NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World

Ever Evolving Evaluator (EV3): Towards Flexible and Reliable Meta-Optimization for Knowledge Distillation

Li Ding* LIDING@UMASS.EDU
University of Massachusetts Amherst
Amherst, MA 01002, USA

Masrour Zoghi MZOGHI@GOOGLE.COM
Guy Tennenholtz GUYTENN@GOOGLE.COM
Maryam Karimzadehgan MARYAMK@GOOGLE.COM
Google Research
Mountain View, CA 94043, USA

Abstract

We introduce EV3, a novel meta-optimization framework designed to efficiently train scalable machine learning models through an intuitive explore-assess-adapt protocol. In each iteration of EV3, we explore various model parameter updates, assess them using pertinent evaluation methods, and then adapt the model based on the optimal updates and previous progress history. EV3 offers substantial flexibility without imposing stringent constraints like differentiability on the key objectives relevant to the tasks of interest, allowing for exploratory updates with intentionally-biased gradients and through a diversity of losses and optimizers. Additionally, the assessment phase provides reliable safety controls to ensure robust generalization, and can dynamically prioritize tasks in scenarios with multiple objectives. With inspiration drawn from evolutionary algorithms, meta-learning, and neural architecture search, we investigate an application of EV3 to knowledge distillation. Our experimental results illustrate EV3's capability to safely explore the modeling landscape, while hinting at its potential applicability across numerous domains due to its inherent flexibility and adaptability. Finally, we provide a JAX implementation of EV3, along with source code for experiments, available at: `https://github.com/google-research/google-research/tree/master/ev3`.

Keywords: Meta-learning, optimization, evolutionary algorithms, neural architecture search, knowledge distillation.

1. Introduction

This work proposes Ever Evolving Evaluator (EV3), a versatile meta-optimization framework for training scalable machine learning models. We introduce a new class of optimizers that incorporate an explore-assess-adapt protocol, carrying out the following steps in each iteration of the training loop:

1. **Explore** the model parameter space by proposing a variety of updates to model parameters, *e.g.*, by doing gradient-descent using different loss functions and optimizers.
2. **Assess** the proposals using one or more evaluation metrics that are relevant to the tasks of interest but are often non-differentiable and cannot be directly used in optimization. The assessments are performed on unbiased data to choose the best updates.
3. **Adapt** model parameters and/or topologies based on the results of statistical significance tests, which compare the performance of the best updates to the top-performing models previously identified by EV3. This adaptation also considers factors such as the history of progress.

*. Work done during an internship at Google.

EV3 has a number of desirable properties that merit elaboration:

- ☞ The explore step can be used to perform hyperparameter tuning for losses, optimizers, and schedules.
- ☞ Given sufficient training data or a sufficiently complex learning task (Feldman et al., 2019), the assess and the adapt steps of EV3 could minimize overfitting by using a validation set or unseen online data.
- ☞ The assess and the adapt steps do not impose a differentiability constraint on the evaluation methods: indeed, all that is required is the ability to compare models (*e.g.*, using human feedback on preferences).
- ☞ In the adapt step, updates generated by the previous steps can be rejected, thus eliminating the necessity for every proposed update to result in an improved model. This has a few important corollaries:
  - ➢ The gradients used to propose updates need not be unbiased, allowing for exploration in the sampling of gradient batches: *e.g.*, boosted sampling (Schapire and Freund, 2012) or keeping only particular types of samples (Hadsell et al., 2020; Soviany et al., 2022).
  - ➢ Distinct losses and optimizers can be used to generate updates when there is no clear winner.
  - ➢ More generally, the explore step can utilize multiple sources of knowledge or supervision (*e.g.*, different teacher models in knowledge distillation).
- ☞ In the multi-objective setting, the adapt step can prioritize certain evaluation measures and deprioritize others in addition to updating the model.

The particular design of EV3 draws inspiration from multiple domains including evolutionary optimization (Ding and Spector, 2022), meta-learning (Flennerhag et al., 2022), preference-based reinforcement learning (Wirth et al., 2017), neural architecture search (Elsken et al., 2019b), and gradient-free optimization methods such as Bayesian optimization (Shahriari et al., 2015).

In this paper, we assess the utility of EV3 to a particular application domain, knowledge distillation (KD), in order to highlight its ability to perform optimization easily and safely. We also provide an open-source implementation of EV3 in JAX (Bradbury et al., 2018) to facilitate its application in more domains.

2. Methods

In this section, we first introduce EV3 as a generic meta-optimization framework for ML models, then focus on an implementation of EV3 for training deep neural networks on the task of knowledge distillation.

