Evolutionary Distributed Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce Evolutionary Distributed Training (EDT), a nature-inspired approach to distributed model training. EDT replaces centralized gradient synchronization with evaluation, pairwise model crossover, and mutation, enabling communication-efficient training across loosely connected devices. While early investigations show limited effectiveness in language model pretraining, EDT demonstrates strong potential in reinforcement learning (RL). In complex multi-agent environments, EDT facilitates diverse reward exploration and emergent strategies by evolving both policy and reward functions, outperforming traditional training in adaptability and strategic diversity. We also hypothesize EDT as a promising framework for post-training and alignment, offering optimization towards multi-objective, non-differentiable goals. This work positions EDT as a scalable, evolutionary recipe for distributed learning, offering early insights into where it may best fit within the deep learning landscape.

1 Introduction

The last decade of machine learning has been defined by scale. Transformer-based language models have demonstrated remarkable emergent capabilities when scaled to billions or even trillions of parameters, prompting discussions of "Sparks of Artificial General Intelligence" [2]. Recent success with Reinforcement Learning via Verifiable Reward (RLVR) brings us a step closer to human-level general intelligence [18][22]. Towards that goal, we ponder a more fundamental question — How did intelligence come to be in the first place? A simple answer would be — natural evolution. From the collective mind of an ant colony to the all-powerful human brain, intelligence emerges from an evolutionary algorithm. In this work, we intend to investigate how evolutionary algorithms may fit into the current landscape of deep learning. More specifically, we make two observations about nature: 1. Natural evolution is highly parallelized and decentralized. 2. Most complex organisms in nature perform pairwise reproduction. From this, we propose Evolutionary Distributed Training (EDT), a nature-inspired recipe for applying evolutionary algorithms as an outer layer to distributed training. Models are evaluated locally, selected based on fitness, and recombined via pairwise crossover and mutation to produce offspring. This process repeats iteratively, without requiring centralized synchronization of gradients. In the following sections we will conduct some early exploration of EDT on Large Language Model (LLM) Pre-training, Post-training, and Reinforcement Learning (RL).

2 Language Model Pretraining

2.1 Motivation

Recent work on distributed language model training has introduced new ways to scale model training beyond traditional data-parallelism. In particular, DiLoCo [5], a variant of federated averaging [11] presents a promising distributed optimization algorithm that matches the performance of data-parallel while communicating 500 times less [5]. DiLoCo works by periodically synchronizing model deltas with a central aggregator using SGD with momentum and Nesterov acceleration. Inspired by the biological process of pairwise reproduction, we propose a decentralized alternative to DiLoCo using our Evolutionary Distributed Training (EDT) recipe. Rather than averaging deltas across all workers, we perform pairwise averaging, producing offspring that inherit characteristics from two parent models. Our hope is that the momentum accumulated over an ancestry might act as a larger batch, while the evolutionary algorithm optimizes for better-performing ancestries that carry the right momentum.

2.2 Method and Experimental Setup

We initialize a 6-million parameter language model based on the LLaMA architecture [21] and GPT-2 tokenizer [15]. The training dataset is derived from TinyStories [6].

To adapt the DiLoCo algorithm, we replace global averaging with pairwise model crossover. The training loop proceeds as follows:

1. **Fitness Evaluation:** Each model is evaluated on a fixed and shared validation set consisting of **400 examples**.
2. **Selection:** We perform **rank-based selection**, where models with higher fitness scores are more likely to be selected as parents.
3. **Crossover:** For each pair of parent models (A, B):
   - Compute the base model as a linear interpolation of their pre-update checkpoints.
   - Average their training deltas to form a gradient.
   - Average the velocity buffers.
   - Apply the gradient to the base model using SGD with momentum and Nesterov acceleration.
   - Carry over the resulting model to the next generation.
4. **Local Training:** Each worker performs **400 inner steps** using the **AdamW optimizer**.

