Ising Networks for Deep Hierarchical Reinforcement Learning

1 Notation

Reinforcement Learning: We review the RL setup wherein an agent interacts with the environment in order to transition to new states and observe rewards by following a sequence of actions. The problem is modeled as a finite-horizon Markov Decision Process(MDP) [66] defined by the tuple $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$ where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are continuous, r presents the reward observed by agent such that $r : \mathcal{S} \times \mathcal{A} \to [r_{min}, r_{max}]$, $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$ presents the unknown transition model consisting of the transition probability to the next state $s_{t+1} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$ at time step t and γ is the discount factor. We consider a policy $\pi_\theta(a_t|s_t)$ as a function of model parameters θ. Standard RL defines the agent's objective to maximize the expected discounted reward $\mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$ as a function of the parameters θ. The action-value function for an agent is represented as $Q(a, s; \theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{T} \gamma^t r(s, a)|s = s_t, a = a_t]$ which is the expected sum of payoffs obtained in state s upon performing action a by following the policy π_θ. We denote the optimal policy π^*_θ such that $Q(a, s; \theta^*) \geq Q(a, s; \theta) \forall s \in S, a \in A$.

Multi-Agent Learning: We now review the cooperative MARL setup. The problem is modeled as a Partially Observable Markov Decision Process (POMDP) [66] defined by the tuple $(\mathcal{S}, \mathcal{A}, r, C, P, \mathcal{Z}, O, \gamma)$ wherein the notations are consistent with the single-agent RL MDP setup with the state space $\mathcal{S}$ and action space $\mathcal{A}$ being discrete and r presenting the reward observed by agents $u \in C$ where C is the collective set of all agents. Note that we use alternate notations of $s' \in \mathcal{S}$ for s_{t+1} and $a' \in \mathcal{S}$ for a_{t+1}. We consider a partially observable setting in which each agent u draws individual observations $z \in \mathcal{Z}$ according to the observation function $O(s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$. Similar to the single-agent RL case, we denote the optimal policy π^*_θ such that $Q(a, s; \theta^*) \geq Q(a, s; \theta) \forall s \in S, a \in A$. In the case of multiple agents, the joint optimal policy can be expressed as the Nash Equilibrium [**?**] of the Stochastic Markov Game as $\pi^* = (\pi^{1,*}, \pi^{2,*}, ...\pi^{C,*})$ such that $Q(a^u, s; \theta^*) \geq Q(a^u, s; \theta) \forall s \in S, a \in A, u \in C$.

2 Introduction

3 Related Work

3.1 Hierarchical Reinforcement Learning

3.1.1 Temporal Abstraction

Various methods have devised hierarchical techniques to abstract the temporal correlation in MDPs [33, 34, 52, 66]. The Hierarchical Deep Q-Network (hDQN) [33] proposes an integration of temporal abstraction and intrinsic motivation in order to solve long-horizon problems. hDQN demonstrates improved long-horizon optimality at the cost of manual feature engineering of the states and reward function. This is addressed by making use of deep successor networks [34] which extract relevant goal embeddings for the agent and direct its behavior towards the goal. A more sophisticated temporal scheme consists of learning abstraction of hierarchies based on expert demonstrations and regularized latent space constraints [74]. However, sub-policies rely on memory-based controllers for learning fine-grained abstraction and often cripple the lower levels of the hierarchy once the task has been accomplished. This leads to limited transfer of skills between policies and hence, results in sub-optimal convergence [14]. Transfer of skills between different policies operating in sub-MDPs is extensively studied using exploration [30]. [44] presents MAVEN, a hierarchical architecture for variational exploration in the case of multi-agent settings. Policies carry out exploration in the latent space using variational inference. MAVEN demonstrates improved coordination between agents in the long-horizon. However, policy behavior distributed across agents which forces sub-policies to rely on other agent policies for optimal convergence. [9] proposes the trajectory autoencoder which enables the agent to exhibit self-consistent behavior and tackle sparse rewards in the long-horizon. Long-horizon suitability can further be improved by jointly training all policies and utilize the skills learned individually [37]. Such kind of sub-policy adaptation stabilizes the behavior of agent and provides efficient training of hierarchies at all levels.

