Diverse Policy Optimization for Structured Action Space

Wenhao Li
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
liwenhao@cuhk.edu.cn

Baoxiang Wang
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
bxiangwang@cuhk.edu.cn

Shanchao Yang
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
shanchaoyang@link.cuhk.edu.cn

Hongyuan Zha*
The Chinese University of Hong Kong, Shenzhen,
Shenzhen Institute of AI and Robotics for Society
Shenzhen, China
zhahy@cuhk.edu.cn

ABSTRACT

Enhancing the diversity of policies is beneficial for robustness, exploration, and transfer in reinforcement learning (RL). In this paper, we aim to seek diverse policies in an under-explored setting, namely RL tasks with *structured action spaces* with the two properties of *composability* and *local dependencies.* The complex action structure, non-uniform reward landscape, and subtle hyperparameter tuning due to the properties of structured actions prevent existing approaches from scaling well. We propose a simple and effective RL method, *Diverse Policy Optimization (DPO)*, to model the policies in structured action space as the energy-based models (EBM) by following the probabilistic RL framework. A recently proposed novel and powerful generative model, GFlowNet, is introduced as the efficient, diverse EBM-based policy sampler. DPO follows a joint optimization framework: the outer layer uses the diverse policies sampled by the GFlowNet to update the EBM-based policies, which supports the GFlowNet training in the inner layer. Experiments on ATSC and Battle benchmarks demonstrate that DPO can efficiently discover surprisingly diverse policies in challenging scenarios and substantially outperform existing state-of-the-art methods.

KEYWORDS

Reinforcement Learning; Generative Model; Diversity; Robustness

ACM Reference Format:
Wenhao Li, Baoxiang Wang, Shanchao Yang, and Hongyuan Zha. 2023. Diverse Policy Optimization for Structured Action Space. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 10 pages.

1 INTRODUCTION

The history of human civilization can be seen as a chronicle of creative capacity, i.e., the diversity of solutions to the same puzzle [32]. Counter-intuitively, a popular consensus in deep learning with theoretical justifications [27] that most local optimas to a non-convex optimization problem are very close to the global optimum has led mainstream AI research to focus on finding a single local solution to a given optimization problem, rather than on which local optimum is dicovered [59]. It is no coincidence that most methods in reinforcement learning (RL) are also designed to seek a single reward-maximizing policy [30, 39, 43].

However, different local optima in the policy space can correspond to strategies that differ in nature, which makes the above consensus problematic in RL tasks where the environment is unstable. For example, in adaptive traffic signal control (ATSC) [45, 48, 49] (conceptual diagram and more examples are included in Figure 1), if two traffic flows are desired to reach the target points from the departure points quickly, multiple control strategies with similar average commuting times may exist due to the combinatorial nature of traffic lights. The performance of a single policy obtained by reward maximization is bound to be affected if the subsequent traffic volumes on other sections of the road network associated with the traveled section of that traffic change. Moreover, if our goal is to discover a diverse set of policies, some of these may prove more valuable than others in different situations.

Therefore, celebrating the diversity of policies is beneficial for many RL applications. In addition to ATSC and the simple game in Figure 1, these RL application areas include but are not limited to conversation generation in intelligent customer service [23], drug discovery in smart healthcare [36], and simulator design in automated machine learning (AutoML) [47]. Furthermore, in addition to robustness, a set of diverse policies can also be useful for exploration [35], transfer [21], and hierarchy [1] in RL.

There is no doubt that RL researchers have demonstrated their creative ability in discovering diverse policies. The majority of the literature has been done in the field of neuroevolution methods inspired by Quality-Diversity (QD), which typically maintains a collection of policies and adapts it using evolutionary algorithms to balance the QD trade-off [7, 11, 25, 31, 34, 38]. In another part of the work, intrinsic rewards have been used for learning diversity in terms of the discriminability of different trajectory-specific quantities [1, 8, 12, 13, 16, 40, 53], or have been used as a regularizer when maximizing the extrinsic reward [10, 22, 29, 41, 56]. There is also a small body of work that transforms the problem into a Constrained Markov Decision Process (CMDP) [6, 42, 54, 59], or implicitly induce diversity to learn policies that maximize the set robustness to the worst-possible reward [21, 52].

This paper considers a more complex, realistic, less focused, and under-explored setting, namely RL tasks with *structured action spaces.* We define structured actions as actions with the following

[*] Corresponding author

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

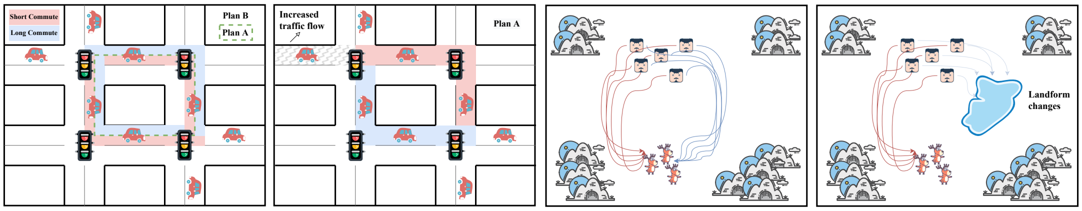

Figure 1: Robustness of diverse policies in two non-stationary environments: (Left) the adaptive traffic signal control and (Right) the predator-prey. In these tasks, diverse policies can quickly adapt to changes in the external environment.

two properties: *composability*, i.e., environmental actions consist of a large number of atomic actions with complete functionality and *local dependencies*, i.e., there are local physical or logical correlations between atomic actions[1]. For example, in ATSC, the phases of all traffic signals on all intersections in the entire road network must be redetermined at certain intervals, and atomic actions are phases of each signal and interact with each other through the physical road network. In addition, for the predator-prey task in Figure 1, the atomic actions are the decisions of each predator, and there is a local spatial, logical association.

The high dimensionality of the RL agent's policy due to the composability of structured actions prevents existing methods from scaling well. Specifically, the combinability will make the underlying reward landscape of the RL problem particularly non-uniform, which may make QD-like methods require substantially large population sizes to fully explore the policy space and prevent the algorithm from collapsing to visually identical policies [44, 59]. Also, due to composability, the complex soft objective introduced by intrinsic reward or CMDP-driven methods will result in non-trivial and subtle hyperparameter tuning [29, 34]. In addition, the existing agents' policies are mainly parameterized categorical distributions or Gaussian distributions. Their extension to structured actions with independent assumptions on atomic actions will prevent the agent from effectively using the structural information of environmental actions to achieve an efficient search for the policy space.

We propose a simple and effective RL method, *Diverse Policy Optimization (DPO)*, to discover a diverse set of policies in tasks with structured action spaces. We follow the probabilistic reinforcement learning (PRL) framework [22] to transform reinforcement learning problems under stochastic dynamics into variational inference problems on probabilistic graphical models and model the policies of RL agents as the energy-based models (EBM). The action distribution induced by this EBM in a structured action space is highly multimodal, and sampling from such a high-dimensional distribution is intractable. To this end, we introduce a recently proposed novel and powerful generative model, *Generative Flow Networks (GFlowNet)* [3, 4, 17, 55], as the efficient diverse policy sampler. GFlowNet can be regarded as amortized Monte-Carlo Markov chains (MCMC), which gradually builds composable environmental actions through the single but trained generative pass of "building blocks (i.e., atomic actions)", so that the final sampled environmental actions obey a given energy-based policy distribution.