We compare the following configurations:

1. **DiLoCo Baseline:** Running on **8 workers** with a per-device **batch size of 1**.
2. **EDT with Rank-Based Pairwise Selection:** Running on **8 workers** with a per-device **batch size of 1**.
3. **EDT with Random Pairwise Selection:** Running on **8 workers** with a per-device **batch size of 1**.
4. **EDT with Increased Update Frequency:** Using $4\times$ more frequent updates (100 inner steps) on **8 workers** with a per-device **batch size of 1**.

2.3 Results

Figure 1 presents the validation perplexity across 32 outer steps. We observe that DiLoCo outperforms EDT significantly over the course of training. While DiLoCo baseline shows signs of continuing to decline in perplexity, our pairwise method appears to be slowing down to a stop. Furthermore, rank-based and random pairwise selection yield nearly identical learning curves, indicating that the evolutionary selection mechanism does not significantly influence model learning in this setting.

Increasing the update frequency in EDT (4x the updates) leads to faster initial perplexity reduction. However, all configurations converge to similar perplexity values after equivalent total training steps. This suggests that more frequent mixing may improve short-term convergence but does not overcome EDT's inherent limitations in aggregating knowledge or carrying gradient information across generations.

Given initial exploration results, we conclude that EDT does not seem to fit directly in pre-training settings.

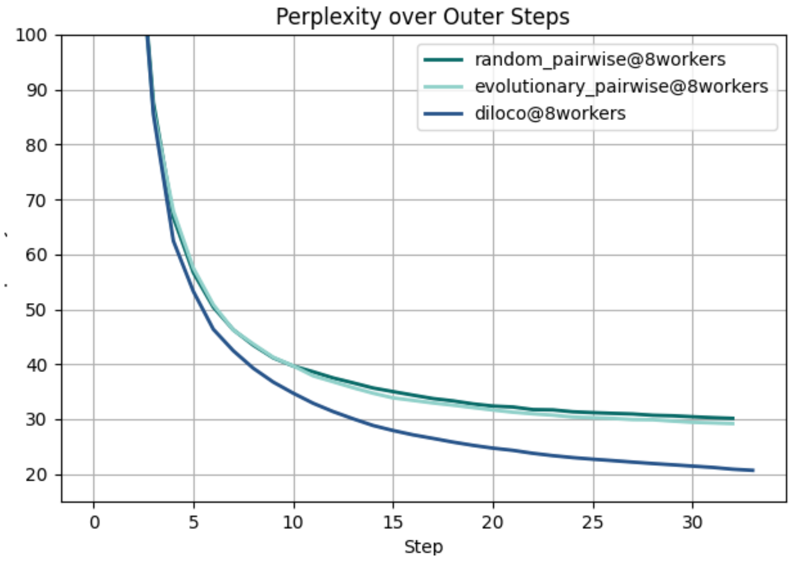

Figure 1: Comparing perplexity over training steps for random selection, evolutionary selection and DiLoCo baseline