Abstraction may be carried out among states [2] or using a different medium such as language [29] which allows the agent to solve temporally-extended tasks in a diverse manner. [12] presents the MAXQ algorithm which proposes an efficient method for training state-based abstractions in the form of hierarchies. Q-values corresponding to each lower-level task are aggregated at the higher-level policy nodes to yield a modular framework for abstraction. While the MAXQ demonstrates suitability for low-dimensional tasks, its performance cannot be assessed in high-dimensional state and action spaces due to the larger number of abstractions produced by the framework. [1] tackles the problem of high-dimensional and long-horizon learning by presenting a theoretical perspective of abstracting states using transitive behavior which helps in optimal computations. Such techniques have motivated active research in the form of modified

abstraction algorithms for performance improvement and schemes leading to richer abstract representations [40].

3.1.2 Option-based Hierarchies

Hierarchies can be constructed on the basis of skills or possible options of policies which the agent may choose from during its interactions in the environment [66, 4]. [4] presents the Options-Critic Architecture which is a collection of various policy options. Agent can select from the collection of sub-policies at various timesteps to yield a policy which is robust and optimal in the long-horizon. The provision of options enables the agent to acquire a wide variety of skills during exploration and obtain a policy which makes best use of these skills. Furthermore, the architecture incorporates learning of termination conditions of a particular option which is essential for switching option policies [22]. The architecture is extended to the Double Actor-Critic architecture which comprises of two parallel MDPs [81]. A dual architecture enables the learning of intra-option policies with termination conditions of each option. While, option-based learning provides diversity in skills and control of temporal extension to the agent, frequent fluctuations between options results in instability in updates and hence, sample-inefficient learning [38]. [38] addresses the instability in training by combining the options framework with off-policy maximum entropy reinforcement learning [19, 18]. Maximization of entropy allows the agent to effectively explore all options and consistently select the optimal sub-policies based on selected options. However, usage of different skills and compactness remains an open problem in the case of real-world tasks consisting of large action spaces. [50] addresses the problem of compact options by improvising communication between policies using binary vectors. Modular policies using binary vectors enable richer communication and mixing of skills. Techniques related to the relative optimization of policy can be extended using advantage-weighted importance sampling [49] in options based on option-value functions.

3.1.3 Hierarchical Control

Hierarchical Reinforcement Learning borrows from multiple approaches in control [51] and adaptive policy optimization [37]. Multiplicative Compositional Policies (MCPs) [51] present the composable nature of hierarchies which arises from coordination of multiple control-based skills. MCPs improve the hierarchical coordination between sub-policies which results in adaptive motor skills in the case of robotic control such as locomotion and manipulation. While MPC demonstrates the practical usage of hierarchical reinforcement learning motivated by adaptive control, [28] extends the hierarchical framework to Quadruped Locomotion. The lower-level policy uses latent commands for robot actuator control and the higher-level policy operates at different time intervals. Abstraction of control between lower and higher-level policies allows the agent to acquire a diverse set of task-related skills. Moreover, policies can be efficiently adapted on

a different task from the same domain. Additionally, diversity can be achieved in a scalable manner by training policies levelwise [60]. However, levelwise training of sub-policies becomes intractable in cases of a large number of hierarchies [17]. [17] tackles the problem of a large number of hierarchies and levelwise training by making use of latent space policies. Each latent space policy is trained using the maximum entropy learning framework in order to directly solve the task. Hierarchies trained in latent space demonstrate suitable initializations on complex control tasks which enable the agent to maintain consistent behavior. Consistency can also be achieved using goal embeddings which motivate goal-directed behavior in the case of navigation tasks [62]. [71] additionally improves consistency in hierarchical agents by making use of information theoretic regularizations which pose a reward penalty on the policy distribution. Hierarchical control in reinforcement learning has been successful in tackling knowledge transfer as well as data-efficiency [78]. Various levels of the hierarchy make use of inductive-biases for sharing off-policy data. Sharing knowledge across policies in a compositional manner results in positive transfer of policies. Such techniques are additionally used to solve numerically challenging control problems consisting of unstructured environments [42].

