Optimizing Energy Dissipation in Magnetization Reversal of the Ising Model: Comparing Constrained Deep Reinforcement Learning and Gradient Free Methods

Joshua Hang Sai Ip, Mira Khare, Kushal Nimkar

December 2023

Abstract

The efficient control of nanoscale magnetic systems is crucial for the development of future data storage technologies. In this pursuit, the magnetization reversal in the Ising model provides an essential theoretical framework to explore strategies that reduce energy consumption during such processes. This study applies and contrasts reinforcement learning (RL) and Neuroevolution of Augmented Topologies (NEAT) to optimize energy dissipation during magnetization reversal, offering insights into the application of machine learning for managing the dynamic behavior of the Ising model. We begin by framing the Ising model problem—flipping the magnetization from all spins down to all spins up within a two-dimensional square lattice—as an RL problem, where agents interact with an environment to attain goals with minimal cost. However, in this setting, the agent must also take actions of flipping spins under temperature and magnetic field constraints, represent a unique challenge in system control.

Our methodology involved testing the implementation of constraints to a traditional Deep RL algorithm, specifically Proximal Policy Optimization (PPO), as well as a gradient-free approach, NEAT, to discover protocols that minimize entropy production—a proxy for energy dissipation. To guide the RL agent toward efficient solutions, we implemented a shear transformation constraint that ensures a smooth progression from initial to final states. We find that the RL indeed solves the problem under the constraints, but the entropy produced was significantly above the theoretical minimum found previously using numerical simulations. In comparison with the Constrained Deep RL approach, NEAT proved to be adept at single-objective optimization, showcasing an ability to generate diverse strategies with varying efficiencies. However, it encountered inefficiencies in multi-objective optimization, where constructing a dense Pareto front is critical. This limitation emphasizes the need for refined or alternative methods for complex problem spaces within nanomagnetic systems.

1 Introduction

The Ising model serves as a pivotal many-body system for understanding magnetization reversal, which is fundamental to the theoretical underpinnings of nanomagnetic storage devices. Investigations into this model shed light on critical processes such as information erasure and replication while also highlighting strategies to diminish energy consumption during magnetization reversal by optimizing dissipation-reducing protocols [1, 2]. Here we investigate two approaches to solve such a system: Deep RL and a gradient free approach using a genetic algorithm.

In RL agents interact with an environment to achieve a predefined goal, often optimizing a particular reward or cost function in the process. An example is the task of maneuvering a vehicle from point A to point B, with the objective of minimizing fuel consumption or energy usage while adhering to certain constraints like speed limits or travel time. The Ising model problem formulation can be likened to an RL scenario where the environment is a two-dimensional lattice populated with binary spin variables instead of roads or paths, and the agent's goal is to attain magnetization reversal—akin to the vehicle reaching its destination. Here, energy plays a twofold role, analogous to the fuel in our vehicular analogy: it is a resource the agent seeks to minimize and also a key factor defining the system's state, which in turn influences the agent's available actions. In the Ising system, the agent's mission is achieved while operating under specific constraints that ensure fair comparison among various "protocols" or strategies: all approaches must start at the initial state (A) with all spins down and a specific magnetic field and temperature, and transition to the final state (B) with all spins up and an opposing magnetic field at the same temperature. This constraint ensures the magnetization indeed reverses, much like ensuring that all vehicles travel a route from A to B for a fair comparison of fuel efficiency.

The objective is to obtain a magnetization reversal with minimal energy dissipation (also referred to as minimal entropy production) in the Ising system. The entropy production term in Eq. 3 tracks the energy cost associated with each action—flipping a particular spin. Modulating the magnetic field and temperature of the system allows us to determine the sequence and timing of spin flips, striking a balance between achieving the target magnetization state and conserving the energy of the system. We emphasize that the agent can only control certain parameters, temperature and external magnetic field, that influence the probability of spins flipping, but has no knowledge of the underlying system dynamics.