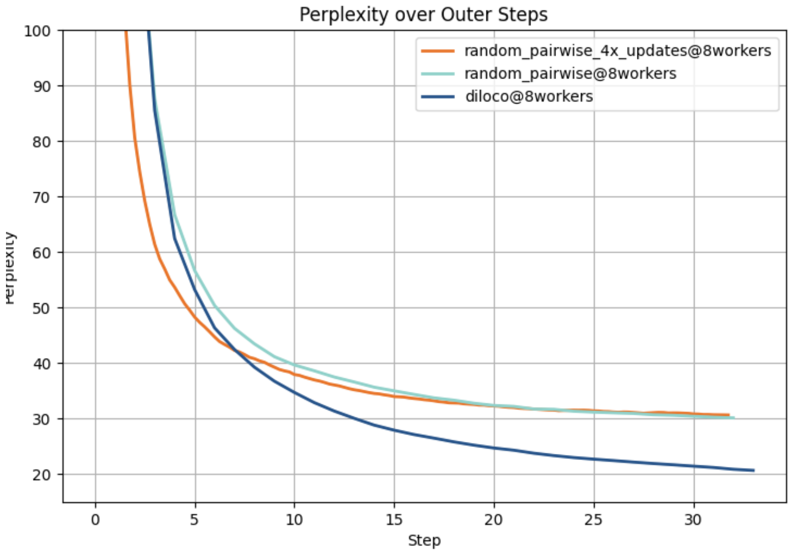

Figure 2: 4x updates shows no improvement in overall perplexity score

3 Reinforcement Learning

3.1 Motivation

Reinforcement Learning (RL) has achieved remarkable success across domains, but remains fundamentally constrained by its reliance on well-shaped reward functions. Sparse or deceptive reward landscapes often result in agents that fail to learn meaningful behaviors. If you use a simple reward function for a complex or long-horizon problem, the model likely will have a hard time learning anything meaningful due to having too much space to explore. While if you manually create a complex reward function, you risk the model overfitting and gaming the reward. In the end we are stuck in an exploration-exploitation dilemma. One solution to this is by scaling up batch size with massively parallel environments [1]. Such a method requires a substantial number of rollout workers [13], which poses significant infrastructural challenges.

We propose that **Evolutionary Distributed Training (EDT)** offers a natural mechanism to encourage exploration without requiring handcrafted rewards or dense feedback.

To this end, we extend EDT to reinforcement learning by treating **combinations of reward functions as mutable DNA sequences**. Each worker trains a local policy model using its own reward configuration, allowing a population of agents to explore diverse behaviors. Models are selected based on performance (e.g., Elo rating in self-play), and offspring inherit both policy parameters and modified reward functions through crossover and mutation. This introduces structured randomness into the optimization landscape, promoting a balance between exploitation of high-performing strategies and exploration of novel ones.

Unlike the LLM pretraining setting, where averaging gradients is key, RL is inherently noisy and local. The decentralized and diverse nature of EDT aligns naturally with RL's challenges.

Furthermore, consider the following **analogy**: Say you want to find a treasure. You decided to make a thousand clones of yourself to do the searching. However, since these are your clones, they behave similarly, they all have the same habits, same ways of thinking. Maybe you will be able to find the treasure one day, but in the process you are doing a lot of repeated exploration, looking under the same rock multiple times. Your search space is confined by your function space [12]. What if instead of a thousand you, we have a thousand different people, each with their own quirks and behaviors, we expand the search space – EDT could promote a more efficient random search for solutions.

3.2 Method and Experimental Setup

We experiment with EDT-RL in a Smash Bros style platform fighting game gym environment [10]. Agents control characters that can move, jump, and perform light or heavy attacks in 8 directions, etc, with the objective of knocking their opponent off the platform.

Model: Each agent is a 64k-parameter transformer-based policy model, trained via Proximal Policy Optimization (PPO) [17].

Fitness: We use an Elo rating system based on double-elimination tournaments to evaluate model performance via self-play matches within the population.

Reward DNA: There are 18 different reward functions, including noop. Each agent is assigned a "DNA sequence" of size 6 encoding the combination of reward functions.

Training Procedure:

1. Each worker trains a local policy guided by its reward DNA for one generation.
2. After evaluation, agents are selected based on Elo score (fitness).
3. Pairs of parents are chosen to produce offspring:
   - **Policy crossover**: Spherical linear Interpolate weights of parent models.
   - **DNA crossover**: Uniformly mix reward shaping terms from both parents.
4. A mutation step applies to a subset of the offspring, randomly altering components of the DNA (e.g., replacing and introducing new reward functions).

We compare EDT-RL to a standard Population-Based Training (PBT) setup with fixed, manually defined reward shapings and identical PPO hyperparameters.

3.3 Results

In the 8-worker setting, EDT-RL produces competitive performance relative to PBT.