3.2 Ising Models

3.2.1 Energy-based Reinforcement Learning

Recent work on energy-based learning [36] has demonstrated significant success in the arena of reinforcement learning [61, 35]. [56] highlights the use of energy-based policies capable of learning large action spaces by making use of factored approximations. Such approximations can be thought of as inference mechanisms [43] using energy-based distributions such as the boltzmann distribution [69]. These have given rise to energy-based extensions of models with different architectures such as actor-critic frameworks [23]. Another suitable method for training energy-based policies in reinforcement learning is by making use of the Helmholtz free energy [18]. [**?**] presents the suitability of improved helmholtz energy operator under various reinforcement learning scenarios with theoretical guarantees on robustness and improved inference in comparison to boltzmann energy. Applications of the energy operator include surprise minimization in the case of multi-agent learning [65]. [65] highlights the practical use of energy in multi-agent learning across agents which improves the joint policy to tackle environments consisting of high variance. The use of energy-based policies can further be generalized to inverse reinforcement learning [13] wherein the maximum entropy framework can be realized as a special case of energy-based models in the generative setting. Although energy plays a key role the training dynamics of reinforcement learning and optimization of the agent to an optimal policy, it does little to improve the computational framework of learning hierarchical relationships in agent's policy [47]. This indicates the requirement for a hierarchical framework capable of mapping relationships between sub-policies in the hierarchy.

3.2.2 Ising Model Learning

Ising models [8] are computational frameworks in physics which consist of interactions between particles in a system or lattice. Particles in the model comprise of finite spin states [10] which governs their relationship to neighboring particles in the model. Ising models obey an energy function [8] which is minimized over time as the model reaches its equilibrium state. Most energy-based models are generalizations of Ising models with their objective function as the loss function [27]. The motivation behind developing Ising models as computational frameworks stems from the literature of physical lattices and spins [8, 10, 27, 31]. [31] presents the first major demonstration of behavior of Ising models near criticality. Assessing the model near criticality is essential as it describes convergence properties corresponding to free energy of the system. This is further extended in [27] wherein the non-equilibrium critical relaxation of the model is assessed under varying dynamics. Estimation of high-dimensional criticality provides suitable scope of the model towards large-scale computational problems. To further investigate the properties of Ising models, [76] highlights the theoretical guarantees of spin-spin correlations under zero field susceptibility. [76] makes use of simplified elliptical transformations to demonstrate criticality of the model. While the Ising model presents suitable convergence properties [26] in the number of particles and time evolution, the model is itself computationally expensive [24]. Manually estimating the state of the model leads to accuracy and efficient inference techniques at the cost of time complexity to assess the state of the system. [24] presents a fast and efficient method for the simulation of Ising models which relies on non-linear relaxation. However, assumptions on spin states and their neighbors renders the state estimates inaccurate and provides a significant need for sampling. [58] addresses the computational problem in Ising structures using maximum-margin parameter estimation and graphical models.

Most works in literature aim to learn the behavior [46] and parameters [41] of the Ising model for particle spin state estimation. [46] presents the learning of Ising model behavior at criticality near phase transitions using generative networks. The approach highlights the requirement of various architectural choices and draws a comparison with shallow counterparts of Restricted Boltzmann Machine (RBM). [41] demonstrates the learning of optimal structure and parameters of the Ising model using interactive screening. Interactive screening is presented as a tractable and optima estimation method which solves the inverse Ising problem universally. [3] presents the application of RBM to reconstruct long-range Ising model configurations in one dimension. Configuration quality in generative scenarios for long-range structures can be assessed by making use of field susceptibility. While learning the Ising model is essential from a structural perspective, the process does not benefit from the computational properties of spin states [72] and their arrangement at thermal equilibrium. One of the successful examples demonstrating the use of Ising models in learning algorithms is that of correlating spiking activities in neurons [72]. The model, consisting of maximum entropy, can be used to reproduce pairwise correlations

