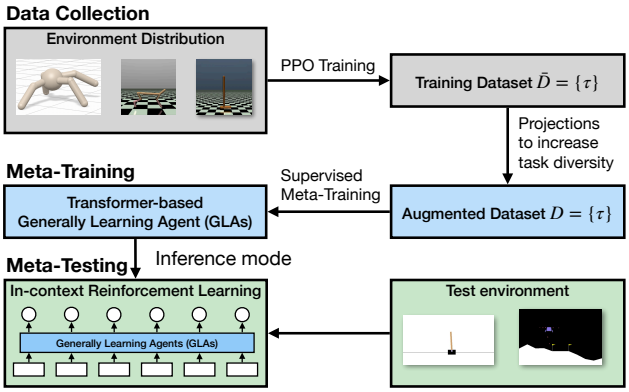

Figure 1: **Our Generally Learning Agents (GLAs) are meta-trained on augmented RL datasets via supervised learning.** First, one or more datasets of improving policies are collected using PPO. Next, these datasets are augmented with random observation and action projections to create a large diversity of tasks (environments). A Transformer is then trained to distill the (sped-up) learning process into a single in-context RL agent. Finally, at meta-test time, we can take an environment from a different domain (different actuators, observations, dynamics, and dimensionalities) and learn from rewards purely in-context without explicit hand-crafted RL algorithms or explicit gradient descent.

NeurIPS 2023 Workshop on Generalization in Planning (GenPlan 2023).

et al., 2016; Wang et al., 2016; Brown et al., 2020; Kirsch et al., 2022a,b). This requires that the entire learning algorithm be meta-learned from scratch, such as (re)discovering the principle of learning by gradient descent (Kirsch & Schmidhuber, 2021; Von Oswald et al., 2023; Akyürek et al., 2023) and credit assignment from rewards. In supervised learning, Large Language Models (LLMs) have led to strong in-context learning capabilities (Brown et al., 2020). Scaling laws were discovered that model the increase in predictive performance and capabilities with more training data and model parameters (Kaplan et al., 2020). Previous work suggested that these are also important drivers for the generality of in-context learning capabilities (Kirsch et al., 2022b). In RL, generalization of in-context learners has been more limited (e.g. Laskin et al., 2023; Melo, 2022).

Inspired by these works, we propose a path towards future agents that will be able to learn-how-to-learn in-context across a wide range of environments. To achieve such generalization, a broad task distribution of diverse and challenging environments will be needed during meta-training. Our Generally Learning Agents (GLAs, Figure 1) are an important first step in this direction. Our GLAs are meta-trained using supervised learning techniques (Laskin et al., 2023; Liu & Abbeel, 2023; Lee et al., 2023) on a dataset of experiences generated from PPO agents. We add augmentations to this dataset in the form of random projections to the observations and actions, which generate sufficient diversity to result in fairly general RL algorithms to be encoded into the neural network weights. We demonstrate that our GLAs are a significant step towards general-purpose cross-domain in-context learners by meta-testing them on different robotic control problems that were not seen during meta-training. We believe to be the first to show such cross-domain generalization for in-context RL.

2 Meta-Learning General-Purpose In-Context Learning Agents

In-context RL agents A reinforcement learning (RL) algorithm is a mapping $f : \tau \mapsto \theta$ from agent experience $\tau := (s_0, a_0, r_0, d_0, s_1, \ldots, s_L, a_L, r_L, d_L)$ to a policy $\hat{\pi}(a|s, \theta)$ with index $i \in \{0, \ldots, L\}$, observations s_i, actions a_i, rewards r_i, and terminations d_i.[1] We refer to those functions f as learning algorithms, where the expected return $\mathbb{E}_{\hat{\pi}}[R]$ is larger than the returns found in τ and tends to increase as new experiences are added to τ. Instead of modeling the learning algorithm f and policy $\hat{\pi}$ separately, we may also combine those to an in-context learning policy $\pi(a|s, \tau)$. Optimizing for π to discover learning algorithms then corresponds to meta-learning. We usually parameterize π using neural networks such as LSTMs (Hochreiter & Schmidhuber, 1997; Gers et al., 2000), Transformers (Vaswani et al., 2017), or linear Transformers (Schmidhuber, 1992; Katharopoulos et al., 2020; Schlag et al., 2021a) due to the sequential nature of τ. See Section 4 for related work.