2 Background and System Setup

The two-dimensional Ising model is situated on a square lattice consisting of $N = 32^2$ sites, subject to periodic boundary constraints in both dimensions. Individually, each lattice site i harbors a binary spin $S_i = \pm 1$. The system's energy function at a given timestep is:

$$E \equiv -J \sum_{\langle i,j \rangle} S_i S_j - h \sum_{i=1}^{N} S_i, \tag{1}$$

where J denotes the Ising coupling constant ($J = 1$ here), h is the external magnetic field. The first summation over $\langle i, j \rangle$ is over pairs of adjacent lattice site spins (models interactions between particles) while the second summation extends over all individual lattice sites. The system-wide magnetization at a given timestep is defined by

$$m \equiv \frac{1}{N} \sum_{i=1}^{N} S_i \tag{2}$$

where initially, the spins are homogeneously oriented downward ($S_i = -1; \forall i$), defining an initial system-wide magnetization of $m_i = -1$.

The dynamics of the system after every timestep are dictated by Glauber Monte Carlo Dynamics. For a given particle, the probability of flipping the spin is

$$P(S_i = k | S_i = -k) = \frac{1}{1 + e^{\beta \Delta E}} \quad \forall k \in \{1, -1\}$$

where $\beta = 1/T$ is the inverse temperature, and ΔE is the energy change of the system if the spin were to be flipped. Thus, at every timestep we iterate through every particle and decide to flip it or not, and the final lattice structure is the next new state for the system.

The entropy production σ we seek to minimize over the course of a trajectory is calculated as:

$$\sigma = \beta_f E_f - \beta_i E_i - \sum_k \beta_k \Delta E_k, \tag{3}$$

where E_f and E_i represent the final and initial energies of the system, respectively; ΔE_k represents the change in energy between consecutive timesteps k and $k-1$, and β_k represent the reciprocal temperature at the k^{th} step. Notably, for any trajectory that unites endpoints of equal temperature but opposing magnetic fields, assuming magnetization has been reversed, the first two terms of this equation negate each other.

3 Case Study Outline

The objective of this study is to navigate the temperature T and magnetic field h from an initial state $\lambda_i = (T_i, h_i) = (0.65, -1)$ to a final state $\lambda_f = (T_f, h_f) = (0.65, 1)$ within a finite number of timesteps t_f, under the constraint of magnetization inversion ($m_i = -1$ to $m_f = 1$), with the goal of minimizing entropy generation (σ). At every timestep t, the agent receives the state (m, t_i) and then adjusts the system (T, h) in an attempt to find an optimal trajectory. Past work involving numerical simulations has found optimal protocols [2, 3], but RL has not been applied in such a scenario. To facilitate this particular problem structure, we incorporate constraint enforcement to both a traditional Deep RL algorithm (PPO) and an evolutionary-based method, Neuroevolution of Augmented Topologies. We find the former finds a (suboptimal) solution while the latter recovers optimal solutions.

3.1 Formulation

Because the input temperature T and external magnetic field h are known endpoints of the trajectory, it is logical to enforce some kind of constraint directly to try to help the policy converge to reasonable trajectories. To do this, the agent will output values that are transformed into real (T, h) at every control input. Specifically,

$$\begin{aligned}(T, h) = \text{Agent}(t/t_f, m) &+ (1 - t/t_f)[\text{Agent}(t/t_f, m) - \text{AgentNetwork}(0, -1)] \\ &+ (t/t_f)[\text{Agent}(t/t_f, m) - \text{AgentNetwork}(1, 1)]\end{aligned}$$

where

$$\text{Agent}(t/t_f, m) = \begin{cases} (T_i, h_i) & t/t_f = 0 \\ (T_f, h_f) & t/t_f = 1 \\ \text{AgentNetwork}(t/t_f, m) & \text{otherwise} \end{cases}$$

where AgentNetwork is the neural network policy being used. The above acts as a forcing function to ensure the agent output space is encouraged to progress towards the final state as time goes on in a smooth manner, and that the initial and final states are correct. We designed a sparse reward for this problem

$$r(t/t_f, m) = \begin{cases} -|m - 1| - K_\sigma * \sigma & t/t_f = 1 \\ 0 & \text{otherwise} \end{cases}$$