Notably, our method does not simply introduce the GFlowNet to RL with structured action spaces. Since in the PRL framework, with the update of the soft Q function, the energy-based policy distribution is also constantly changing. This violates the assumption of the fixed energy model in GFlowNet and makes DPO face a more complex optimization problem. Therefore, we model DPO as a joint optimization problem: the outer layer uses the diverse policies sampled by the GFlowNet to update the soft Q function, and the inner layer trains the GFlowNet through an EBM based on the soft Q function (see Figure 3). Furthermore, a two-timescale alternating optimization method is proposed to solve it efficiently.

We empirically validate DPO on ATSC tasks [2] where atomic actions have local physical dependencies, and more generally, Battle scenarios [58] where atomic actions have logical local dependencies. Experiments demonstrate that DPO can reliably and efficiently discover surprisingly diverse strategies in all these challenging scenarios and substantially outperform existing baselines. The contributions can be summarized as follows:

(1) We propose a novel algorithm, *Diverse Policy Optimization*, for discovering diverse policies for structured action spaces. The GFlowNet-based sampler can efficiently sample diverse policies from the high-dimensional multimodal distribution induced by structured action spaces.
(2) We propose an efficient joint training framework to interleaved optimize the soft-Q-function-based EBM and the reward-conditional GFlowNet-based sampler.
(3) Our algorithm is general and effective across structured action spaces with physical and logical local dependencies.

2 PRELIMINARIES AND NOTATIONS

The proposed DPO follows the PRL to model policies as a high-dimensional multimodal energy-based probability distribution and introduces GFlowNet to efficiently sample policies with diversity from this distribution. Below, we briefly review the PRL and GFlowNet.

2.1 Probabilistic Reinforcement Learning

PRL aims to learn the maximum entropy optimal policy:

$$\pi^*_{\text{ent}} = \arg\max_{\pi} \sum_{t} \mathbb{E}_{(s_e^t, a_e^t)\sim\rho_\pi}\left[r\left(s_e^t, a_e^t\right) + \alpha\mathcal{H}\left(\pi\left(\cdot \mid s_e^t\right)\right)\right],$$

where $s_e^t \in \mathcal{S}_e$ and $a_e^t \in \mathcal{A}_e$ denotes the state and action respectively. The subscript e represents the "environment", which is used to distinguish related concepts in RL from GFlowNets, and the α is the coefficient to trade off between entropy and reward. Function $\mathcal{H}$ denotes the entropy term. By defining the soft Q function as:

$$Q^*_{\text{soft}}(s_e^t, a_e^t) := r_e^t + \mathbb{E}_{s_e^{t+\ell}\sim\rho_\pi}\left[\sum_{\ell=1}^{\infty}\gamma^\ell\left(r_e^{t+\ell} + \alpha\mathcal{H}\left(\pi^*_{\text{ent}}\left(\cdot \mid s_e^{t+\ell}\right)\right)\right)\right]. \quad (1)$$

[1] In this paper, only pairwise relationships between atomic actions are considered.

The optimal maximum entropy policy can be proved as in [22]

$$\pi^*_{\text{ent}} = \exp\left(\frac{1}{\alpha}\left(Q^*_{\text{soft}}\left(s^t_e, a^t_e\right) - V^*_{\text{soft}}\left(s^t_e\right)\right)\right), \tag{2}$$

where the soft value function V^*_{soft} is defined by

$$V^*_{\text{soft}}\left(s^t_e\right) = \alpha \log \int_{\mathcal{A}_e} \exp\left(\frac{1}{\alpha} Q^*_{\text{soft}}\left(s^t_e, a'_e\right)\right) da'_e. \tag{3}$$

Thus the policy learning can be treated as the approximation to the Boltzmann-like distribution of optimal Q function. Taking the soft Q-Learning (SQL) [14] method as an example, it provides the optimal Q is the fixed point of soft Bellman backup, which satisfies the soft Bellman equation

$$Q^*_{\text{soft}}\left(s^t_e, a^t_e\right) = r^t_e + \gamma \mathbb{E}_{s^{t+1}_e \sim p_{s_e}}\left[V^*_{\text{soft}}\left(s^{t+1}_e\right)\right]. \tag{4}$$

Due to the infinite set of states and actions, it takes parameterized Q and uses a function π as an approximate sampler of Boltzmann-like distribution of Q. Specifically, it updates Q and π as:

$$\begin{cases} \min_\theta J_Q(\theta) := \mathbb{E}_{s^t_e, a^t_e, r^t_e, s^{t+1}_e \sim D}\left[\frac{1}{2}\left(r^t_e + V^{\bar{\theta}}\left(s^{t+1}_e\right) - Q^\theta\left(s^t_e, a^t_e\right)\right)^2\right], \\ \min_\phi J_\pi\left(\phi; s^t_e\right) := \text{KL}\left(\pi^\phi\left(\cdot|s^t_e\right) \| \exp\left(\frac{1}{\alpha}\left(Q^\theta\left(s^t_e, \cdot\right) - V^\theta(s^t_e)\right)\right)\right), \end{cases} \tag{5}$$

where function V^θ is denoted as

$$V^\theta\left(s^t_e\right) := \alpha \log \mathbb{E}_{a'_e \sim q_{a'_e}}\left[\exp\left(\frac{1}{\alpha} Q^\theta\left(s^t_e, a'_e\right)\right) / q_{a'_e}\left(a'_e\right)\right], \tag{6}$$

and $\theta, \bar{\theta}, \phi$ denote the parameters of critic, target critic and policy respectively; $q_{a'}$ is an arbitrary policy distribution. The policy distribution induced by the EBM (i.e., the Boltzmann-like distribution of Q) under structured action spaces is highly multimodal, and sampling from such a high-dimensional distribution is intractable. In this paper, DPO introduces a powerful generative model, *Generative Flow Networks (GFlowNet)*, as the efficient diverse policies sampler.

2.2 Generative Flow Networks

Generative flow networks, which are trainable generative policies, model the generation or sampling process of composite objects $x \in \mathcal{X}$ by a sequence of discrete *actions* that incrementally modify a partially constructed object (*state*). Note that action and state here do not refer to the concepts in RL [55]. In this paper, we model *actions in RL* problems with structured action spaces as *states in GFlowNet*, and *actions in GFlowNet* correspond to the *atomic actions* that compose structured actions. In other words, the composite object x generated by the GFlowNet is a_e, and $\mathcal{X}$ is equivalent to $\mathcal{A}_e$. The partially constructed object and corresponding action sequence space can be represented by a directed acyclic graph (DAG, See the DAG consisting of traffic lights and roads in Figure 2) $G = (\mathcal{S}_g, \mathcal{A}_g)$, where the subscript g denotes the "GFlowNet". The vertices in $\mathcal{S}_g$ are states and the edges in $\mathcal{A}_g$ are actions that modify one state to another. The tails of incoming edges and the heads of outgoing edges of a state are denoted as the *parents* and *childrens*, respectively. The sampling process of the composite object a_e starts from the *initial state* s^0_g and transits to the *terminal* state $s^n_g \in \mathcal{A}_e$, which is a state without outgoing edges, after $n \in (0, T]$ steps and T is the maximum length. Note that the same terminal state may correspond to multiple action sequences.