Meta-learning via supervised learning Meta-optimization often involves standard RL techniques (Wang et al., 2016; Duan et al., 2016) by directly maximizing the average return over multiple episodes (together referred to as the lifetime of the agent). Recently, supervised learning has been very successful in language modeling (Brown et al., 2020) but also has shown great promise in reinforcement learning (Schmidhuber, 2019; Chen et al., 2021; Reed et al., 2022).

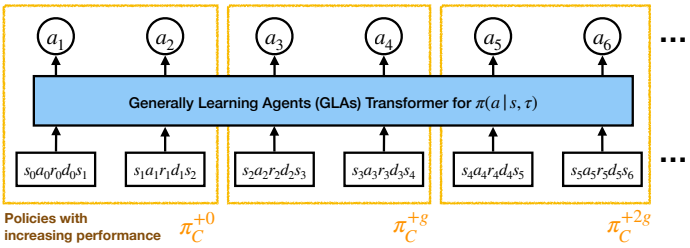

Figure 2: **Our RL agents reinforcement-learn purely in-context.** A Transformer is used to condition on previously observed environment transitions $(s_i, a_i, r_i, d_i, s_{i+1})$ and predicts the next action. Meta-Training is done on sequences of transitions generated from policies with increasing performance.

In this paper, we use supervised learning techniques for meta-learning π and demonstrate their potential to be used effectively for training across a broad environment distribution to discover novel generalizing RL algorithms. To do so, we collect sequences of transitions $\tau := (x_0, \ldots, x_L)$ with $x_i := (s_i, a_i, r_i, d_i, s_{i+1})$ that correspond to agent behavior with improving performance. Here, we generate those by running PPO (Schulman et al., 2017) on the meta-training environments. We then auto-regressively model the action distribution $p(a_i|s_i, \tau_{:i-1})$ given the previous transitions

[1] Here we assume that π acts in an MDP such that s is a sufficient state representation, but this can be extended to POMDPs by providing a representation of multiple previous observations instead.

$\tau_{:i-1}$. To go beyond algorithm cloning/distillation (Kirsch & Schmidhuber, 2021; Laskin et al., 2023) and exceed the performance of the human-engineered learning algorithm that generated the data, we sub-sample the data: Given a subset of the data $\tau_{j:k}$ generated from collection policy π_C^0, we predict the actions in $\tau_{l:m}$ that are generated from the policy π_C^{+g} that was updated by PPO for g iterations. We refer to g as the 'gap' between $\tau_{j:k}$ and $\tau_{l:m}$. We expect that as we increase the gap g, we meta-learn RL algorithms that learn more quickly. In summary, our supervised learning objective that we maximize using gradient ascent is

$$J(\theta) = -\sum_{i=l}^{m} D[p(a_i|s_i)|\pi_\theta(a_i|s_i, \tau_{j:k}, \tau_{l:i-1}, \eta_i)] \tag{1}$$

where D is a divergence, here the sum of the reverse and forward KL. The distribution $p(a_i|s_i)$ corresponds to the policy action distribution as recorded during PPO training. We use a decoder-only causally-masked Transformer with parameters θ to model π_θ, depicted in Figure 2. Additionally, we may condition on auxiliary information η_i that describes the amount of improvement that is expected. Options for η_i are the gap g, indicating how many PPO updates to distill into π, or the policy performance (return) at index i, or the location l within the lifetime $l \in 0, \ldots, L$ of the agent.

This supervised learning scheme allows us to perform efficient meta-training on offline data with a stationary objective and without the need for collecting additional data. Because the whole data sequence is known in advance, without intermediate environment interactions, this allows efficient training of Transformers to model π. Meta-training is summarized in Algorithm 1.