2.1 The General EV3 Framework

An overview of EV3 is provided in Algorithm 1, which describes the single-objective version of the algorithm. EV3 introduces a protocol of three steps: explore, assess, and adapt, which are described as follows:

Explore In the explore step, various strategies are proposed to update the model parameters. This is a highly exploratory phase where ideas are generated, much like brainstorming. The use of different data subsets, optimizers, and learning rates fall under this category.

Assess The assess step is mainly about comparing the efficacy of the different updates proposed in the explore step. The use of i.i.d samples ensures a fair comparison.

Adapt The adapt step is the most decisive. It is where we decide whether changes need to be made to the model, including its parameters and architecture, or to modify our optimization strategy going forward.

2.2 EV3 for Knowledge Distillation

Knowledge distillation (KD) (Hinton et al., 2015) is a representative methods for model compression and acceleration, which effectively learns a small student model from a large teacher model. It has received

Algorithm 1 Ever Evolving Evaluator (EV3)

Input:

- Initial model graph $\mathbf{\Gamma}_1$ belonging to a graph family $\mathcal{F}$
- Initial model parameters $\mathbf{w}_1$
- Labeled data distribution $\mathfrak{D}$
- Evaluation method $\boldsymbol{\mu}$
- Smooth losses $\{\ell_i\}$ that correlate with $\boldsymbol{\mu}$

$\mathbf{w}^s = \mathbf{w}_1$
for $t = 1, \ldots$ **do**
  // I. Explore: propose candidate updates using losses.
  Sample (potentially biased) gradient batches $\{B_g\}$
  $\{\mathbf{g}_j\} = \text{EXPLORE}(\mathbf{\Gamma}_t, \mathbf{w}_t, \{\ell_i\}, \{B_g\})$
  // II. Assess: select the best updates using the evaluation method.
  $\widehat{\mathbf{w}} = \text{ASSESS}(\mathbf{\Gamma}_t, \mathbf{w}_t, \{\mathbf{g}_j\}, \boldsymbol{\mu}, \mathfrak{D})$
  // III. Adapt: decide how to update model parameters and graph.
  $\mathbf{\Gamma}_{t+1}, \mathbf{w}^s, \mathbf{w}_{t+1} = \text{ADAPT}(\mathbf{\Gamma}_t, \mathbf{w}^s, \mathbf{w}_t, \widehat{\mathbf{w}}, \boldsymbol{\mu}, \mathfrak{D})$

increasing attention from the community (Gou et al., 2021). In this work, we are particularly interested in applying EV3 to KD because KD methods often do not rely on labeled data for training. EV3 can thus use labeled data for validation to mitigate the risk of overfitting. We describe the essential components and design choices of using EV3 for KD in the following.

2.2.1 Disentangling Exploration and Evaluation

One major benefit of using EV3 for KD is that we can easily disentangle exploration and evaluation to reduce the risk of overfitting. In the explore step of EV3, we propose updates to the student model by distilling knowledge from the teacher model without using the labels. In the assess and adapt steps, the student model is evaluated on the labeled training data using metrics of interest. Based on this evaluation, we can decide which action to take, such as updating the model parameters or changing the model topology. In this work, we use accuracy as the evaluation metric for the image classification task. The adapt step employs a heuristic that updates the model parameters only if the validation accuracy significantly surpasses that of the current model, with a significance level set at 0.05. If there are more than a few updates that do not meet this criterion of significance, the heuristic will instead opt to expand the model.

2.2.2 Model Expansion

One useful feature of EV3 is the ability to increase model capacity when updating the parameters alone does not lead to improvements in performance. In order to retain the model's performance when expanding the model, we adopt the approach of network morphism (Chen et al., 2016; Wei et al., 2017; Elsken et al., 2019a). The basic idea is to initialize the parameters of the larger neural network based on the weights of the smaller network so that they have exactly the same output for any given input. Formally, a network morphism is a *function-preserving* operator $(T, \widetilde{w})$ that satisfies $N_w(x) = (TN)_{\widetilde{w}}(x)$ for every input $x \in X$, where N represents a model such as neural networks. In other words, N_w and $(TN)_{\widetilde{w}}$ represent the same function. This can be achieved by properly initializing new parameters $\widetilde{w}$ based on the current parameters w. This performance-preserving initialization averts the large cost of training all architectures from scratch.