A *complete trajectory* is a state sequence from a initial state to a terimal state $s^0_g \rightarrow s^1_g \rightarrow \ldots \rightarrow s^n_g$, where each transition $s^t_g \rightarrow s^{t+1}_g$ is an action in $\mathcal{A}_g$. A *trajectory flow* is a unnormalized density or a non-negative function, $F : \mathcal{T} \rightarrow \mathbb{R}_{\geq 0}$, on the set of all complete trajectories $\mathcal{T}$. The flow is called *Markovian* if there exist distributions $P_F(\cdot \mid s_g)$ over the children of every non-terminal state s_g and a constant Z, such that for any complete trajectory τ we have $P_F(\tau) = F(\tau)/Z$ with $P_F(\tau) = P_F\left(s^1_g \mid s^0_g\right) P_F\left(s^2_g \mid s^1_g\right) \ldots P_F\left(s^n_g \mid s^{n-1}_g\right)$. $P_F\left(s^{t+1}_g \mid s^t_g\right)$ is called a *forward policy*, which is used to sample the composite object a_e from the density F. $P_T(a_e)$ then denotes the probability that a complete trajectory sampled from P_F terminates in a_e.

The problem we are interested in is fitting a Markovian flow to a fixed *energy function* on $\mathcal{A}_e$. Given an energy function $\mathcal{E}(a_e) := -\log R(a_e)$ and the associated non-negative *reward function* (again, not a reward in RL) $R_g : \mathcal{A}_e \rightarrow \mathbb{R}_{\geq 0}$, one seeks a Markovian flow F such that the likelihood of a complete trajectory sampled from F terminating in a given a_e is proportional to $R_g(a_e)$, i.e., $P_T(a_e) \propto R_g(a_e)$. This F can be obtained by imposing the *reward-matching* constraint: $R_g(a_e) = \sum_{\tau=\left(s^0_g \rightarrow \ldots \rightarrow s^n_g\right), s^n_g = a_e} F(\tau)$. The details of how to parameterize a GFlowNet and train a Markovian flow F that satisfies the reward matching constraint will be explained soon.

3 DIVERSE POLICY OPTIMIZATION

This section proposes a simple and effective RL method, *Diverse Policy Optimization (DPO)*, to discover diverse policies in structured action spaces. We follow the probabilistic reinforcement learning (PRL) framework [22] to transform RL problems under stochastic dynamics into variational inference problems on probabilistic graphical models and model the policies of RL agents as EBMs. PRL framework corresponds to a maximum entropy variant of reinforcement learning or optimal control, where the optimal policy aims to maximize the expected reward and maintain high entropy. Due to the maximum entropy objective, some existing works [14, 15] have proposed algorithms for low-dimensional continuous action spaces to discover diverse policies based on this framework.

Our method is an instance of the maximum entropy actor-critic algorithm in the PRL framework, which adopts a message-passing approach and can produce lower-variance estimates. In addition, to make the policy still scalable in the structured action space, we do not use an explicit policy parameterization but fit only the message, i.e., the Q-value function, similar to soft Q-learning [14]. Specifically, we opt for using general energy-based policies $\pi\left(a_e \mid s_e\right) \propto \exp\left(-\mathcal{E}\left(s_e, a_e\right)\right)$, where $\mathcal{E}$ is an energy function. Furthermore, we set $\mathcal{E}\left(s_e, a_e\right) = -\frac{1}{\alpha} Q_{\text{soft}}\left(s_e, a_e\right)$, then the optimal maximum entropy policy is an EBM that satisfies Equation (2).

However, The action distribution induced by this EBM in a structured action space is highly multimodal, and sampling from such a high-dimensional distribution is intractable. Fortunately, the composability and local dependencies of the structured action space make generative flow networks naturally suitable for efficiently sampling diverse and high-quality policies from it. And we only need to set the energy function that needs to be fitted by the Markovian flow $F(a_e)$ (where the action a_e corresponding to the composite

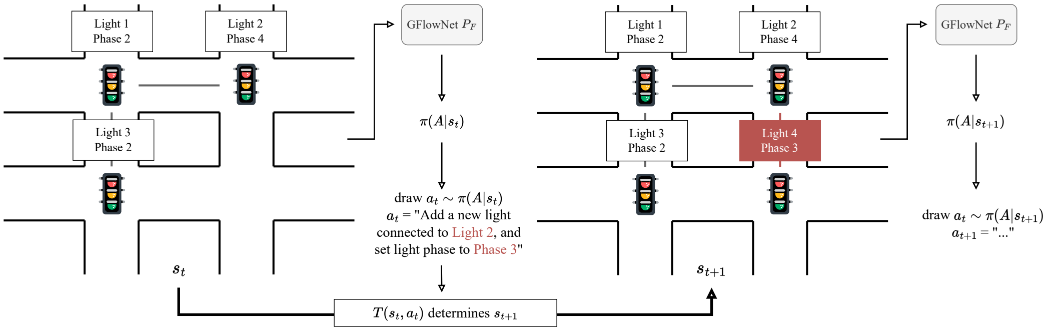

Figure 2: A GFlowNet iteratively constructs an composite object, e.g., a traffic light network. s_t represents the state of the partially constructed object, a_t represents the action taken by the GFLowNet to transition to state $s_{t+1} = T(s_t, a_t)$. The GFlowNet take a 3-lights traffic network as input and determines an action to take. This process repeats until an exit action is sampled or maximum light number is achieved and the sample is complete.

object x) to be $(-1/\alpha) \cdot Q_{\text{soft}}\ (s_e, a_e)$, and its associated reward function $R_g(a_e)$ to be set to $\exp\left((1/\alpha) \cdot Q_{\text{soft}}\ (s_e, a_e)\right)$, we can elegantly introduce GFlowNet as an efficient and diverse sampler.

Nevertheless, the unreasonable part of the above modeling is that there is no place left for the environment state s_e in the input of the Markovian flow and the reward function. The reason is that π in the PRL framework is a *conditional* distribution, but GFlowNet is an *unconditional* sampler. To this end, we will introduce a variant of GFlowNet, namely *reward-conditional GFlowNet*, to model the policy of RL agents, and details will be explained shortly.

Since in the PRL framework, with the update of the Q_{soft}, the energy-based policy distribution is also constantly changing. DPO adopts a joint training framework where the EBM and the GFlowNet are optimized alternately, similar with [55]: The energy function serves as the negative log-reward function for the GFlowNet, which is trained with the trajectory balance [28] objective to sample from the evolving energy-based policies. In contrast, the energy function is trained with soft Bellman backup, where the GFlowNet provides diverse samples. The schematic diagram of RL based on reward-conditional GFlowNet as the agent's policy and the joint training framework are shown in Figure 3 and Algorithm 1.

In the following, we will explain the generation process of structured action, the parameterization and training of reward-conditional GFlowNet, and its interleaved update with EBM, respectively.

3.1 Structured Action Generative Process

The framework of diverse policy optimization is introduced in the previous section, and this section will describe the process of generating structured actions based on the reward-conditional GFlowNet. The local dependencies of structured actions indicate that there may be two correlations between atomic actions: locally physical and locally logical correlations. The former is a typical graph, while the latter belongs to a typical set. For the unity of the framework, this paper only considers the physical correlation between atomic actions. It transforms the logical correlation into the physical correlation without loss of generality.

Expressly, we assume that atomic actions with local logical correlations have a fixed influence range with a radius d in Euclidean space. An atomic action can then establish a physical correlation with others within its influence range. Of course, other types of topologies, such as fully connected, star, hierarchical, etc., can also be used in addition to adjacency topologies. This paper adopts the adjacency topology to make a trade-off between efficiency and performance. The experimental results also show that the algorithm performance is not sensitive to the influence radius d.