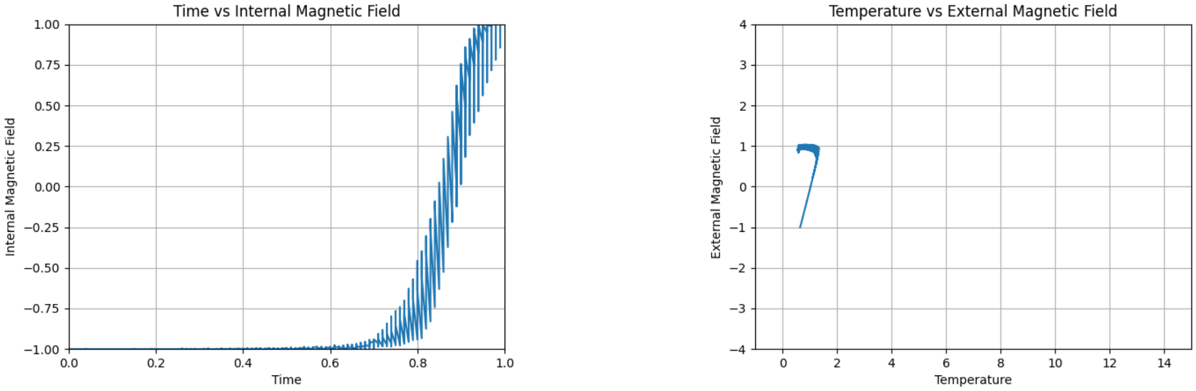
where σ is the entropy generated so far, and $K_\sigma = 10^{-2}$ is a hyperparameter weighting the entropy term and magnetization flip term. We note that once the magnetization flip is achieved, only the entropy term will remain, which should encourage trajectories that reduce the entropy.

4 Constrained Deep RL

4.1 Results

We find that applying this constraint shear results in an agent that does successfully result in the agent learning a protocol that results in an internal magnetization flip ($m = 1$ to $m = -1$; Figure 1a). The set of actions chosen by the agent is shown in Figure 1b. However, looking at the entropies generated by evaluation trajectories during training 2 we see that we approach a value of ~ 2400, which is far from an optimal trajectory (which is ~ 100). We do note that this trajectory is better than the naive trajectory of jumping from the initial conditions to the final conditions, which results in an entropy of around 4096, suggesting some learning indeed happened. Looking at the shape of the entropy of trajectories as training continues, we see it sharply levels off after 400000 training steps (Figure 2).

To gauge the possibility of converging to local minima, we tuned hyperparameters in Table 1. We also compared our shear constraint scheme to a naive constraint scheme where we simply forced the initial and final (T, h) to be correct, and let the agent directly spit out (T, h) as actions (Table 2). Without the shear constraint, we see that the RL agent fails to find meaningful trajectories. We also observed that we obtained similar results when we tried another Deep RL technique such as DQN utilizing our constraint enforcement technique (not shown). We also attempted to implement a reward structure where intermediate timesteps had a bonus $r \sim -\beta_k \Delta E_k$ and the final reward was scaled differently, to try to motivate small changes in (T, h) but found that did not change the results significantly.



(a)

(b)

Figure 1: (a) Plotting the internal magnetic field (m) as a function of time. PPO with the shear constraint does achieve the desired bit flip. (b) The temperature vs external magnetic field (action space) taken by the agent.

Hyperparameters	ent_coef=0	ent_coef=0.001	ent_coef=0.01
learning_rate =3e-4	2412 ± 55	4815 ± 168	10117 ± 2714
learning_rate=3e-5	2972 ± 73	4502 ± 85	23158 ± 2536

Table 1: Adjust the learning rate and entropy coefficient hyperparameters for PPO. Values are average generated entropy over 50 evaluation trajectories, ± 1 standard deviation.