Training on broad task distributions If we have already solved the environments in the meta-training distribution with PPO, why is meta-learning a novel learning algorithm still useful? We hypothesize that meta-training across a sufficiently broad task distribution allows us to discover novel in-context RL algorithms that can be reused on many unseen RL problems that we typically care about. As a proof of concept, in this paper we augment the supervised training dataset with random linear projections in its observations and actions to generate sufficient task diversity. For the augmentations, we adopt the randomization methodology of previous work in supervised in-context learning (Kirsch et al., 2022b). We linearly project observations to 64 dimensions and actions to 16 dimensions. This also enables us to meta-test our GLAs across domains where actuators and observations are of varying dimensionalities. Kirsch et al. (2022b) have shown that such randomization techniques do not only result in in-context learners that learn to undo such projections but lead to fairly strong generalization even outside the trained-on tasks. We demonstrate that this approach also results in increasingly generalizable in-context learning algorithms for RL.

Meta-Testing During meta testing we simply auto-regressively evaluate the Transformer starting with an empty history $\tau \leftarrow ()$, growing τ with each experienced transition for K environment interactions. This implements in-context RL and is described in Algorithm 2. To make the quadratic complexity of the full self-attention tractable at inference time, we limit attention to the last 4 thousand transitions in our experiments.

Algorithm 1 Supervised Meta-Training for GLAs

procedure TRAIN(E) ▷ Meta-train on a set of MDPs E
  $\bar{D} \leftarrow \{\tau_e\}$ ▷ Collect a dataset of trajectories with increasing performance using PPO on environments $e \in E$
  $\theta \leftarrow$ random parameters ▷ Randomly initialize learning agent π
  while not converged **do**
    $D \leftarrow \text{augment}(\bar{D})$ ▷ Augment D; here using random projections on s_i and a_i
    $B \leftarrow \text{sample}(D)$ ▷ Sub-sample transitions from D
    $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta; B)$ ▷ Update learning agent π_θ via SGD on Equation 1
return Generally Learning Agent π_θ

Algorithm 2 Meta-Testing for GLAs

procedure TEST(e, π_θ) ▷ Meta-test on a new MDP e with a generally learning agent π_θ
  $\tau \leftarrow ()$ ▷ Initialize empty history
  for $k \leftarrow 1$ to K **do**
    Use policy $\pi_\theta(a|s, \tau)$ to obtain a new transition $\xi = (s, a, r, d, s')$ from environment e
    $\tau_k \leftarrow \xi$ ▷ Update history

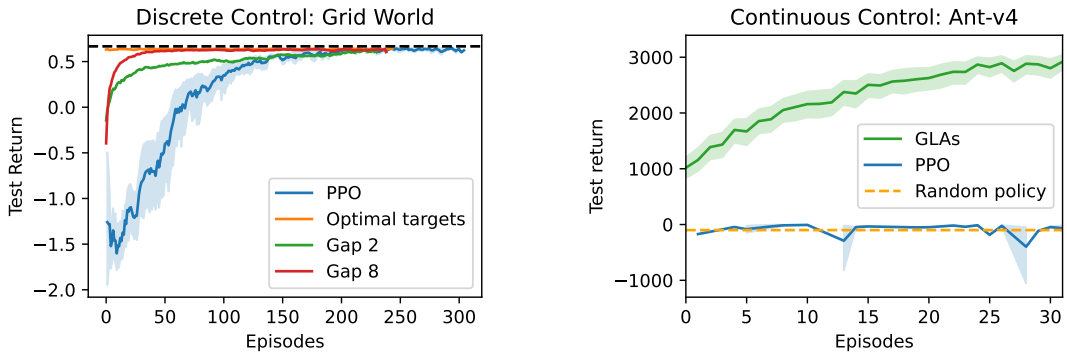

Figure 3: **Supervised learning discovers in-context learning agents on single tasks with controllable efficiency.** In both grid worlds and continuous control the mean return increases in-context with more environment interactions when running the GLAs Transformer. The rate of learning can be controlled by the gap g used during meta-training. On Ant-v4, the initial agent is already better than a random policy, suggesting that the learned in-context RL algorithm leverages task-specific knowledge. Shading indicates 95% confidence intervals with 64 meta-test training seeds.