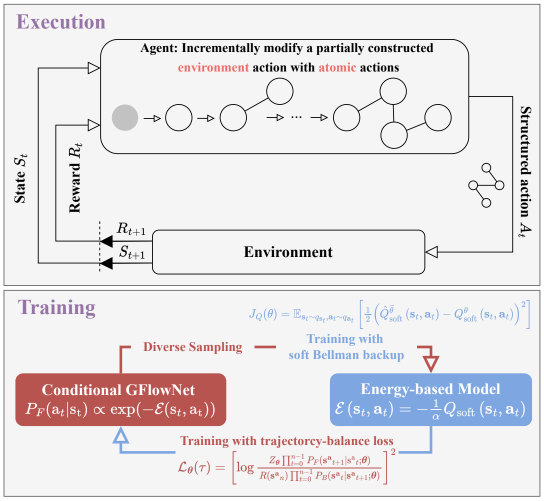
Execution
Agent: Incrementally modify a partially constructed environment action with atomic actions
State S_t
Reward R_t
Structured action A_t
R_{t+1}
S_{t+1}
Environment
Training
$J_Q(\theta) = \mathbb{E}_{\mathbf{s}_t \sim q_{\mathbf{s}_t}, \mathbf{a}_t \sim q_{\mathbf{a}_t}} \left[\frac{1}{2} \left(\hat{Q}^{\bar{\theta}}_{\text{soft}} (\mathbf{s}_t, \mathbf{a}_t) - Q^{\theta}_{\text{soft}} (\mathbf{s}_t, \mathbf{a}_t) \right)^2 \right]$
Diverse Sampling
Training with soft Bellman backup
Conditional GFlowNet
$P_F(\mathrm{a}_t|\mathrm{s}_t) \propto \exp(-\mathcal{E}(\mathrm{s}_t, \mathrm{a}_t))$
Energy-based Model
$\mathcal{E}(\mathbf{s}_t, \mathbf{a}_t) = -\frac{1}{\alpha} Q_{\text{soft}} (\mathbf{s}_t, \mathbf{a}_t)$
Training with trajectorcy-balance loss
$\mathcal{L}_{\theta}(\tau) = \left[\log \frac{Z_{\theta} \prod_{t=0}^{n-1} P_F(\mathbf{s^a}_{t+1}|\mathbf{s^a}_t;\theta)}{R(\mathbf{s^a}_n) \prod_{t=0}^{n-1} P_B(\mathbf{s^a}_t|\mathbf{s^a}_{t+1};\theta)} \right]^2$

Figure 3: The schematic diagram of RL based on GFlowNet as the agent's policy and the joint training framework.

In the structured action space, the action consists of N atomic actions in K-dimensional discrete space, i.e., $a_e \in \mathcal{A}_e \triangleq [K]^N$, where $[K] \triangleq \{0, \ldots, K-1\}$. a_e could be a phase configuration of N traffic lights, and each traffic light contains K phases or the joint action of N predators, and each predator can go in K directions. We model the generation or sampling of vectors in $\mathcal{A}_e$ by a reward-conditional GFlowNet. The state space of GFlowNet is denoted as $\mathcal{S}_g$, and we have $\mathcal{S}_g \triangleq \left\{ \left(s_g^1, \ldots, s_g^N\right) \mid s_g^n \in [K] \cup \oslash, n = 1, \ldots, N \right\}$, where the void symbol $\oslash$ represents a yet unspecified atomic action. The DAG structure on $\mathcal{S}_g$ is the N-th Cartesian power of the DAG with states $[K] \cup \oslash$, where $[K]$ are children of $\oslash$. Concretely, the children of a state $s_g = \left(s_g^1, \ldots, s_g^N\right)$ are vectors that can be obtained from s_g by changing any one atomic action $\mathbf{s}_g^n$ from $\oslash$ to $[K]$, and its parents are states that can be obtained by changing a single atomic action $s_g^n \in [K]$ to $\oslash$.

Moreover, $\mathcal{A}_e$ is naturally identified with $\{s_g \in \mathcal{S}_g : |s_g| = D\}$ where $|s_g| \triangleq \#\left\{s_g^n \mid s_g^n \in [K], n = 1, \ldots, N\right\}$. Similarly, the initial state is denoted as $s_g^0 \triangleq (\oslash, \oslash, \ldots, \oslash)$, which means that the reward-conditional GFlowNet-based RL policy needs to take N steps to sample a structured action, i.e., constructing a trajectory from s_g^0 to $a_e \in \mathcal{A}_e$. The forward policy $P_{F|e}(\cdot|s_g, s_e)$ of a reward-conditional GFlowNet (will explained soon), extends from §2.2, is a distribution over all paths to select a position with a void atomic action in s_g and a value $k \in [K]$ to assign to this atomic action based on the environmental state $s_e \in \mathcal{S}_e$. Thus the action space for a state s_g has size $K(N - |s_g|)$. Since $k \ll N$, the action space of the forward policy (same as the backward policy below) grows *linearly* with the atomic actions increase, so DPO has a good scalability. Correspondingly, the backward policy $P_{B|e}(\cdot|s_g, s_e)$ is a distribution over the $|s_g|$ paths to select a position with a nonvoid atomic action in s_g.

More efficient generation. As we mentioned earlier, as an amortized version of MCMC, GFlowNets can alleviate the mix-moding problem [18, 37] of the MCMC method, thereby improving the sampling efficiency of diverse samples. However, if the two modes are close enough, the MCMC method will have higher sampling efficiency because it only perturbs the previous sample slightly. However, GFlowNets, for this case, need to rebuild the entire structured action sequentially, although only a minimal number of atomic actions have changed. To this end, we introduce a small trick: adding a *termination* action in the action space. GFlowNets are trained to successfully sample from two close modes by deciding to terminate at different modes at different runs. Since the physical meaning of the termination action is quite different from other actions, we use a different output head to predict it separately, as shown in Figure 4. Once the forward policy $P_{F|e}(\cdot|s_g, s_e)$ decides to take the termination action, the output of the other head will be ignored. Experiments show that this small trick can significantly improve the learning efficiency of the algorithm in some tasks.

3.2 GFlowNet Parameterization

After showing how to sample structured actions using the GFlowNet, this section elaborates on how to parameterize it and train a Markovian flow F that satisfies the reward matching constraint. As stated earlier, if we take the form of the GFlowNet in §2.2, there will be no place for the environment state s_e in the forward policy P_F as well as in the backward policy P_B. Thus, we use an extended version of flow networks by conditioning each component on some information, which is *external* to the flow network but influences the terminating flows. In our setting, the external information is RL's environmental state s_e. Since the external information s_e affects the reward function R_g in §2.2, this conditional GFlowNet is also called *reward-conditional GFlowNet* [4, Definition 29].

Since reward-conditional GFlowNets are defined using the same components as the unconditional one, they inherit from all the properties of the GFlowNet for all DAGs $G_e = (\mathcal{S}_g, \mathcal{A}_g, \mathcal{S}_e)$ and flow functions $F_e : \mathcal{T} \times \mathcal{S}_e \to \mathbb{R}_{\geq 0}$, where e represents the "environment" in RL again. In particular, we can directly extend notions of §2.2 to reward-conditional GFlowNets with forward policy $P_{F|e}(\cdot|s_g, s_e)$, backward policy $P_{B|e}(\cdot|s_g, s_e)$, energy function $\mathcal{E}(a_e|e) := -\log R_(g|e)(a_e|s_e)$ and the associated non-negative reward function $R_{g|e} : \mathcal{A}_e \times \mathcal{S}_e \to \mathbb{R}_{\geq 0}$; The only difference is that now every term explicitly depends of the conditioning variable, environmental state $s_e \in \mathcal{S}_e$ under the RL context.