Scaling to **32 workers over 40 generations** reveals striking emergent behavior. Certain reward functions, while harmful in isolation, led to novel strategies when combined and refined across generations. One such mutation was the `Floor is Lava` penalty, which forces the model to keep jumping. Another penalized the use of light attacks. While suboptimal at first, these mutations encouraged exploration of aerial tactics. By generation 20, a new strategy emerged—**"ground slamming"** — where an agent jumped above their opponent and executed heavy vertical attacks repeatedly. This tactic gained dominance after agents discovered that stacking enough damage enabled a physics engine exploit: slamming opponents through the stage floor, bypassing typical

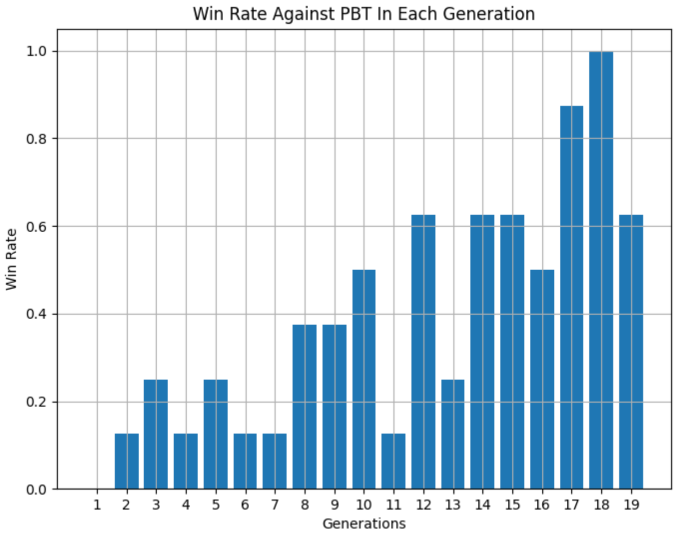

Figure 3: EDT@8workers vs PBT@8workers win rate at each generation. (Tie is considered a loss.)

knockback mechanics. By generation 40, this strategy dominates the population. However, the "floor is lava" gene that led to this strategy became scarce over the generations as agents that are not confined by this restriction were able to combine grounded strategies with "ground slam", achieving greater performance.

3.4 Discussion

These results highlight the unique strengths of EDT-RL:

- **Exploration via Mutation:** Introducing stochastic reward variation enabled the population to discover creative, high-impact strategies that were not explicitly encoded in the reward design.
- **Pairwise Crossover:** Pairwise crossover encourages diversity while maintaining the ability to propagate knowledge across the population, allowing efficient use of compute.
- **Scalability:** Larger populations enabled deeper exploration and more effective recombination, supporting rapid behavioral innovation. O(1) communication overhead allows decentralized and potentially asynchronous training over poorly connected devices.

4 Language Model Post-Training - Concept

4.1 Motivation

While pretraining focuses on fitting to large-scale data distributions, post-training aims to steer pretrained language models toward helpful, safe, and capable behavior—often using only a small amount of data. At this stage, we are not optimizing for maximum likelihood, but for alignment with specific goals and constraints. This makes post-training a natural candidate for **Evolutionary Distributed Training (EDT)**.

Post-training often involves multi-objective tradeoffs (e.g., helpfulness vs. harmlessness), non-differentiable goals (e.g., model judged quality), and the risk of overfitting to narrow supervision signals. Evolutionary algorithms are well-suited for such settings due to their ability to optimize:

- **Multi-objective functions** without requiring scalar reward collapse.
- **Non-differentiable objectives**, such as evaluations from LLM-as-a-judge or task-specific win-rates.