3 Experiments

Supervised learning discovers learning agents on single tasks To begin, we demonstrate that our supervised meta-RL algorithm can discover in-context learning policies that encode a learning algorithm specific to a task. Figure 3 shows how the mean return increases at meta-test time in a simple grid world and in continuous control. The grid world consists of a 3x3 grid with directional movement actions and a goal position at a fixed location. For continuous control, we meta-train on the Ant-v4 MuJoCo environment. We observe that the initial test return is already larger than a random policy on Ant-v4 – suggesting that the learned learning algorithm leverages task-specific knowledge.

In-context Learning can be sped-up when the gap is increased How can the speed of learning be controlled? In Figure 3 (left), we demonstrate how an increased gap g can speed up learning at meta-test time. We also test the limiting case of a maximally large gap g that corresponds to the action targets taken by the optimal policy in the dataset. We find that in the case of a single task, the network simply learns the optimal policy.

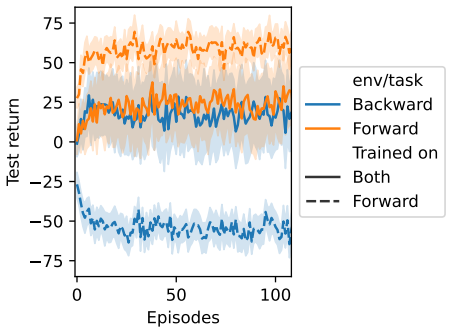

Figure 4: **In-context learning agents trained via supervised-learning can adapt to task changes.** On the standard Ant-Dir meta-learning benchmark the agent adapts when having seen both tasks during training, but does not learn in-context on an unseen task. Shading indicates 95% confidence intervals with 64 meta-test training seeds.

Standard meta-learning benchmarks (simple task distributions) How does the meta-test behavior change when we move from single tasks to task distributions? Here, we begin by using a standard meta-learning task distribution, the Ant-Dir environment (Finn et al., 2017) that involves the forward and backward task. The task is not part of the policy inputs and thus has to be inferred from observations and rewards. In the Ant-Dir environment (Figure 4), we observe that training on a single task only allows learning on that particular task, but does not generalize to the task of moving in the other direction. Conversely, meta-training on both environments allows for in-context learning in either task. We observe that for some meta-test training seeds, the task is incorrectly recognized, resulting in larger confidence intervals. The policy then follows the incorrect task. We hypothesize that this is a difficulty with the supervised meta-training objective on such meta-task distributions where the task is determined early in meta-test training and future actions simply condition on states seen during meta-training, independent of the task rewards. This may be alleviated by broader task distributions.

Generalization fails to significantly different tasks and environments We have motivated this work with the goal of in-context learning agents that generalize to a wide range of environments, across domains. Given this rather simple training distribution, we would not expect the agent to generalize to a different environment such as Cartpole or Reacher. To do so, we need to scale the diversity of the data distribution.

Scaling the task distribution enables cross-domain generalization Next, we augment the dataset by randomly projecting observations and actions, as described in Section 2. Here, we meta-train on the augmented Ant-v4 environment. This makes it impossible to directly encode the optimal policy and forces GLAs to learn in-context. Instead of just meta-testing the in-context learning agent on the same Ant-v4 environment or variations thereof, we also apply it to entirely different domains in Figure 5. We observe that the agent to some extent implements a learning algorithm that applies across domains. It performs in-context learning not just on the seen Ant-v4 environment (with an unseen random projection), but also generalizes its learning algorithm to the Reacher-v4, HalfCheetah-v4, and DeepMind-Control Cartpole environment. To the best of our knowledge, this is the first time that such strong generalization has been observed. The resulting performance is still suboptimal, but there is visible improvement on tasks with different actuators, task dynamics, and observations. Combining a broad task distribution of many existing continuous control tasks combined with augmentations as proposed here may significantly improve these results.