In this work, we implement network morphism operators to expand a popular architecture in computer vision, deep residual networks (ResNet) (He et al., 2016). We design a network morphism expansion method that deepens the network by adding a ResNet block at each feature resolution.

2.2.3 Synergetic Training

Utilizing the components discussed above, we implement two variants of EV3 for KD.

EV3 We implement a basic EV3 optimizer without extensive exploration, *i.e.*, only one update is generated in the explore step. If the model's performance does not improve significantly for a few iterations, we expand the model using a network morphism in the adapt step. In this work, the significance test is performed as a two-population z-score test with a confidence level of $\alpha = 95\%$.

EV3-Synergy Recent work by Wang et al. (2023) indicates that KD methods do not always yield improved results when using larger models as the source of knowledge. Inspired by this finding, we introduce a synergetic training strategy for EV3, termed 'EV3-Synergy'. In this approach, the explore step involves proposing updates by distilling knowledge from a variety of models with different sizes. The process of EV3-Synergy begins by using the basic EV3 framework to generate a diverse set of models. We then further refine these models by running EV3 again, this time using both the initial teacher model and the newly created model collection as sources for distillation. The underlying rationale for this synergetic approach is the hypothesis that intermediate-sized models might lead to better performance outcomes compared to solely relying on the largest model for KD.

3. Experiments

To demonstrate the capability of EV3, we evaluate it on a knowledge distillation task for image classification on CIFAR-100 (Krizhevsky et al., 2009). The teacher model is ViT-B/16 (Dosovitskiy et al., 2020) and the students are ResNet models with different numbers of layers (ResNet-8/14/26/50). In terms of sizes, the teacher model has 86M parameters, and the student models vary from 0.08M to 0.77M parameters. The teacher model is pre-trained on ImageNet (Deng et al., 2009) and fine-tuned on CIFAR-100.

3.1 Implementation Details

We compare EV3 to two baselines: vanilla KD and network morphism. Experiments are single-trial with the same random seed. The vanilla KD baseline trains models of different sizes independently through KD for a fixed schedule of 110k iterations. For network morphism, we start from ResNet-8 and iteratively expand to larger models (ResNet-14/26/50), and train a fixed schedule of 110k iterations before each expansion. For EV3 and EV3-Synergy, instead of using a pre-defined schedule, the methods automatically determines when to expand the model. The main difference between EV3 and EV3-Synergy is that EV3 proposes only one update in the explore step by distilling ViT. On the other hand, EV3-Synergy utilizes a multi-pass training strategy. The first pass is identical to EV3, but in the second pass, multiple updates are proposed in the explore phase by distilling from the ViT model as well as all the ResNet models with the best performance.

3.2 Results

The experimental results, as illustrated in Fig.1, demonstrate the effectiveness of our proposed methods. Initially, in Fig.1 (left), we observe that EV3 outperforms network morphism in terms of accuracy, while maintaining similar computational costs. For instance, EV3 improves the accuracy of ResNet-14 (0.182M

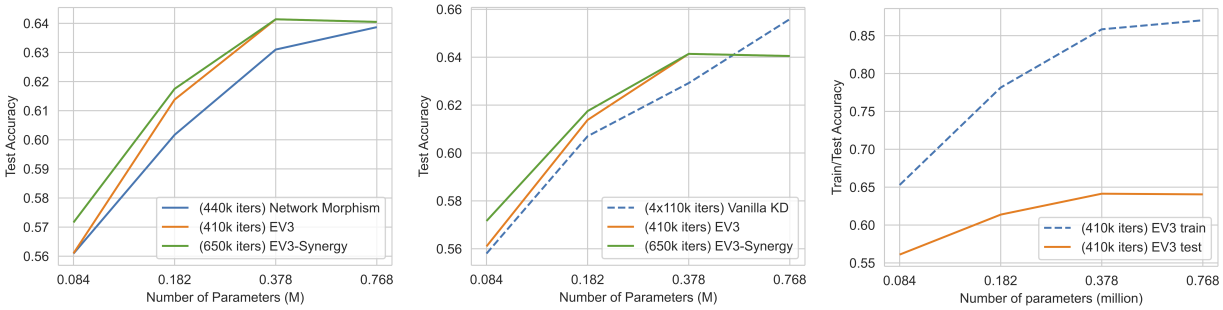

Figure 1: Results from left to right: 1) EV3 and EV3-Synergy compared to Network Morphism: Both EV3 and EV3-Synergy outperform Network Morphism. EV3-Synergy shows that the synergetic training strategy improves the performance of models of smaller sizes; 2) EV3 and EV3-Synergy compared to vanilla training: Both outperform vanilla KD except for the model of largest size. 3) Train vs. test error of EV3: the generalization gap becomes larger as the model size increases.

parameters) from 60.17 to 61.38, and ResNet-26 (0.378M parameters) from 63.10 to 64.14. This suggests that the adapt step of EV3 effectively finds a better schedule for model expansion using network morphism. Additionally, EV3-Synergy shows superior performance on smaller-scale models, such as enhancing the test accuracy of ResNet-8 from 56.10 to 57.16, validating our hypothesis that smaller models can benefit from distillation with intermediate-sized models.