In our experiments, we parameterize the forward and backward policy with deep neural networks $P_{F|e}(\theta_F)$ and $P_{B|e}(\theta_B)$ respectively, and for convenience, we omit the input without introducing ambiguity. As P_F incrementally builds structured actions, its action space gradually decreases, similar to the traveling salesman problem (TSP) [33]. Considering the effectiveness of the pointer network [46] in dealing with such problems, we introduce the modified graph pointer network (GPN, [26]) as the forward and backward policy (see Figure 2) to further model the structured information of the action space. The forward process of the modified GPN can be divided into the following three stages:

Environmental state encoding: In this stage, the i-th row of the adjency matrix ℓ_i and local observed information o_i of each atomic action are concatenated as $s_{i|e} = [\ell_i \| o_i]$, and then $s_{i|e}$ is embedded into a higher dimensional vector $\tilde{s}_{i|e} \in \mathbb{R}^d$ by a shared feed-forward network, where d is the hidden dimension. The context information is then obtained by encoding all atomic actions' embeddings s_e via a graph neural network (GNN, [20, 50]), where $s_e = [\tilde{s}_{1|e}^\top, \ldots, \tilde{s}_{N|e}^\top]^\top$. Each layer of the GNN is expressed as:

$$s_{i|e}^\ell = \gamma s_{i|e}^{\ell-1}\Theta + (1-\gamma)\phi_\theta\left(\frac{1}{|\mathcal{N}(i)|}\left\{s_{j|e}^{\ell-1}\right\}_{j\in\mathcal{N}(i)\cup\{i\}}\right), \tag{7}$$

where $s_{i|e}^\ell \in \mathbb{R}^{d_\ell}$ is the ℓ-th layer variable with $\ell \in \{1, \ldots, L\}$, $s_{i|e}^0 = s_{i|e}$, γ is a trainable parameter, $\Theta \in \mathbb{R}^{d_{\ell-1}\times d_\ell}$ is a trainable weight matrix, $\mathcal{N}(i)$ is the adjacency set of atomic action i, and $\phi_\theta : \mathbb{R}^{d_{\ell-1}} \longrightarrow \mathbb{R}^{d_\ell}$ is the aggregation function [20], which is represented by a neural network in our experiments.

GFlowNet state encoding: In this stage, we use the vectors pointing from the newly added atomic action to all others as the embedding of s_g, which is similar with Ma et al. [26]. Specifically, for the newly added atomic action $\tilde{s}_{i|e}$, suppose $s_{\mathrm{E|i}} = \left[\tilde{s}_{i|e}^\top, \ldots, \tilde{s}_{i|e}^\top\right]^\top \in \mathbb{R}^{N\times d}$ is a matrix with identical rows $\tilde{s}_{1|e}$. We define $s_g = s_{i|e}^L - s_{\mathrm{E|i}} = \left[s_{i|g}^\top, \ldots, s_{N|g}^\top\right]^\top \in \mathbb{R}^{N\times d}$. Then s_g is passed into the GNN again and the embedding of each atomic action after GFlowNet state encoding is denoted as $s_{i|g}^L$.

Atomic action selection: The atomic action selector is based on the Linear Transformer [19], which has the advantage of not suffering from the quadratic scaling in the input size. This architecture relies on a linearized attention mechanism, defined as

$$Q = s_g^L W_Q \quad K = s_g^L W_K \quad V = s_g^L W_V,$$

$$\mathrm{LinAttn}_k(s_g^L) = \frac{\sum_{j=1}^N \left(\psi(Q_k)^\top \psi(K_j)\right) V_j}{\sum_{j=1}^N \psi(Q_k)^\top \psi(K_j)}, \tag{8}$$

where $\psi(\cdot)$ is a non-linear feature map, and Q, K, and V are linear transformations of s_g^L corresponding to the queries, keys, and

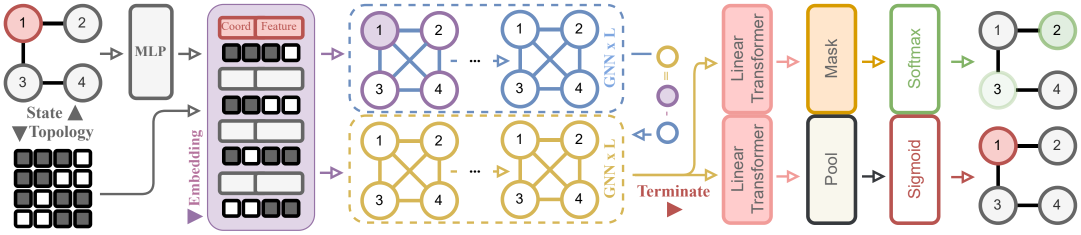

Figure 4: The parameterized forward and backward policy based on the modified graph pointer network.

values respectively, as is standard with Transformers. The pointer vector outputted by the Linear Transformer is first masked by the mask $\mathbf{m}$ associated with the physical dependencies in structured action space and is then passed to a softmax layer to generate a distribution over the next candidate intersections. Similar to pointer networks [46], the masked pointer vector $\mathbf{u}_i$ is defined as:

$$\mathbf{u}_i^{(j)} = \begin{cases} \mathbf{u}_i^{(j)} & \text{if } j \neq \sigma(k), \forall k < j, \\ -\infty & \text{otherwise,} \end{cases} \tag{9}$$

where $\sigma(k)$ denotes k-th processed atomic action and $\mathbf{u}_i^{(j)}$ is the j-th entry of the vector $\mathbf{u}_i$.

3.3 Reward-Conditional GFlowNet Training

After parameterizing the GFlowNet, we now describe how reward-conditional GFlowNets could be trained toward matching a given conditional reward. Recall from §2.2, §3 and §3.1, given a non-negtive conditional reward function $R_{g|e} : \mathcal{A}_e \times \mathcal{S}_e \to \mathbb{R}_{\geq 0}$, a reward-conditional GFlowNet can be trained so that its terminating probability distribution matches the associated energy-based model. To be precise, the marginal likelihood that a trajectory sampled from the forward policy $P_{F|e}(\cdot|s_g, s_e)$ terminates at a given structured action is propotional to the action's soft Q value $P_T(a_e|s_e) \propto \exp\left((1/\alpha) \cdot Q_{\text{soft}}\ (s_e, a_e)\right)$, where $a_e \in \mathcal{A}_e$ and $s_e \in \mathcal{S}_e$.