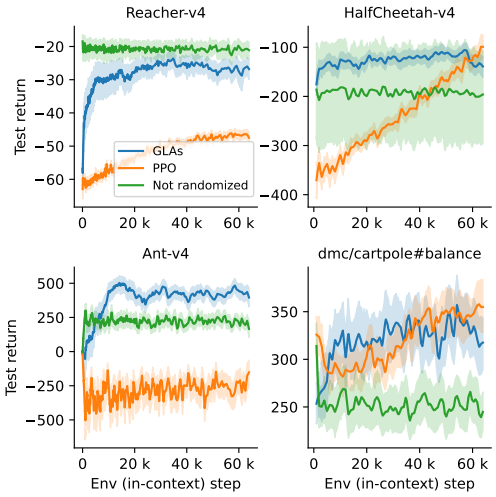

Figure 5: **GLAs generalize to novel domains via in-context RL.** After meta-training on augmented Ant-v4, the agent can learn in-context on Reacher-v4, HalfCheetah-v4, and DeepMind-Control Cartpole. Shading indicates 95% confidence intervals with 32 meta-test training seeds.

4 Related Work

In-context meta-RL and generalization Several works in supervised learning (Hochreiter et al., 2001; Santoro et al., 2016; Mishra et al., 2018) demonstrated that neural networks such as LSTMs can learn to learn in-context by updating their activations/memory based on an input feedback signal (Schmidhuber, 1993b). This mechanism has attracted significant recent attention in large language models and Transformer models (Brown et al., 2020; Chan et al., 2022; Garg et al., 2022; Kirsch et al., 2022b) and has been shown (Kirsch & Schmidhuber, 2021; Schlag et al., 2021a) to be closely related to fast-weight programmers (Schmidhuber, 1992, 1993a; Ha et al., 2017; Miconi et al., 2018; Najarro & Risi, 2020; Schlag et al., 2021b). Later, this concept was applied to reinforcement learning (Duan et al., 2016; Wang et al., 2016; Melo, 2022). Although these approaches discover policies that learn from interaction, their learning capabilities do not generalize to a wide range of environments. Thus, standard reinforcement learning algorithms, such as PPO, are still required when training on novel and unseen RL problems. Recent research attempted to broaden the generalization of in-context RL with parameter-shared neural networks (Kirsch et al., 2022a) or expressive game simulators (Team et al., 2023) but broad generalization has not yet been achieved. In this paper, we investigate how data diversity through augmentations can aid general-purpose in-context RL.

Supervised learning of policies There have been various attempts to use supervised learning techniques to train RL policies such as offline reinforcement learning (Levine et al., 2020), the distillation of many task-specific agents into a generalist agent (Reed et al., 2022), and the conditioning on returns (Schmidhuber, 2019; Chen et al., 2021). Unlike GLAs in this paper, these approaches implement zero-shot generalization instead of in-context learning.

Supervised meta-learning of RL algorithms Meta-learning of in-context RL agents has recently been approached with supervised learning techniques. Laskin et al. (2023) distilled learning tra-

jectories generated from PPO agents into neural networks, related to learning algorithm cloning in supervised learning (Kirsch & Schmidhuber, 2021). Later work demonstrated that sorting offline experiences by their return and conditioning Transformer models on those leads to improved policy behavior (Liu & Abbeel, 2023). Using optimal transitions as supervised targets (Lee et al., 2023) can also result in in-context learning behaviors. In this paper, we argue that such methods are primarily useful when the discovered reinforcement learning algorithms generalize to a wide range of problems. This is particularly relevant when the meta-learning algorithm requires existing handcrafted learning algorithms, such as PPO, to generate the data. Furthermore, we suggest that data can be subsampled by introducing 'gaps' in the increase of returns in concatenated trajectories to discover in-context learning agents that learn faster. This includes Lee et al. (2023) as a special case, where the gap is maximal to use the best-performing (optimal) policy.

5 Conclusion

Reinforcement Learning (RL) research has produced a wide range of methods to learn from rewards. In this paper, we instead searched for novel RL algorithms purely by conditioning on previous experience from the environment, without any explicit optimization at meta-test time. Compared to previous work in memory-based and in-context meta-RL, we have shown that, given a sufficiently rich environment distribution, the discovered RL algorithms start generalizing across domains, moving us closer to automating RL research through meta-learning. To achieve this, we collected an offline dataset of agent experience with improving performance and then augmented the dataset with random projections in the observation and action space. When meta-trained on these environments using supervised learning, the resulting RL algorithms encoded in our Generally Learning Agents (GLAs) generalize to robotic control problems that are significantly different from training, such as meta-training on Ant and learning in-context on the Reacher and Cartpole environments.