The middle part of Fig. 1 indicates that EV3 generally outperforms the vanilla KD baseline across different model sizes, except for the largest one. Notably, neither EV3 nor network morphism surpasses the baseline on the largest model, suggesting two potential issues with model expansion via network morphism: 1) the possibility of being trapped in local optima due to multiple expansions, particularly when specific initializations (like identity weights in expanded layers) might deteriorate generalization, and 2) suboptimal scheduling for model expansion, potentially leading to overfitting, as indicated by EV3 selecting models with lower validation but higher test error.

Finally, Fig. 1 (right) displays the training error of EV3. It reveals an increasing generalization gap between training and test errors as the model size grows, suggesting a tendency of overfitting in larger models despite the absence of training labels in the EV3 updates. To address this in future work, we aim to conduct experiments with larger-scale datasets and explore the use of online data for KD.

4. Related Work

The proposed EV3 framework sits at the crossroads of several research areas. In this section, we discuss relevant research spanning evolutionary strategies, meta-gradients, neural architecture search, and preference-based reinforcement learning (RL).

Evolutionary Strategies Evolutionary algorithms (EAs) (Bäck and Schwefel, 1993) have been an instrumental force in the optimization of non-differentiable functions, leveraging mechanisms inspired by biological evolution such as reproduction, mutation, and selection. Recent research has focused on using EAs in machine learning, especially deep networks (Such et al., 2017; Jaderberg et al., 2017; Ding and Spector, 2021). There has also been a trend of using hybrid methods of gradient descend and components of EA (Cui et al., 2018; Ding and Spector, 2022; Ding et al., 2022). In a broader scope, Real et al. (2020) proposed an evolutionary AutoML framework that automatically discovers complete algorithms using basic mathematical operations as building blocks. In line with current trends, we introduce a meta-optimization framework,

which seeks to combine different learning approaches to provide a flexible tool for comprehensive training and development across various machine learning applications.

Meta-Learning Meta-learning aims to optimize the learning process itself and has seen substantial advancements, particularly in the domain of meta-gradients, which have emerged as pivotal for hyperparameter adaptation in non-stationary RL environments (Xu et al., 2018; Luketina et al., 2022). Research has delved into meta-optimizer characteristics, emphasizing the value of incorporating contextual information (Lange et al., 2023; Almeida et al., 2021). Besides gradient-based approaches, black-box methods that parameterize the entire update rule as neural networks show promise (Andrychowicz et al., 2016; Kirsch et al., 2022), while the intersection of bootstrapping techniques with meta-learning presents a nuanced dimension in RL scenarios (Flennerhag et al., 2021; Guo et al., 2020). The EV3 framework, while encompassing aspects of meta-learning, offers a distinctive perspective that emphasizes an iterative exploration, assessment, and adaptation paradigm, as opposed to directly utilizing meta-gradients. It fundamentally diverges from traditional meta-gradient approaches due to its broad meta-optimization scope, weaving architectural evolution and task-specific metric optimizations, instead of solely focusing on adapting learning strategies.

Neural Architecture Search Neural Architecture Search (NAS) (Elsken et al., 2019b) focuses on automating the process of selecting the most suitable network architecture for a given task. Early methods in this area were computationally expensive, involving exhaustive search through a vast architecture space (Stanley and Miikkulainen, 2002). However, more recent approaches have employed efficient search strategies. Real et al. (2017), Xie and Yuille (2017) and Real et al. (2019) showcased the use of evolutionary principles in NAS, while Liu et al. (2018) proposed a gradient-based method to search the architecture space efficiently. In the context of our work, we mainly adopt the network morphism strategy (Wei et al., 2017; Chen et al., 2016; Elsken et al., 2019a), which aims to speed up training by initializing the weights of novel architectures based on weights of architectures that have been trained before. One critical challenge with network morphisms and other architecture expansion methods is the tuning of schedule for training and expansion, which we aim to solve by using the EV3 framework.