To train the parameters θ_F and θ_B of the reward-conditional GFlowNet, we use the trajectory balance objective [28] that optimizes the following objective along complete trajectories $\tau = (s_g^0 \to s_g^1 \to \ldots \to \ldots \to s_g^n)$:

$$\mathcal{L}_\Theta(\tau|s_e) = \left[\log \frac{Z\left(s_e; \theta_Z\right) \prod_{t=0}^{n-1} P_F\left(s_g^{t+1}|s_g^t, s_e; \theta_F\right)}{R\left(s_g^n|s_e\right) \prod_{t=0}^{n-1} P_B\left(s_g^t|s_g^{t+1}, s_e; \theta_B\right)}\right]^2, \tag{10}$$

where $\Theta \triangleq \{\theta_F, \theta_B, \theta_Z\}$. The scalar function $Z(\cdot)$ is parametrized in the log domain, as suggested by Malkin et al. [28]. With the trajectory balance objective, we train the reward-conditional GFlowNet with stochastic gradient $\mathbb{E}_{\tau \sim \pi_\Theta(\tau|s_e)}\left[\nabla_\Theta \mathcal{L}_\Theta(\tau|s_e)\right]$ with some training trajectory distribution $\pi_\Theta(\tau)$. Akin to RL settings, we take π_Θ to be the distribution over trajectories sampled from a tempered version of current forward policy $P_{F|e}(\cdot|s_g, s_e)$. That is, τ is sampled with $\mathbf{s}_g^{t+1} \sim P_{F|e}(\cdot|s_g^t, s_e)$ starting from s_e^0, mixed with a uniform action policy to ensure π_Θ has full support.

Learning about total flow Z**.** Experiments show that learning the scalar function $Z(\cdot)$ end-to-end is very difficult. Since Z represents the total flow in the entire flow network, many samples are required for an accurate estimation. Unlike the original work of trajectory balance [28], in our setting, the scalar function Z needs to condition on the external environmental state s_e thus has higher sample complexity. Interestingly, since the target EBM of GFlowNets is derived from the PRL framework in our method, Z has an additional physical meaning, i.e., the soft value function $V_{\text{soft}}^*(\cdot)$ in §2.1. Since the soft value function is dependent on the soft Q value, Z can be updated by a mechanism similar to the bootstrap learning adopted by RL, thereby improving the sample efficiency. To this end, in addition to end-to-end training of Z using Equation (10), we estimate $V_{\text{soft}}^*(\cdot)$ in the same way as in Haarnoja et al. [14] and fit Z to it. The experimental results show that this form of mixed gradient update can improve the learning efficiency of Z.

3.4 Joint Training with EBM

Reward-conditional GFlowNets' training relies on a given function $R_{g|e}(a_e|s_g, s_e)$ to provide reward signals. However, in the PRL framework, the energy-based policy distribution is also constantly changing with the update of the soft Q function. Therefore, we propose a joint training framework (Algorithm 1), where the EBM and the reward-conditional GFlowNet are optimized alternately:

(1) **GFlowNet updating step:** the soft Q function serves as the reward function for the GFlowNet, which is trained with the trajectory balance objective to sample from the evolving EBM;
(2) **EBM updating step:** the EBM is trained with soft Q iteration [14, §3.1] where the GFlowNet provides diverse samples.

Moreover, again inspired by soft Q-learning [14], we find it advantageous to evaluate the forward policy, backward policy and total flow function in (10) with a separate target network, where the parameters $\bar{\theta}_F$, $\bar{\theta}_B$ and $\bar{\theta}_Z$ are updated softly [24].

4 EXPERIMENTS

In this section, we will empirically validate DPO on two RL problems with structured action space, which include ATSC tasks [2] where atomic actions have *physical* local dependencies; and more generally, Battle scenarios [58] where atomic actions have *logical* local dependencies (see Appendix for more environment details). It is worth noting that we did not use the *population diversity* (PD) proposed by Parker-Holder et al. [34] or the modified PD proposed by Zhou et al. [59] as one of the evaluation metrics. In our experiments, we find that due to the high dimensionality and local dependencies of structured actions, PD, a locality indicator, cannot well reflect the diversity of policies. Therefore, we evaluate different global metrics for different tasks to verify the diversity.

Algorithm 1 Joint Training Framework of *DPO*

```
$\{\theta_Q, \theta_F, \theta_B, \theta_Z\} \sim$ some initialization distributions, assign target parameters $\{\bar{\theta}_Q, \bar{\theta}_F, \bar{\theta}_B, \bar{\theta}_Z\}$, $\mathcal{D} \leftarrow$ empty replay buffer;
for each epoch until some convergence conditions do
    for each timestep $t$ until the maximum limitation do
        Sample an structured action $a_e^t$ via $P_{F|e}(\cdot|\cdot, s_e^t; \theta_F)$;
        Save the new experience: $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{\left(s_e^t, a_e^t, r_e^t, s_e^{t+1}\right)\right\}$;
        Sample a minibatch: $\{(s_e^{(i)}, a_e^{(i)}, r_e^{(i)}, s_e'^{(i)})\}_{i=0}^N \sim \mathcal{D}$.
        EBM updating step:
            Update $\theta_Q$ according to computed empirical gradient in (5) and empirical soft values in (6);
        GFlowNet updating step:
            Update $\{\theta_F, \theta_B, \theta_Z\}$ with computed empirical gradient of (10), update $\theta_Z$ with MSE loss with computed empirical soft values additionally;
        Update target parameters similar with Lillicrap et al. [24].
```

4.1 Adaptive Signal Traffic Control

We choose the following algorithms as baselines, mainly including the state-of-the-art methods for the ASTC task and for encouraging policy diversity: **Max-Pressure** control (MP) where the phase combination with the maximal joint pressure is enabled as described in [5]; **MPLight**-implementation is based on the FRAP open source implementation [57] along with the ChainerRL [9] DQN implementation and pressure sensing; **DvD** [34] is a population-based RL method for effective diversity; **SQL** [14] method is the skeleton of the proposed DPO, which can obtain diverse policies in the low-dimensional continuous action space; Recent proposed **RSPO** [59] transforms the problem of seeking diversity policies into a constrained Markov decision process.

Table 1: Performance (↓) on the ATSC benchmark.

MP	Ing. Reg.	Col. Reg.	***DvD***	Ing. Reg.	Col. Reg.
Avg. Delay	59.64	22.06	Avg. Delay	73.22	55.91
Avg. Trip Time	197.23	86.02	Avg. Trip Time	212.81	115.54
Avg. Wait	20.19	5.46	Avg. Wait	31.36	28.35
Avg. Queue	0.8	0.38	Avg. Queue	1.42	2.28
SQL	Ing. Reg.	Col. Reg.	***RSPO***	Ing. Reg.	Col. Reg.
Avg. Delay	67.65	58.32	Avg. Delay	90.42	57.28
Avg. Trip Time	205.44	116.29	Avg. Trip Time	226.5	120.53
Avg. Wait	26.45	30.01	Avg. Wait	44.16	28.19
Avg. Queue	1.15	2.06	Avg. Queue	1.74	2.59
MPLight	Ing. Reg.	Col. Reg.	***DPO***	Ing. Reg.	Col. Reg.
Avg. Delay	78.16	60.42	Avg. Delay	**57.2**	**20.28**
Avg. Trip Time	215.72	123.93	Avg. Trip Time	**192.75**	**81.42**
Avg. Wait	34.57	30.34	Avg. Wait	**18.26**	**4.77**
Avg. Queue	1.48	2.33	Avg. Queue	**0.65**	**0.32**

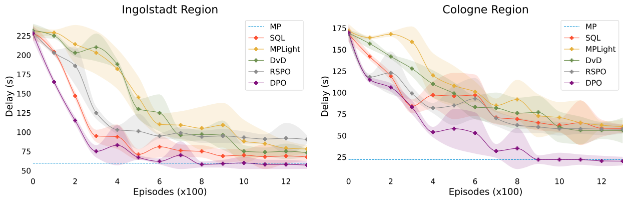

Figure 5: Learning curves of decay (↓) on the ATSC.