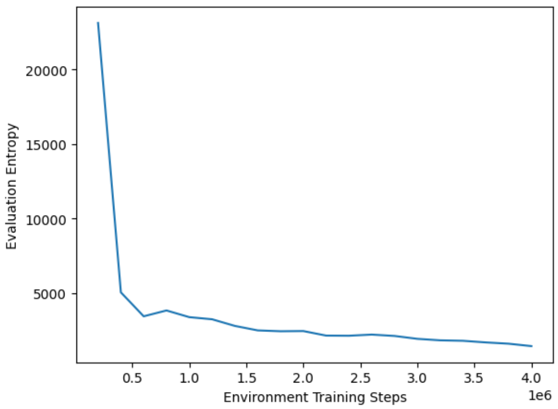


Figure 2: Tracking how entropy of evaluation trajectories. Every 200000 training steps 50 trajectories were generated, with each data point being the average entropy generated.

Constraint Scheme	Shear Constraint	Fixed Endpoints Only
Trajectory Entropy	2412 ± 55	35036 ± 4581

Table 2: Comparing naive constraint scheme to the sheared one. Values are average entropy over 50 evaluation trajectories, ± 1 standard deviation

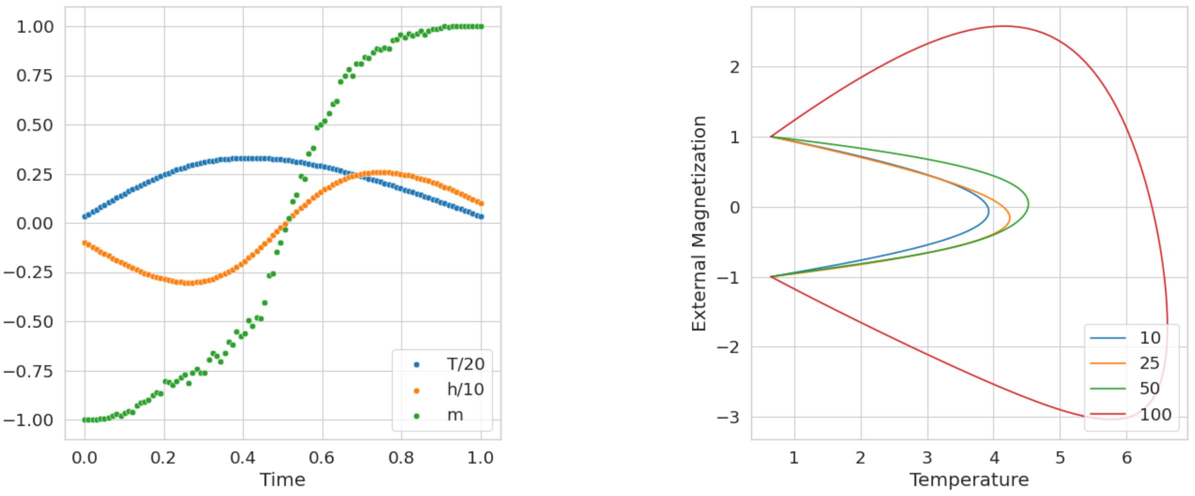
5 Neuroevolution of Augmented Topologies (NEAT)

Given the limitations observed with Deep RL, particularly in its convergence to suboptimal solutions and local minima as seen in the PPO results, we sought an alternative approach capable of exploring the possible solutions in a more diverse and exploratory manner. To this end, we employed the NeuroEvolution of Augmenting Topologies (NEAT) approach, a genetic algorithm that can evolve both the weights and architectures of artificial neural networks (ANNs) [4]. NEAT fundamentally differs from Deep RL by starting with a minimal network and evolving complexity via evolutionary methods, such as selective breeding for stronger networks, mutations for new abilities, and growth with additional nodes and connections. This approach allows NEAT to innovate, discovering unique, efficient strategies that Deep RL's gradient optimization might miss. Its speciation mechanism also prevents hasty convergence, fostering a wide range of strategies and effectively exploring policy space. NEAT's adaptive structure and exploration make it ideal for tackling complex problems like the Ising model of magnetization reversal. Note that we modify the NEAT implementation as in [5], returning the averaged reward of a policy in order to account for the stochastic nature of the problem. For a brief explanation of how NEAT operates and its underlying principles, please see the appendix section of this paper.