between particle spins which is found to be analogous to correlations in the spiking activity of neurons. Moreover, the networks are found to operate closer to critical point and demonstrate behavior similar to spin glasses. [80] further highlights the use of Ising models as learning mechanisms by transforming spin structures to Boltzmann Machines. The method is generalized to many-spin interactions and exact mapping which can be improved to fewer spin states by compromising on the number of degrees of freedom. [72] and [80] present some of the few instances of learning capabilities of Ising models in the case of unstructured environments. Although the methods are suitable for supervised tasks and modular structures, no concrete evidence of spin-based learning in hierarchical structures is found.

4 The Ising Model

The Ising Model [8] is an energy-based model which consists of physical particles in a lattice structure. Particles acquire spin states in the lattice which affect the states of neighbouring particles as well as the overall configuration of the system [10]. Consider an Ising model consisting of N particles with $N \geq 1$ and $N \neq \infty$. The set of all particles can then be represented as $\{1, 2, ..., N\}$. Mathematically, let σ_i be the spin state of particle i in the system, then σ_i may take a value from the set $\mathcal{S}$ which comprises of all the possible spin states. Most Ising models are bivariate which consist of only two spin states $\mathcal{S} = \{-1, +1\}$ [31]. In physics, the two spin states correspond to clockwise and counter-clockwise spins of particles of a ferromagnetic material [8]. However, modern Ising models are often presented to contain more than two spin states [27]. Particles in an Ising model interact with each other by means of their spin states. Consider two particles i and j with spin states σ_i and σ_j. The interaction between the two particles $< \sigma_i, \sigma_j >= \mu_{ij}$ can be mathematically expressed as $\mu_{ij} = \sigma_i.\sigma_j$. These interactions are also called spin-spin interactions since they take place between spin states of the two particles [10]. Considering an external magnetic field h_j [76] on particle j, we can express the total energy E of the Ising model using the Hamiltonian function [21] as followed-

$$
\begin{aligned}
E(\sigma_i, \sigma_j) &= -\sum_{<i,j>} J_{ij}\sigma_i\sigma_j - \sum_j h_j\sigma_j, \qquad \forall i, j \in \{1, 2, ..., N\} \\
&= \sum_{<i,j>} J_{ij}\mu_{ij} - \sum_j h_j\sigma_j \qquad \forall i, j \in \{1, 2, ..., N\}
\end{aligned}
$$

Here, $< i, j >$ represent the indices of all particles in the Ising model, J_{ij} indicates an interaction parameter which quantifies the probability of interaction between i and j particles. Note that the total energy $E(\sigma_i, \sigma_j)$ is a function of only the spin states of particles indicating that only the internal composition of the lattice is responsible for minimizing the energy of the system [70]. Moreover, the partition function Z_β [36] corresponding to the energy function is given by

Equation 1. Here $\beta = \frac{1}{T}$ represents the inverse of temperature T.

$$Z_\beta = \sum_{\sigma_i, \sigma_j} \exp\left(-\beta E(\sigma_i, \sigma_j)\right) \qquad \forall i, j \in \{1, 2, ..., N\} \tag{1}$$

Given the energy $E(\sigma_i, \sigma_j)$ and partition function Z_β corresponding to the states of the system, we can express the configuration probability $P_\beta(\sigma_i, \sigma_j)$ of the model using the Boltzmann distribution [21] with $\beta \geq 0$. $P_\beta(\sigma_i, \sigma_j)$ for a given configuration is presented in Equation 2 as the ratio between the configuration $\exp\left(-\beta E(\sigma_i, \sigma_j)\right)$ and the partition function Z_β denoting the sum over all possible configurations. The negative sign in the exponent denotes assigns a higher probability to low energy states. Similarly, a high β value accentuates the probabilities of the low energy states. Upon carrying out simulated annealing [43] of T closer to the low energy states, the system tends to reach thermal equilibrium.