We believe that our approach provides a foundation for new large models, trained across an extremely diverse set of RL environments, to enable efficient learning from feedback far beyond our current RL algorithms. Based on our initial experiments in this paper, we plan to further improve generalization, robustness, and performance at meta-test time. Further broadening the task distribution with generated, real, and augmented environments will improve the (learning) capabilities of our GLAs. Finally, the cost of auto-regressive inference with Transformers and its relatively short context length are a limiting factor for discovering RL algorithms that make use of hundreds of thousands of environment transitions. Developing sequence models that can efficiently process and compress hundreds of thousands of environment transitions (e.g. Schlag et al., 2021a; Gu et al., 2022; Lu et al., 2023) will be important to improve the efficiency and expressivity of both learning and acting at meta-test time.

Acknowledgements

We thank the anonymous reviewers for their comments and feedback. This work was supported by the ERC Advanced Grant (no: 742870) and computational resources by the Swiss National Supercomputing Centre (CSCS, projects s1127 and s1205). We also thank NVIDIA Corporation for donating several DGX machines as part of the Pioneers of AI Research Award, IBM for lending a Minsky machine, and weights & biases (Biewald, 2020) for their great experiment tracking software and support.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

Lukas Biewald. Experiment Tracking with Weights and Biases, 2020. URL `https://www.wandb.com/`.

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.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, 2022.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

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

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pp. 87–94. Springer, 2001.

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.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Louis Kirsch and Jürgen Schmidhuber. Meta learning backpropagation and improving it. *Advances in Neural Information Processing Systems*, 34:14122–14134, 2021.

Louis Kirsch, Sjoerd van Steenkiste, and Juergen Schmidhuber. Improving generalization in meta reinforcement learning using learned objectives. In *International Conference on Learning Representations*, 2020.

Louis Kirsch, Sebastian Flennerhag, Hado van Hasselt, Abram Friesen, Junhyuk Oh, and Yutian Chen. Introducing symmetries to black box meta reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7202–7210, 2022a.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*, 2022b.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations*, 2023.

Jonathan N Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *arXiv preprint arXiv:2306.14892*, 2023.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Hao Liu and Pieter Abbeel. Emergent agentic transformer from chain of hindsight experience. *arXiv preprint arXiv:2305.16554*, 2023.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. *arXiv preprint arXiv:2303.03982*, 2023.

Luckeciano C Melo. Transformers are meta-reinforcement learners. In *International Conference on Machine Learning*, pp. 15340–15359. PMLR, 2022.

Thomas Miconi, Kenneth Stanley, and Jeff Clune. Differentiable plasticity: training plastic neural networks with backpropagation. In *International Conference on Machine Learning*, pp. 3559–3568. PMLR, 2018.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Elias Najarro and Sebastian Risi. Meta-learning through hebbian plasticity in random networks. *Advances in Neural Information Processing Systems*, 33:20719–20731, 2020.

Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado P van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. *Advances in Neural Information Processing Systems*, 33:1060–1070, 2020.

Jack Parker-Holder, Raghu Rajan, Xingyou Song, André Biedenkapp, Yingjie Miao, Theresa Eimer, Baohe Zhang, Vu Nguyen, Roberto Calandra, Aleksandra Faust, et al. Automated reinforcement learning (autorl): A survey and open problems. *Journal of Artificial Intelligence Research*, 74: 517–568, 2022.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pp. 9355–9366. PMLR, 2021a.

Imanol Schlag, Tsendsuren Munkhdalai, and Jürgen Schmidhuber. Learning associative inference using fast weight memory. In *International Conference on Learning Representations*, 2021b.

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. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Jürgen Schmidhuber. Reducing the ratio between learning complexity and number of time varying variables in fully recurrent nets. In *International Conference on Artificial Neural Networks*, pp. 460–463. Springer, 1993a.

Jürgen Schmidhuber. A 'self-referential'weight matrix. In *International conference on artificial neural networks*, pp. 446–450. Springer, 1993b.

Jürgen Schmidhuber. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, et al. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.