5.1 Results

Using NEAT, a population of 50 genomes was optimized across different generational spans—shown in Fig. 3b—provides a closer look at the evolutionary progress enabled by NEAT. By analyzing the developments in the temperature (T) and external magnetic field (h) relationships for protocols trained for varying numbers of generations, we gain a window into the algorithm's refinement process. It becomes evident that as the number of generations increases, the protocols evolve to feature more nuanced control of the system's parameters. Not only do these generational improvements permit a closer approach to the target magnetization state, but they also reveal an evolution toward operational subtlety, pinpointing intermediate states that circumvent the critical temperature even in early generations.

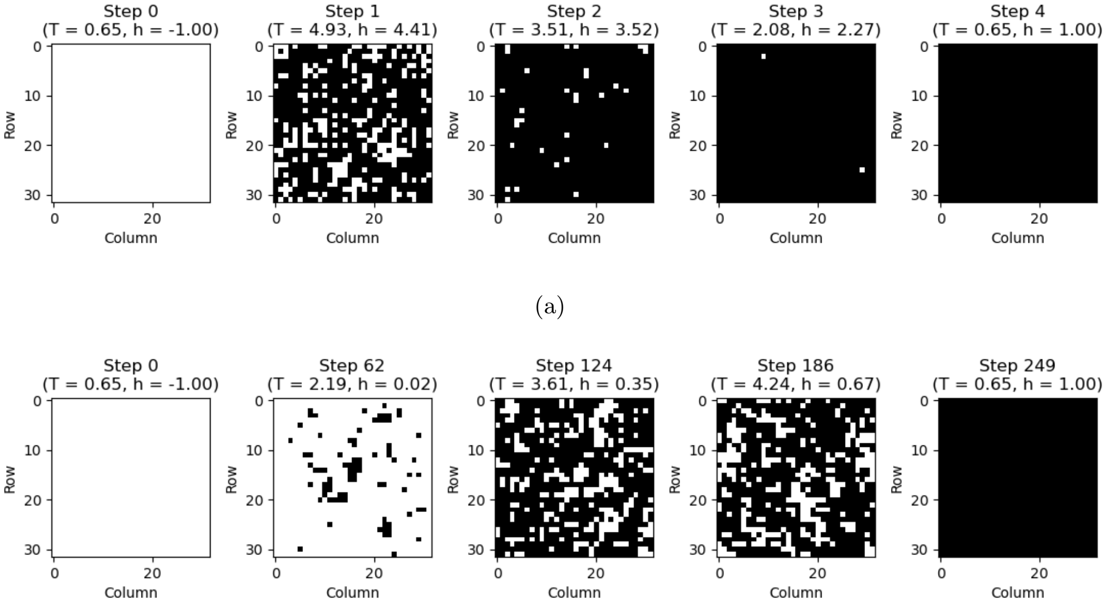
Additionally, a population of 50 genomes was optimized for 100 generations for different trajectory lengths. Analysis of this data, as evidenced in Fig. 4, reveals a clear effect of trajectory length on the process of magnetization reversal. Longer control protocols, exemplified by Fig. 4b, demonstrate a more gradual approach to achieving the magnetization flip compared to shorter protocols, as seen in Fig. 4a.



(a)

(b)