From the experimental results in Table 1 and Figure 5, it can be seen that DPO achieves state-of-the-art (SOTA) performance and convergence speed on two coordinated control tasks in TAPAS Cologne and InTAS scenarios. It is worth noting that classical MP methods based on heuristic rules and expert knowledge also show good results. DPO can outperform the MP method through a reinforcement learning mechanism, showing its superiority in solving the ATSC problem. While among the three algorithms that encourage policy diversity, the DvD performs the worst, which we believe is due to the limitations of how it computes the distance between two policies on complex problems. The other two algorithms, SQL and RSPO, can show near-SOTA performance on small-scale problems, i.e., the TAPAS Cologne scenario where a structured action consists of 8 atomic actions. However, in the larger-scale InTAS scenario, its performance drops sharply, which shows that existing algorithms that encourage policy diversity have certain limitations when dealing with structured action spaces.

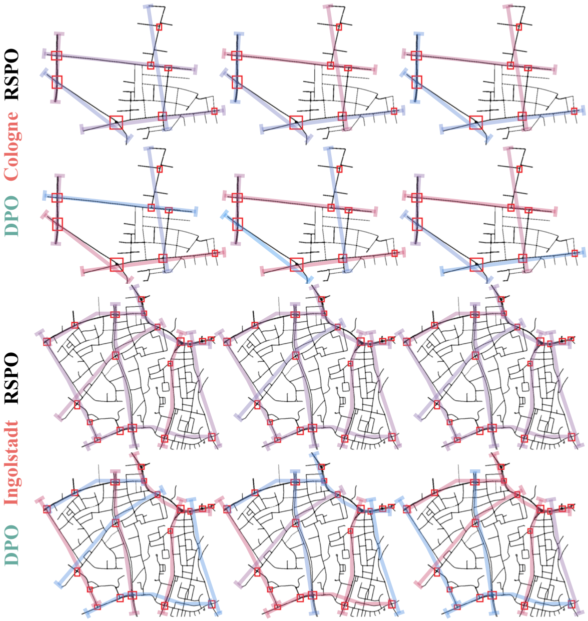

Figure 6: Comparison of policy diversity between DPO and RSPO under the ATSC benchmark. Different colors represent different commute times.

Figure 6 shows the comparison of the policy diversity between RSPO and DPO (see the appendix for more results). We ignore the atomic action level, that is, the diversity of each traffic light's phase selection strategy, but the diversity of the entire road network's traffic control strategy. To this end, we calculate the average commute time of the main road under multiple random seeds for different algorithms in different scenarios. Furthermore, for visualization convenience, we normalized each algorithm separately. Red indicates longer commute time; otherwise, it is shown in blue. As seen from the figure, DPO learns policies with sufficient diversity in structured action spaces of different scales, but RSPO only shows some effect in small-scale tasks.

4.2 Battle Scenario

In the Battle scenario, the atomic action is each agent's action, and we transform the logical correlation between each agent into the physical correlation without loss of generality. Expressly, we assume that atomic actions with local logical correlations have a fixed influence range with a radius $d = 4$ in Euclidean space. In this benchmark, we additionally select **IDQN**, the built-in algorithm in the MAgent, and **MFQ** [51], the state-of-the-art algorithm on the Battle as baselines.

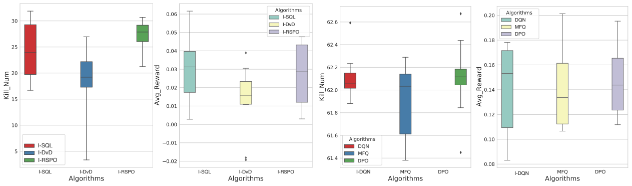

Figure 7: Boxplot of average kill number (↑) and average agent reward (↑) of 50 runs on Battle Game. Results compares the average agent number of blue army killed by red army (left part of each figure) and the average individual rewards of each agent (right part of each figure) respectively.

We first train the IDQN in a self-play way and the blue agent loads the checkpoint and fixes the model parameters. The red agent is then trained with different algorithms, and the final result is shown in Figure 7. It is worth noting that DvD, SQL, and RSPO are less scalable. So in the Battle scenario, we combine independent learning to obtain I-DvD, I-SQL, and I-RSPO variants. Independent learning does not constrain the algorithm's performance, while the IDQN algorithm also shows promising results. As seen from the figure, the three algorithms that encourage policy diversity do not show good results in large-scale structured action spaces, while DPO can still stably approach the performance of SOTA.

Figure 8 shows the diversity of policies between I-RSPO and DPO in the early and middle stages of the game (see appendix for more results). As seen from the figure, the policies learned by DPO show a variety of deployment strategies in the early stage; in the middle stage, the enemy can be surrounded by different formations to maximize the attack power. Although I-RSPO based on independent learning shows a specific diversity at the individual level, it is not easy to generate different policies as a whole.

Diverse policies are more difficult to be exploited by opponents in competitive scenarios and can better adapt to changes in opponents' policies. In order to verify the above point, we let the red agents trained based on different algorithms compete against each other and count the average winning rate. The results are shown in Figure 9. As seen from the figure, DPO shows good robustness against different opponents.

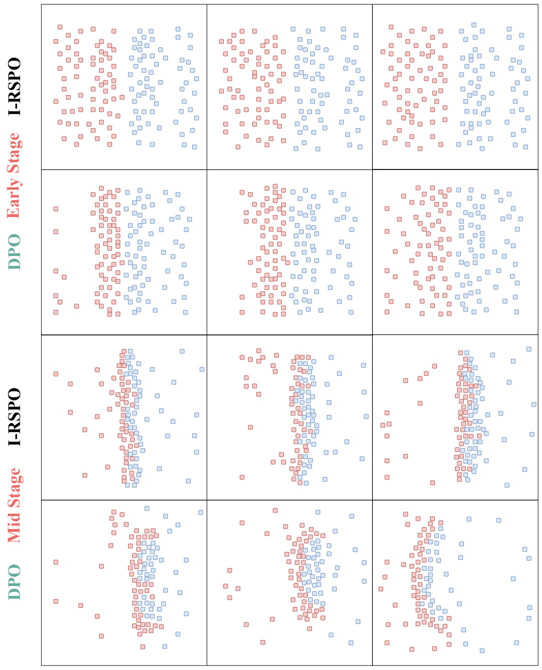

Figure 8: Comparison of policy diversity between DPO and RSPO under the early and middle stages of the Battle.

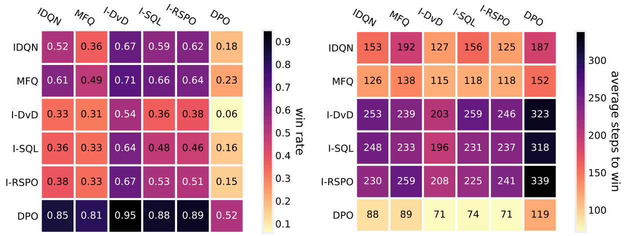

Figure 9: The heatmap of the (Left) win ratio (↑) and (Right) average steps to win (↓) among DPO and others of the testing phase of the Battle benchmark.