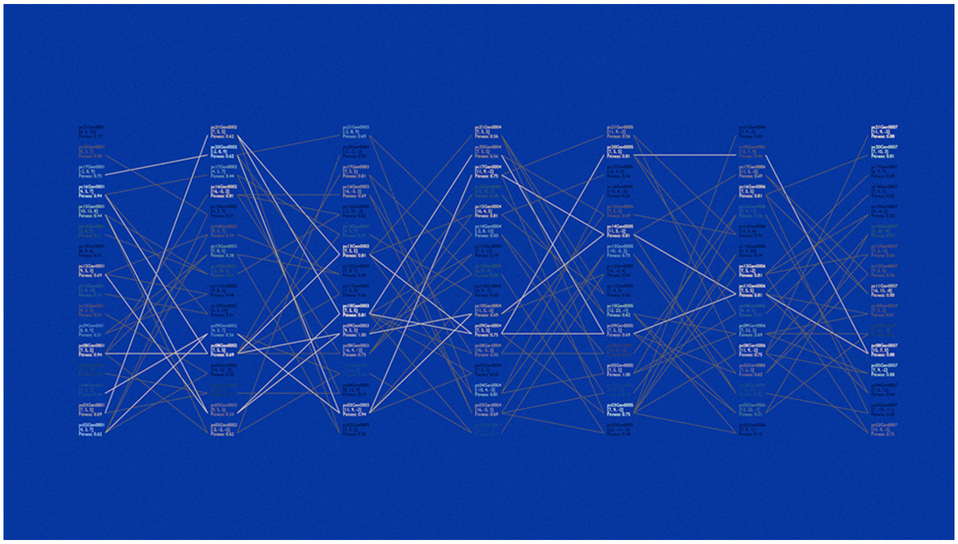

Figure 4: EDT post-training framework concept

- **Policy-space exploration**, encouraging behavioral diversity and robustness.

We propose EDT as a flexible wrapper around post-training. Models can be evaluated using any mix of automated metrics, benchmarks, or preference judgments. Pairwise crossover and mutation allow exploration across behavioral variants, while decentralized evaluation prevents synchronization bottlenecks. The goal is not to fine-tune for narrow targets, but to *stabilize training, prevent overfitting, and encourage emergent capabilities* through a process of natural evolution.

Due to computational constraints, we are able to conduct experiments at the moment. This direction could be explored in future work.

5 Related Work

5.1 Evolutionary Algorithms and Genetic Optimization

Evolutionary algorithms (EAs) have long been used to solve optimization problems through principles inspired by natural selection. Early work in this domain includes Genetic Algorithms (GAs) [7] which use crossover and mutation to evolve solutions over generations. In the context of neural networks, Neuroevolution techniques like NEAT (NeuroEvolution of Augmenting Topologies) [20] explore ways to evolve both architectures and weights. More recent large-scale efforts such as OpenAI's Evolution Strategies [16] demonstrated scalability of black-box optimization for deep reinforcement learning, rivaling traditional policy gradients in some tasks.

While these methods are powerful, they are rarely used in the era of large-scale language models due to their sample inefficiency and communication overhead. Our proposed Evolutionary Distributed Training (EDT) differs in that it applies EA concepts specifically to decentralized training settings — targeting communication-constrained environments and non-differentiable objectives.

5.2 Federated Learning and Distributed LLM Training

Federated learning [11] enables decentralized optimization by allowing clients to train local models and periodically synchronize updates with a central server. While originally proposed for privacy-preserving applications, its communication efficiency has inspired variants in large-scale language model training.

Recent work such as DiLoCo (Distributed Low-Communication) [5], shows extremely competitive performance in Large Language Model training compared to data-parallel, while supported by scaling laws [3] and open source reproduction [8].

5.3 Reinforcement Learning and Exploration

Exploration remains a central challenge in reinforcement learning, particularly in environments with sparse or deceptive rewards. Classical techniques include ϵ-greedy exploration, entropy regularization, etc. More recently, large-scale self-play has proven effective in discovering complex strategies, as in AlphaZero [19] and OpenAI Five [13].

Evolutionary approaches have also been used in RL to promote exploration. Novelty Search [9] and Quality-Diversity algorithms [4] evolve agents to encourage behavioral diversity, often outperforming reward-based methods in hard-exploration domains. Population-Based Training (PBT) [**?**] introduced hybrid training-evolution schemes, combining gradient descent with hyperparameter evolution. Our EDT-RL builds on these ideas, extending the concept to evolve both policy weights and reward functions in a decentralized fashion.

5.4 Post-Training and Model Steering

Post-training — the phase following large-scale pretraining — is where language models are adapted to specific behaviors and goals. Typical methods include supervised fine-tuning on curated instructions [14], Reinforcement Learning from Human Feedback (RLHF) [**?** 14], and more recently, Reinforcement Learning via Verifiable Rewards (RLVR) [18].

6 Limitations

Most of the initial experiments were done on a single laptop RTX4070 by simulating having multiple workers in a loop. Later, we found a way to scale up to 32 RTX4080s, and was able to complete most experiments in the RL section. However, when we started experimenting with post-training, we were noticed by the university and informed that personal research is not allowed on university infrastructure. Due to limitation in compute, our experimental results are limited.