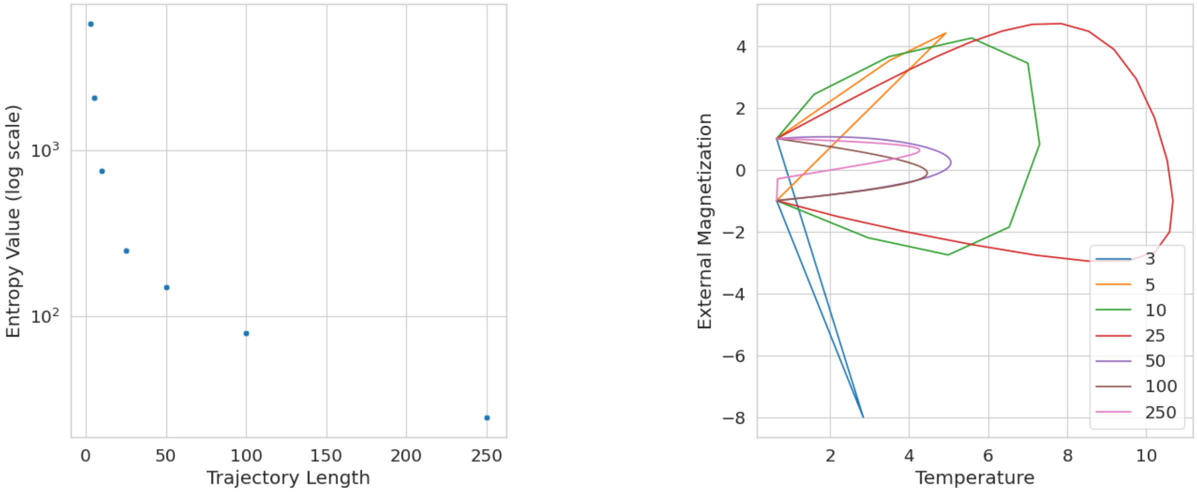
Figure 3: (a) Single control protocol trajectory demonstrating magnetization flip after 100 generations. T represents temperature, h represents external magnetization, and m represents internal magnetization. (b) Single control protocol trajectories illustrating the relationship between temperature and external magnetization for different numbers of training generations (represented in the legend)



(a)

(b)

Figure 4: Spatial evolution of lattices under different control protocols. Fig 4a shows the evolution of a lattice with a control trajectory length of 5. Fig. 4b shows the evolution of a lattice with a control trajectory length of 250.



(a)

(b)

Figure 5: (a) Pareto front representing the relationship between entropy value (on a logarithmic scale) and trajectory length for a bit flip in an Ising system. Each point is the average entropy generated by the trained protocol over 100 trajectory repeats. (b) Single control protocol trajectories illustrating the relationship between temperature and external magnetization for different control trajectory lengths (represented in the legend)

This careful modulation of control time showcases the inherent entropy tradeoff: with increased duration, the system benefits from a more controlled evolution, resulting in less entropy expenditure. This tradeoff is graphically articulated in the data represented in Fig. 5a, presenting, to our knowledge, the first example of a Pareto front in this system that articulates the inverse relationship between entropy generation and control time. Note that even protocols with short control intervals learn to avoid the critical temperature, in Fig. 5b.

While NEAT has proven effective for solving single objective control problems in nanoscale systems, it is not well-suited for addressing multi-objective problems due to sample inefficiency. This limitation is evident in our application of NEAT for generating a Pareto front. Despite successful generation, the result is noticeably sparse, implying NEAT's inefficiency in dealing with multi-objective problems.

6 Discussion

Our study leveraged the capabilities of reinforcement learning (RL) to approach the problem of minimizing energy dissipation in the process of magnetization reversal in a two-dimensional Ising model. By employing a Constrained Deep RL algorithm, specifically Proximal Policy Optimization (PPO), and contrasting it with the evolutionary-based Neuroevolution of Augmented Topologies (NEAT), we were able to explore different methodological avenues for optimizing this task. Our results indicate that the application of constraints within the PPO framework, especially the shear transformation, enabled the emergence of an agent capable of learning a protocol to flip the magnetization from the ground state ($m = -1$) to the reversed state ($m = 1$). Although the RL agent managed to outperform the naive protocol by reducing entropy production significantly, the final entropy values achieved remained suboptimal compared to those theoretically possible. The plateau in entropy reduction during the course of PPO training suggests the emergence of a local minimum in the optimization landscape. We suspect the problem is likely some combinations of the following factors. One is the sparse reward nature of this problem is hindering the performance (e.g. only at the end of the trajectory do we know the overall performance). Secondly, we suspect that despite the entropy regularization, the algorithm is not balancing exploration and exploitation sufficiently. In future work we would like to implement alternative exploration bonuses.