5 CLOSING REMARKS

In this paper, we aim to seek diverse policies in an under-explored setting, namely RL tasks with *structured action spaces* with the *composability* and *local dependencies*. The complex action structure, non-uniform reward landscape, and subtle hyperparameter tuning due to the structured actions prevent existing methods from scaling well. We propose a simple and effective method, *Diverse Policy Optimization (DPO)*, to model the policies in structured action space as the energy-based models by following the probabilistic RL framework. DPO adopts a joint training framework, where the energy-based model, and the generative flow network, which is introduced as the efficient, diverse EBM-based policy sampler, are optimized alternately: The energy function serves as the negative log-reward function for the GFlowNet, which is trained with the trajectory balance objective to sample from the evolving energy-based policies. In contrast, the energy function is trained with soft Bellman backup, where the GFlowNet provides diverse samples. Experiments demonstrate that the proposed DPO is both general and practical across structured action spaces with physical and, more generally, logical local dependencies.

ACKNOWLEDGMENTS

This work was supported in part by Postdoctoral Science Foundation of China (2022M723039), NSFC (62106213, 72150002), SSTP (RCBS20210609104356063, JCYJ20210324120011032), and a grant from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

REFERENCES

[1] Safa Alver and Doina Precup. 2022. Constructing a Good Behavior Basis for Transfer using Generalized Policy Updates. In *ICLR*.

[2] James Ault and Guni Sharon. 2021. Reinforcement Learning Benchmarks for Traffic Signal Control. In *NeurIPS*.

[3] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. 2021. Flow network based generative models for non-iterative diverse candidate generation. In *NeurIPS*.

[4] Yoshua Bengio, Tristan Deleu, Edward J Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. 2021. Gflownet foundations. *arXiv preprint arXiv:2111.09266* (2021).

[5] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI*.

[6] Kenneth Derek and Phillip Isola. 2021. Adaptable Agent Populations via a Generative Model of Policies. In *NeurIPS*.

[7] Miguel Duarte, Jorge Gomes, Sancho Moura Oliveira, and Anders Lyhne Christensen. 2017. Evolution of repertoire-based control for robots with complex locomotor systems. *IEEE Transactions on Evolutionary Computation* 22, 2 (2017), 314–328.

[8] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *ICLR*.

[9] Yasuhiro Fujita, Prabhat Nagarajan, Toshiki Kataoka, and Takahiro Ishikawa. 2021. Chainerrl: A deep reinforcement learning library. *The Journal of Machine Learning Research* 22, 1 (2021), 3557–3570.

[10] Tanmay Gangwani, Qiang Liu, and Jian Peng. 2019. Learning Self-Imitating Diverse Policies. In *ICLR*.

[11] Tanmay Gangwani, Jian Peng, and Yuan Zhou. 2021. Harnessing Distribution Ratio Estimators for Learning Agents with Quality and Diversity. In *CoRL*.

[12] Anirudh Goyal, Shagun Sodhani, Jonathan Binas, Xue Bin Peng, Sergey Levine, and Yoshua Bengio. 2020. Reinforcement Learning with Competitive Ensembles of Information-Constrained Primitives. In *ICLR*.

[13] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. 2016. Variational intrinsic control. *arXiv preprint arXiv:1611.07507* (2016).

[14] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *ICML*.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*.

[16] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. 2020. Dynamical Distance Learning for Semi-Supervised and Unsupervised Skill Discovery. In *ICLR*.

[17] Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. 2022. Biological Sequence Design with GFlowNets. In *ICML*.

[18] Ajay Jasra, Chris C Holmes, and David A Stephens. 2005. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statist. Sci.* 20, 1 (2005), 50–67.

[19] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*.

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[21] Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. 2020. One solution is not all you need: Few-shot extrapolation via structured maxent rl. In *NeurIPS*.

[22] Sergey Levine. 2018. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *ArXiv* abs/1805.00909 (2018).

[23] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *EMNLP*.

[24] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[25] Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. 2022. Dynamics-Aware Quality-Diversity for Efficient Learning of Skill Repertoires. In *ICRA*.

[26] Qiang Ma, Suwen Ge, Danyang He, Darshan Thaker, and Iddo Drori. 2019. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. *arXiv preprint arXiv:1911.04936* (2019).

[27] Tengyu Ma. 2021. Why Do Local Methods Solve Nonconvex Problems? *Beyond the Worst-Case Analysis of Algorithms* (2021), 465.

[28] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. 2022. Trajectory Balance: Improved Credit Assignment in GFlowNets. *arXiv preprint arXiv:2201.13259* (2022).

[29] Muhammad A Masood and Finale Doshi-Velez. 2019. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *arXiv preprint arXiv:1906.00088* (2019).

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[31] Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted map-elites. In *GECCO*.

[32] Alex F Osborn. 1953. *Applied imagination.* Scribner's.

[33] Christos H Papadimitriou. 1977. The Euclidean travelling salesman problem is NP-complete. *Theoretical computer science* 4, 3 (1977), 237–244.

[34] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. 2020. Effective diversity in population based reinforcement learning. In *NeurIPS*.

[35] Zhenghao Peng, Hao Sun, and Bolei Zhou. 2020. Non-local policy optimization via diversity-regularized collaborative exploration. *arXiv preprint arXiv:2006.07781* (2020).

[36] Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P. Arrais. 2021. Diversity oriented Deep Reinforcement Learning for targeted molecule generation. *Journal of Cheminformatics* 13 (2021).

[37] Emilia Pompe, Chris Holmes, and Krzysztof Łatuszyński. 2020. A framework for adaptive MCMC targeting multimodal distributions. *The Annals of Statistics* 48, 5 (2020), 2930–2952.

[38] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* (2016), 40.

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

[40] Archit Sharma, Michael Ahn, Sergey Levine, Vikash Kumar, Karol Hausman, and Shixiang Gu. 2020. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. *arXiv preprint arXiv:2004.12974* (2020).

[41] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *ICLR*.

[42] Hao Sun, Zhenghao Peng, Bo Dai, Jian Guo, Dahua Lin, and Bolei Zhou. 2020. Novel policy seeking with constrained optimization. *arXiv preprint arXiv:2005.10696* (2020).

[43] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction.*

[44] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *ICLR*.

[45] Elise Van der Pol and Frans A Oliehoek. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)* 1 (2016).

[46] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *NeurIPS*.

[47] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. 2019. POET: open-ended coevolution of environments and their optimized solutions. *GECCO* (2019).

[48] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *KDD*.

[49] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *KDD*.

[50] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[51] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean field multi-agent reinforcement learning. In *ICML*.

[52] Tom Zahavy, André Barreto, Daniel Jaymin Mankowitz, Shaobo Hou, Brendan O'Donoghue, Iurii Kemaev, and Satinder Singh. 2021. Discovering a set of policies for the worst case reward. *ArXiv* abs/2102.04323 (2021).

[53] Tom Zahavy, Avinatan Hasidim, Haim Kaplan, and Yishay Mansour. 2020. Planning in hierarchical reinforcement learning: Guarantees for using local policies. In *ALT*.

[54] Tom Zahavy, Yannick Schroecker, Feryal M. P. Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. 2022. Discovering Policies with DOMiNO: Diversity Optimization Maintaining Near Optimality. *ArXiv* abs/2205.13521 (2022).

[55] Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. 2022. Generative Flow Networks for Discrete

Probabilistic Modeling. In *ICML*.

[56] Yunbo Zhang, Wenhao Yu, and Greg Turk. 2019. Learning novel policies for tasks. In *ICML*.

[57] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. 2019. Learning phase competition for traffic signal control. In *CIKM*.

[58] Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *AAAI*.

[59] Zihan Zhou, Wei Fu, Bingliang Zhang, and Yi Wu. 2022. Continuously Discovering Novel Strategies via Reward-Switching Policy Optimization. In *ICLR*.