7 Conclusion

We introduced Evolutionary Distributed Training (EDT), a nature-inspired recipe for decentralized and parallelized model training grounded in the principles of natural evolution — local evaluation, pairwise reproduction, and mutation. While our initial experiments show that EDT does not offer significant advantages in language model pertaining, it shows strong promise in reinforcement learning. In particular, EDT-RL enabled the emergence of novel strategies through reward function mutation and policy recombination, demonstrating its potential as a tool for structured exploration in sparse or deceptive environments.

This work represents an early investigation into the possible roles evolutionary algorithms can play in distributed training systems. We intend to use this paper as a stepping stone towards collecting feedback and finding potential collaborators.

References

[1] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula, 2020.

[2] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[3] Zachary Charles, Gabriel Teston, Lucio Dery, Keith Rush, Nova Fallen, Zachary Garrett, Arthur Szlam, and Arthur Douillard. Communication-efficient language model training scales reliably and robustly: Scaling laws for diloco, 2025.

[4] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2018.

[5] Arthur Douillard, Qixuan Feng, Andrei A. Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc'Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models, 2024.

[6] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

[7] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1st edition, 1975. 2nd Edition, MIT Press, 1992.

[8] Sami Jaghouar, Jack Min Ong, and Johannes Hagemann. Opendiloco: An open-source framework for globally distributed low-communication training, 2024.

[9] Joel Lehman and Kenneth Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19:189–223, 06 2011.

[10] Andrew Magnuson. Utmist ai2 2025. `https://github.com/ajwm8103/UTMIST-AI2-2025`, 2025.

[11] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

[12] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:241–276, September 1999.

[13] OpenAI, :, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning, 2019.

[14] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[15] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[16] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[19] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

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

[21] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian

Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[22] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that EDT is a nature-inspired decentralized training framework drawing from evolutionary principles such as mutation, local evaluation, and recombination. They correctly frame the approach as exploratory, with the strongest results demonstrated in reinforcement learning rather than pretraining. Limitations—such as the underperformance in language model pretraining—are acknowledged up front, and the scope is appropriately set as an early investigation into the use of evolutionary strategies for distributed training.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated section discussing limitations, particularly highlighting the gap between EDT and traditional SGD-based methods in large-scale pretraining. It acknowledges the sensitivity of EDT to hyperparameter choices, the difficulty in tuning mutation operations, and the scalability concerns for very large model sizes. The authors also mention the open question of how well EDT generalizes beyond the tasks explored, especially in highly synchronized or gradient-sensitive settings.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not present new theoretical results or formal proofs. Its contributions are empirical and algorithmic in nature, focusing on the design and evaluation of EDT across various tasks.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper includes detailed descriptions of the experimental setup, including pseudocode for EDT, environment configurations for RL tasks, architectural details of the models used, and descriptions of the reward mutation mechanisms. Evaluation metrics, and infrastructure used are clearly reported. The algorithm itself should be reproducable as all implementation details are given.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The authors have not released code so far due to time constraints. However, code may be released in the short future as supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Full details may be submitted through supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We were compute constrained and unable to run experiments multiple times.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper discloses compute details for the experiments in the limitations section

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research adheres to the NeurIPS Code of Ethics. It uses publicly available datasets and environments with no personally identifiable data or sensitive information. No human subjects or high-risk applications were involved. Environmental impact and compute use are transparently disclosed, and reproducibility is prioritized through open-sourcing.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses how EDT can democratize large-scale training by reducing reliance on centralized orchestration and allowing training across heterogeneous, fault-prone resources, potentially making cutting-edge AI more accessible.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any pretrained models or datasets with high risk for misuse. The proposed methodology is architectural and experimental in nature, evaluated on standard, publicly available benchmarks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and existing baselines are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No assets are used in the paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve human subjects or crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The work does not involve research with human subjects, and therefore no IRB approval is necessary.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs are not used in a non-standard way or as a core component of the methods described in the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.