Preference-based Reinforcement Learning Traditional RL optimizes an objective defined by a scalar reward signal. However, Preference-based RL (Wirth et al., 2017) proposes a paradigm shift, where feedback is based on preferences between trajectories rather than scalar rewards. This form of feedback often aligns better with human judgment, especially when scalar rewards are hard to specify. Recent works like (Christiano et al., 2017; Stiennon et al., 2020) have shown how such approaches can be deployed successfully in real-world scenarios, ensuring that AI systems capture human intent better. In the context of EV3, preference feedback can be used to guide the adapt step of the algorithm, illustrating our framework's flexibility and relevance to real-world optimization challenges.

5. Conclusion

In this paper, we propose EV3, a flexible meta-optimization framework designed for training machine learning (ML) models. Our investigation specifically focused on applying EV3 to knowledge distillation (KD) tasks. Experimental results demonstrate that EV3 outperforms network morphism in KD for image classification. Additionally, the EV3-Synergy variant, which incorporates a synergetic training strategy, further enhances these results. Future work will aim to narrow down the generalization gap observed in EV3 by employing strategies such as dataset splitting and online training. We also plan to extend EV3 to various other architectures, datasets, and tasks, such as using transformers for language-related tasks. Another potentially fruitful direction is the application of EV3 to multi-objective problems, including ML fairness.

Acknowledgments

We thank the Mudcats team, Esteban Real, and other colleagues at Google for helpful discussion, Lee Spector and Scott Niekum for their advice, and the anonymous reviewers for their thoughtful feedback. We acknowledge the funding and computational resources provided by Google Research.

References

Diogo Almeida, Clemens Winter, Jie Tang, and Wojciech Zaremba. A generalizable approach to learning optimizers. *arXiv preprint arXiv:2106.00958*, 2021.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. URL `http://arxiv.org/abs/1511.05641`.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Xiaodong Cui, Wei Zhang, Zoltán Tüske, and Michael Picheny. Evolutionary stochastic gradient descent for optimization of deep neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

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.

Li Ding and Lee Spector. Evolving neural selection with adaptive regularization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1717–1725, 2021.

Li Ding and Lee Spector. Optimizing neural networks with gradient lexicase selection. In *International Conference on Learning Representations*, 2022.

Li Ding, Ryan Boldi, Thomas Helmuth, and Lee Spector. Lexicase selection at scale. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 2054–2062, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Efficient multi-objective neural architecture search via lamarckian evolution. In *International Conference on Learning Representations*, 2019a. URL `https://openreview.net/forum?id=ByME42AqK7`.

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

Vitaly Feldman, Roy Frostig, and Moritz Hardt. Open problem: How fast can a multiclass test set be overfit? In *Conference on Learning Theory*, pages 3185–3189. PMLR, 2019.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. *arXiv preprint arXiv:2109.04504*, 2021.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. In *International Conference on Learning Representations*, 2022.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, pages 3875–3886. PMLR, 2020.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

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, pages 7202–7210, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Robert Lange, Tom Schaul, Yutian Chen, Chris Lu, Tom Zahavy, Valentin Dalibard, and Sebastian Flennerhag. Discovering attention-based genetic algorithms via meta-black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 929–937, 2023.

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

Jelena Luketina, Sebastian Flennerhag, Yannick Schroecker, David Abel, Tom Zahavy, and Satinder Singh. Meta-gradients in non-stationary environments. In *Conference on Lifelong Learning Agents*, pages 886–901. PMLR, 2022.

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

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

Esteban Real, Chen Liang, David So, and Quoc Le. Automl-zero: Evolving machine learning algorithms from scratch. In *International conference on machine learning*, pages 8007–8019. PMLR, 2020.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 05 2012. ISBN 9780262301183. doi: 10.7551/mitpress/8291.001.0001. URL `https://doi.org/10.7551/mitpress/8291.001.0001`.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

Yuzhu Wang, Lechao Cheng, Manni Duan, Yongheng Wang, Zunlei Feng, and Shu Kong. Improving knowledge distillation via regularizing feature norm and direction. *arXiv preprint arXiv:2305.17007*, 2023.

Tao Wei, Changhu Wang, and Chang Wen Chen. Modularized morphing of neural networks. *arXiv preprint arXiv:1701.03281*, 2017.

Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Lingxi Xie and Alan Yuille. Genetic cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1379–1388, 2017.

Zhongwen Xu, Hado P van Hasselt, and David Silver. Meta-gradient reinforcement learning. *Advances in neural information processing systems*, 31, 2018.