$$P_\beta(\sigma_i, \sigma_j) = \frac{\exp\left(-\beta E(\sigma_i, \sigma_j)\right)}{Z_\beta} \qquad \forall i, j \in \{1, 2, ..., N\} \tag{2}$$

The partition function aids in the assessment of necessary thermodynamical and computational quantities such as internal energy U which is defined as the negative derivative of the partition function Z_β [75]. U is mathematically expressed in .

$$U = -\frac{\partial Z}{\partial \beta} = -\frac{1}{Z} \sum_{\sigma_i \sigma_j} \exp\left(-\beta E(\sigma_i, \sigma_j)\right) \qquad \forall i, j \in \{1, 2, ..., N\} \tag{3}$$

Similarly, the free energy per particle F can be obtained as the average of log partition $\log Z_\beta$ in the limit of the number of particles in the model [70]. This is mathematically expressed in Equation 4.

$$F = F(\beta, E, N) = \lim N \to \infty \frac{1}{N} \log Z_\beta(E, N) \tag{4}$$

The limit $N \to \infty$ in Equation 4 is called the thermodynamic limit. In practical scenarios, computation of the free energy F is non-trivial as there is no guarantee of the existence of thermodynamic limit $N \to \infty$ as the lattice system can grow at different rates and in different directions. Such constraints on the growth and size of the system hinder a closed form expression for F and hence pose restrictions on the solution of the lattice. Moreover, the structure and complexity of the model increases with the number dimensions [58] which further makes simulations intractable and inaccurate by making use of approximations. As a result of this, most modern computational frameworks [80] make use of the energy and partition functions in an indirect manner by making use of structures which can be realized and yield probabilistically approximate estimates [69].

5 Ising Networks

5.1 The Spin-based Objective

5.2 Learning Spin Values of Hierarchies

6 Intuition for Spin Values

7 Implementation Details

This section highlights the implementation details for THE Ising networks framework when combined with different RL methods. Corresponind to each setting, we first discuss the baseline implementations and their hyperparameters followed by the details of the hierarchical Ising networks.

7.1 Continuous Control

We first look at the continuous control setting wherein our action space $\mathcal{A}$ is continuous. We carry out our experiments in the MuJoCo [73] and DM Control Suite [68] domains. Each action in these domains is bounded in the $[-1, 1]$ which depicts the force or torque require to control the agent. We divide our experiments into two schemes- (1) state-based learning and (2) learning from pixels. In the first scheme the state input provided to the agent is a feature vector consisting of the agent's position and velocity while in the second scheme, the agent is presented with an image of its current setting. Learning from pixels has proven to be difficult in literature [68, 20] and is an active area of work [35, 61].

7.1.1 State-based Learning

In the case of state-based learning we combine the IS framework with SAC [19] which is a state-of-the-art algorithm for off-policy continuous control RL. Various extensions of SAC have proven to be sample-efficient [64] which we consider in our experiments for comparison. We compare our IS-SAC implementation with ESAC [64], SAC [19], TD3 [16] which is an improved version of DDPG [39], PPO [59] and ES [55]. Note that we compare our algorithm to on-policy, off-policy and evolutionary methods to assess the capability of IS-SAC w.r.t each type of RL agent. All agents were trained on a total of 21 environments (9 MuJoCo tasks and 12 DM Control tasks) in the OpenAI Gym Suite [6] for a total of 5 random seeds.

Baselines: Our experiments make use of author-provided baseline implementations for ES [55], [16], [19] and [64]. In the case of PPO [59] we use the implementation provided in OpenAI baselines [11] as it demonstrates consistent results. Moreover, we omit the Virtual BatchNorm [54] component from ES as it is often found to hinder scalability and does not affect the performance. All implementations make use of the same architecture consisting of two hidden

layers of 1024 units with ReLU non-linearity [48]. The output layer makes use of a Tanh activation in order to bound the actions in the $[-1, 1]$ range.