Recognizing the success of reinforcement learning in establishing fundamental control policies yet acknowledging its challenges in efficiently navigating the complex energy landscape of the Ising model, we turned to NEAT as an alternative solution. NEAT's evolutionary approach is a robust method capable of yielding innovative and diverse control strategies. In contrast to PPO's incremental policy updates, NEAT employs a different strategy by evolving both network topology and weights through a process of natural selection. Our data illustrates that NEAT successfully balances the exploration of the control space with the exploitation of effective policies, as evidenced by the generation of a range of entropy values over different trajectory lengths. Furthermore, our analysis shows a clear Pareto relationship between entropy and trajectory length; an extended control time allows for a more nuanced and energy-efficient approach to guiding the system towards the reversed magnetization state. Nevertheless, despite NEAT's demonstrated proficiency in optimizing single-objective problems, the sparse nature of the produced Pareto front in our multi-objective setting highlights a challenge. The genetic algorithm illustrates sample inefficiency when tasked with balancing multiple objectives, an area where other multi-objective optimization approaches might offer advantages.

In summary, our study juxtaposes two distinct ML approaches to minimize energy dissipation in a nanomagnetic system. RL, with constraint enforcement, shows promise in learning non-trivial protocols; however, it requires further refinement to achieve near-optimal performance. Meanwhile, NEAT's evolutionary approach excels in navigating single-objective landscapes but may need adaptation or hybridization for handling multi-objective scenarios efficiently. The divergence in the strategies and outcomes of these methods underlines the complexity of the problem and the potential for future research to enhance the application of ML in material science and nanotechnology. As energy efficiency becomes increasingly paramount in data storage and other applications, advancements in this domain could have far-reaching implications.

7 Appendix

7.1 PPO

PPO (Proximal Policy Optimization) is a policy gradient method that is commonly used in RL problems due to it having elements of Trust Region Policy Optimization (TRPO), but also being more sample efficient and easier to implement (due to having fewer hyperparameters).

It builds upon the policy gradient algorithm and retains the first order optimization property, yet has a similar performance with TRPO with the inclusion of the clipped surrogate objective. This allows a lower bound estimate of the policy. In the formulation, the probability ratio is "clipped" between $1 - \epsilon$ and $1 + \epsilon$ where ϵ is a hyperparameter, and the minimum between the policy and the clipped policy is taken [6].

By combining the above with the squared error loss and an entropy bonus, the loss objective of PPO can be constructed.

Algorithm 1 PPO (Proximal Policy Optimization) in the magnetization reversal context

```
for Iteration i = 1, ..., N do
    for Actor j = 1, ..., K do
        for Timestep t = 1, ... ,T do
            Run $\theta_{old}$
            Compute advantage $\hat{A}_t$ with truncated GAE
        Compute clipped surrogate objective
        Compute squared error loss
        Compute entropy bonus
        Compute Total loss/surrogate
        Optimize L w.r.t $\theta$ using minibatch SGD/Adam
        Update $\theta_{old} \leftarrow \theta$
```

7.2 NEAT

NeuroEvolution of Augmenting Topologies (NEAT) is an innovative algorithm for evolving artificial neural networks that adeptly integrates the evolution of network topology with the optimization of connectivity weights. This method is distinctive in its capacity to unearth both simple and sophisticated solutions tailored to a wide array of tasks. In stark contrast to other evolutionary techniques that commence with predetermined or randomly complex network architectures, NEAT begins with strikingly simple networks and incrementally enriches these structures with new nodes and connections through mutations over the course of generations. Utilizing genetic operators such as selection, crossover, and mutation, NEAT fosters the gradual emergence of complex behaviors and network topologies that are adapted to the task at hand.

The evolutionary process underpinning NEAT is orchestrated by several foundational principles. The concept of complexification is central to NEAT, as it allows the algorithm to start with uncomplicated initial networks and increase their complexity progressively, introducing new nodes and connections only as necessary through strategically targeted mutations.

Speciation stands out as a pivotal feature in NEAT that confers protection to novel mutations by clustering similar genomes into species, thus safeguarding diversity within the population and nurturing the refinement and optimization of innovative traits. This strategy is key in averting premature convergence and helps to ensure that new advantageous adaptations are retained and selectively amplified over time. The incorporation of fitness sharing within each species acts to prevent any individual genome from establishing dominance and promotes the exploration of distinct evolutionary strategies. This element encourages the probing of different niches within the vast search landscape and facilitates a broad and thorough exploration of the solution space.