IS-SAC:

7.1.2 Learning from Pixels

In the case of learning from pixels we combine our approach with SAC+AE [79] and denote it with IS-SAC+AE for our experiments. We compare between IS-SAC+AE and SAC+AE on the 100k benchmark [61] for a total of 5 random seeds on 12 DM control suit environments. The 100k benchmark evaluates the algorithm for only 100k timesteps and assesses data-efficiency in pixel-based learning.

Baseline: Our baseline implementation of SAC+AE makes use of the author-provided implementation [79] with tuned hyperparameter.

IS-SAC+AE:

7.2 Discrete Control

Our experiments in discrete control make use of two learning domains- (1) Atari 2600 suite in OpenAI Gym [6] and (2) Mazelab [82]. We make use of the Atari 2600 suite to present the large-scale generalization of hierarchies utilizing spin-spin alignments in the case of long-horizon environments. On the other hand, we use Mazelab to demonstrate their success in sparse environments consisting of limited optimal behavior.

7.2.1 Atari 2600 Games

In the case of Atari 2600 tasks, we train our agents on 28 games for a total of 5 random seeds. All agents were trained for 5 million steps with learning starting after 10k steps. We combine our framework with Rainbow [25] which is a state-of-the-art model-free off-policy RL algorithm utilizing the original DQN [25] implementation. and denote it with IS-Rainbow for our experiments. IS-Rainbow is compared to Rainbow [25], A2C [45], PPO [59] and ACKTR [77]. Additionally, we present the scores of an average human subject and a random agent for comparison and reference.

Baselines: Our implementation of Rainbow is based on [61]. However, we limit the replay memory to the most recent 10^6 steps for a fair comparison with other methods. In the case of A2C, PPO and ACKTR, we make use of the implementations provided in [32] as these are found to be stable with respect to random seeds in comparison to OpenAI baseline implementations [11]. Agents are tuned to their optimal hyperparameter values as per author-provided details.

IS-Rainbow:

7.2.2 Mazelab

Baselines:

IS-hDQN:

7.3 Multi-Agent Learning

We select StarCraft II scenarios particularly for three reasons. Firstly, micromanagement scenarios consist of a larger number of agents with different action spaces. This requires a greater deal of coordination. Secondly, micromanagement scenarios consist of partial observability wherein agents are restricted from responding to enemy fire and attacking enemies when they are in range. This allows agents to explore the environment effectively and find an optimal strategy purely based on collaboration rather than built-in game utilities. Lastly, micromanagement scenarios in StarCraft II consist of multiple opponents which introduce a greater degree of surprise within consecutive states. Irrespective of the time evolution of an episode, environment dynamics of each scenario change rapidly as the agents need to respond to enemy's behavior. Agents were trained for a total of 5 random seeds consisting of 2 million steps in each environment. A total of 32 validation episodes carried out at every 10,000 step intervals were interleaved during agent's interactions. All baselines implementation consist of a Recurrent Neural Network (RNN) agent having memory consisting of past states and actions. We use an ϵ-greedy exploration scheme wherein ϵ is annealed from 1 to 0.01 during the initial stages of training.

Baselines: Our baselines consist of state-of-the-art MARL methods for cooperative control. We compare our framework to EMIX [65], QMIX [53], [63], [15] and [67]. Additionally, we use a model-free implementation of SMiRL [5] combined with QMIX to assess surprise-robust behavior of our framework. We denote this implementation as SMiRL-QMIX for our experiments. All implementations were adopted from the PyMARL [57] framework with hyperparameters tuned to their optimal parameters using author-provided notes. In the case of EMIX, we tune the the temperature parameter β to 0.03 as it provides us with consistent results across different seeds. We make use of the generalized implementation of SMiRL [7] in SMiRL-QMIX wherein we utilize the standard deviations of states across batches in order to reduce variance in rewards. Moreover, the temperature parameter in SMiRL-QMIX is tuned to 0.1 in steps of 0.02.