The selection process inherent in NEAT guarantees that the fittest genomes within each species are preferentially chosen for reproduction, engendering a steady enhancement in the overall population's performance. Note that we evaluate each genome multiple times and return the averaged reward to select the best species, as in [5], in order to address the stochasticity of the system. Additionally, NEAT leverages mutation, which breathes new life into the genome pool by adding fresh traits, and crossover, which recombines extant traits, fostering greater genetic diversity. Historical markings hold a crucial role in facilitating coherent crossover between genetically distinct entities.

Integrating these principles into a cohesive evolutionary framework allows NEAT to excel at finding optimal network structures for tasks where a predetermined network topology is absent. The algorithm's evolutionary path mirrors the principles of natural selection and is proficient in cultivating highly specialized neural structures that are both effective and efficient for their intended environments. This evolutionary journey marked by NEAT is characterized by a gradual and organic amplification in network sophistication, resulting in the natural selection of increasingly fit neural network structures for the problems they are designed to solve. For additional details of how NEAT operates, pseudocode of the algorithm is below and more details can be found in [4].

Algorithm 2 NEAT: This algorithm iteratively refines a population of minimal neural networks through speciation, selection, crossover, and mutation based on their performance, with the choice of fitness function allowing for either novelty or objective-based search

```
$population \leftarrow$ InitializePopulation($X$, $a$)
$bestPredictions \leftarrow \emptyset$
$speciesAvgFitness \leftarrow \emptyset$
**for** $generation = 1$ to $Y$ **do**
    $species \leftarrow$ Speciate($population$)
    $fitnessList \leftarrow \emptyset$
    $performanceList \leftarrow \emptyset$
    **for** each $genome$ in $population$ **do**
        $controller \leftarrow$ Control($genome$, $sys$)
        $fitness, performance \leftarrow$ Fitness($controller$), Performance($controller$)
        $fitnessList$.append($fitness$)
        $performanceList$.append($performance$)
    $bestPredictions$.append($population$[argmax($performanceList$)])
    **for** each $specie$ in $species$ **do**
        $avgFitness \leftarrow$ mean(FitnessesOf($specie$))
        $speciesAvgFitness$.append($avgFitness$)
        **if** NotImproving($avgFitness$) **then**
            RemoveSpecies($population$, $specie$)
    Mutate($population$[argmax(PerformanceOfSpecies($specie$))])
    **if** $generation = Y$ **then**
        **return** $bestPredictions$
$performanceMatrix \leftarrow$ InitializeMatrix(size=$M\times$ len($bestPredictions$))
**for** $trial = 1$ to $M$ **do**
    **for** $index$, $prediction$ in enumerate($bestPredictions$) **do**
        $controller \leftarrow$ Control($prediction$, $sys$)
        $performanceMatrix[index][trial] \leftarrow$ Performance($controller$)
**return** $population$[argmax(mean($performanceMatrix$, axis=1))]
```

References

[1] B. Lambson, D. Carlton, and J. Bokor, "Exploring the thermodynamic limits of computation in integrated systems: Magnetic memory, nanomagnetic logic, and the landauer limit," *Phys. Rev. Lett.*, vol. 107, p. 010604, Jul 2011. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.107.010604

[2] G. M. Rotskoff and G. E. Crooks, "Optimal control in nonequilibrium systems: Dynamic riemannian geometry of the ising model," *Phys. Rev. E*, vol. 92, p. 060102, Dec 2015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.92.060102

[3] S. Whitelam, "Demon in the machine: Learning to extract work and absorb entropy from fluctuating nanosystems," *Phys. Rev. X*, vol. 13, p. 021005, Apr 2023. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.13.021005

[4] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[5] J. O'Leary, M. M. Khare, and A. Mesbah, "Novelty search for neuroevolutionary reinforcement learning of deceptive systems: An application to control of colloidal self-assembly," in *Proceedings of the American Control Conference*, San Diego, 2023, pp. 2776–2781.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.