IS-EMIX:

References

[1] D. Abel, D. Arumugam, L. Lehnert, and M. Littman. State abstractions for lifelong reinforcement learning. In *International Conference on Machine Learning*, 2018.

[2] D. Andre and S. J. Russell. State abstraction for programmable reinforcement learning agents. In *AAAI/IAAI*, 2002.

[3] K.-I. Aoki and T. Kobayashi. Restricted boltzmann machines for the long range ising models. *Modern Physics Letters B*, 2016.

[4] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture, 2016.

[5] G. Berseth, D. Geng, C. Devin, D. Jayaraman, C. Finn, and S. Levine. Smirl: Surprise minimizing rl in entropic environments. 2019.

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[7] J. Z. Chen. Reinforcement learning generalization with surprise minimization, 2020.

[8] B. A. Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 1987.

[9] J. D. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.

[10] G. Delfino and G. Mussardo. The spin-spin correlation function in the two-dimensional ising model in a magnetic field at t= tc. *Nuclear Physics B*, 1995.

[11] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov. Openai baselines, 2017.

[12] T. G. Dietterich. State abstraction in maxq hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 2000.

[13] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models, 2016.

[14] C. Florensa, Y. Duan, and P. Abbeel. Stochastic neural networks for hierarchical reinforcement learning, 2017.

[15] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients, 2017.

[16] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. *CoRR*, abs/1802.09477, 2018.

[17] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine. Latent space policies for hierarchical reinforcement learning, 2018.

[18] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies, 2017.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[20] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[21] L. N. Hand and J. D. Finch. *Analytical Mechanics*. Cambridge University Press, 1998.

[22] A. Harutyunyan, W. Dabney, D. Borsa, N. Heess, R. Munos, and D. Precup. The termination critic, 2019.

[23] N. Heess, D. Silver, and Y. W. Teh. Actor-critic reinforcement learning with energy-based policies. Proceedings of Machine Learning Research, 2013.

[24] H. Herrmann. Fast algorithm for the simulation of ising models. *Journal of statistical physics*, 1986.

[25] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.

[26] J. Honorio. Convergence rates of biased stochastic optimization for learning sparse ising models, 2012.

[27] N. Ito. Non-equilibrium relaxation and interface energy of the ising model. *Physica A: Statistical Mechanics and its Applications*, 1993.

[28] D. Jain, A. Iscen, and K. Caluwaerts. Hierarchical reinforcement learning for quadruped locomotion. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[29] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn. Language as an abstraction for hierarchical deep reinforcement learning. In *Advances in Neural Information Processing Systems 32*. 2019.

[30] N. K. Jong, T. Hester, and P. Stone. The utility of temporal abstraction in reinforcement learning. In *AAMAS (1)*, 2008.

[31] L. Kadanoff. Scaling laws for Ising models near T(c). 1966.

[32] I. Kostrikov. Pytorch implementations of reinforcement learning algorithms, 2018.

[33] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, and J. B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation, 2016.

[34] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman. Deep successor reinforcement learning, 2016.

[35] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data, 2020.

[36] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and et al. A tutorial on energy-based learning. In *PREDICTING STRUCTURED DATA*, 2006.

[37] A. Li, C. Florensa, I. Clavera, and P. Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2020.

[38] C. Li, X. Ma, C. Zhang, J. Yang, L. Xia, and Q. Zhao. Soac: The soft option actor-critic architecture, 2020.

[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

[40] S. Lin and R. Wright. Evolutionary tile coding: An automated state abstraction algorithm for reinforcement learning. In *Proceedings of the 8th AAAI Conference on Abstraction, Reformulation, and Approximation*, 2010.

[41] A. Y. Lokhov, M. Vuffray, S. Misra, and M. Chertkov. Optimal structure and parameter learning of ising models. *Science advances*, 2018.

[42] Q. Ma, S. Ge, D. He, D. Thaker, and I. Drori. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning, 2019.

[43] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[44] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems 32*. 2019.

[45] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 2016.

[46] A. Morningstar and R. G. Melko. Deep learning the ising model near criticality. *J. Mach. Learn. Res.*, 2017.

[47] O. Nachum, S. Gu, H. Lee, and S. Levine. Near-optimal representation learning for hierarchical reinforcement learning, 2018.

[48] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[49] T. Osa, V. Tangkaratt, and M. Sugiyama. Hierarchical reinforcement learning via advantage-weighted information maximization, 2019.

[50] A. Pashevich, D. Hafner, J. Davidson, R. Sukthankar, and C. Schmid. Modulated policy hierarchies, 2018.

[51] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies, 2019.

[52] D. Precup. Temporal abstraction in reinforcement learning, 2000.

[53] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML 2018: Proceedings of the Thirty-Fifth International Conference on Machine Learning*, 2018.

[54] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, 2016.

[55] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.

[56] B. Sallans and G. E. Hinton. Reinforcement learning with factored states and actions. *The Journal of Machine Learning Research*, 2004.

[57] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge, 2019.

[58] N. N. Schraudolph and D. Kamenetsky. Efficient exact inference in planar ising models. In *Advances in Neural Information Processing Systems 21*. 2009.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[60] Y. Song, J. Wang, T. Lukasiewicz, Z. Xu, and M. Xu. Diversity-driven extensible hierarchical reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 2019.

[61] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning, 2020.

[62] S. Sukhbaatar, E. Denton, A. Szlam, and R. Fergus. Learning goal embeddings via self-play for hierarchical reinforcement learning, 2018.

[63] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2085–2087, 2018.

[64] K. Suri, X. Shi, K. Plataniotis, and Y. Lawryshyn. Evolve to control: Evolution-based soft actor-critic for scalable reinforcement learning. *arXiv preprint*, 2020.

[65] K. Suri, X. Q. Shi, K. Plataniotis, and Y. Lawryshyn. Energy-based surprise minimization for multi-agent value factorization. *arXiv preprint arXiv:2009.09842*, 2020.

[66] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2018.

[67] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[68] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. Deepmind control suite, 2018.

[69] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *The Journal of Machine Learning Research*, 2003.

[70] C. J. Thompson. *Mathematical Statistical Mechanics*. Princeton University Press, 1972.

[71] D. Tirumala, H. Noh, A. Galashov, L. Hasenclever, A. Ahuja, G. Wayne, R. Pascanu, Y. W. Teh, and N. Heess. Exploiting hierarchy for learning and transfer in kl-regularized rl, 2019.

[72] G. Tkacik, E. Schneidman, M. J. B. II, and W. Bialek. Ising models for networks of real neurons, 2006.

[73] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. IROS, 2012.

[74] W. Wang, Y. Hu, and S. Scherer. Learning temporal abstraction with information-theoretic constraints for hierarchical reinforcement learning, 2020.

[75] G. H. Wannier. Statistical physics. 1967.

[76] T. T. Wu, B. M. McCoy, C. A. Tracy, and E. Barouch. Spin spin correlation functions for the two-dimensional Ising model: Exact theory in the scaling region. *Phys. Rev. B*, 1976.

[77] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, 2017.

[78] M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. Tobias Springenberg, M. Neunert, N. Siegel, T. Hertweck, T. Lampe, N. Heess, and M. Riedmiller. Compositional transfer in hierarchical reinforcement learning. *Robotics: Science and Systems XVI*, 2020.

[79] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

[80] N. Yoshioka, Y. Akagi, and H. Katsura. Transforming generalized ising models into boltzmann machines. *Physical Review E*, 2019.

[81] S. Zhang and S. Whiteson. Dac: The double actor-critic architecture for learning options, 2019.

[82] X. Zuo. mazelab: A customizable framework to create maze and gridworld environments., 2018.