GDI: Rethinking What Makes Reinforcement Learning Different from Supervised Learning

Jiajun Fan,[1] **Changnan Xiao,** [2] **Yue Huang** [2]
[1] Tsinghua University
[2] ByteDance Inu
fanjj21@mails.tsinghua.edu.cn, xiaochangnan@bytedance.com, yuehuanghit@gmail.com

Abstract

Deep Q Network (DQN) firstly kicked the door of deep reinforcement learning (DRL) via combining deep learning (DL) with reinforcement learning (RL), which has noticed that the distribution of the acquired data would change during the training process. DQN found this property might cause instability for training, so it proposed effective methods to handle the downside of the property. Instead of focusing on the unfavorable aspects, we find it critical for RL to ease the gap between the estimated data distribution and the ground truth data distribution while supervised learning (SL) fails to do so. From this new perspective, we extend the basic paradigm of RL called the Generalized Policy Iteration (GPI) into a more generalized version, which is called the Generalized Data Distribution Iteration (GDI). We see massive RL algorithms and techniques can be unified into the GDI paradigm, which can be considered as one of the special cases of GDI. We provide theoretical proof of why GDI is better than GPI and how it works. Several practical algorithms based on GDI have been proposed to verify its effectiveness and extensiveness. Empirical experiments prove our state-of-the-art (SOTA) performance on Arcade Learning Environment (ALE), wherein our algorithm has achieved **9620.98%** mean human normalized score (HNS), **1146.39%** median HNS and **22** human world record breakthroughs (HWRB) using only **200M** training frames. Our work aims to lead the RL research to step into the journey of conquering the human world records and seek real superhuman agents on both performance and efficiency.

Introduction

Machine learning (ML) can be defined as improving some measure performance P at some task T according to the acquired data or experience E (Mitchell et al. 1997). As one of the three main components of ML (Mitchell et al. 1997), the training experiences matter in ML, which can be reflected from many aspects. For example, three major ML paradigms can be distinguished from the perspective of the different training experiences. Supervised learning (SL) is learning from a training set of **labeled experiences** provided by a knowledgable external supervisor (Sutton and Barto 2018). Unsupervised learning (UL) is typically about seeking structure hidden in collections of **unlabeled experiences** (Sutton and Barto 2018). Unlike UL or SL, reinforcement learning (RL) focuses on the problem that agents learn from **experiences gained through trial-and-error interactions with a dynamic environment**. As (Mitchell et al. 1997) said, there is no free lunch in the ML problem - no way to generalize beyond the specific training examples. The performance can only be improved through learning from the acquired experiences in ML problems (Mitchell et al. 1997). All of them have revealed the importance of the training experiences and thus the selection of the training distribution appears to be a fundamental problem in ML.

Recalling these three paradigms, SL and RL receive explicit learning signals from data. In SL, there is no way to make up the gap between the distribution estimated by the collected data and the ground truth without any domain knowledge unless collecting more data. Researchers have found RL explicitly and naturally transforming the training distribution (Mnih et al. 2015), which makes RL distinguished from SL. In the recent RL advances, many researchers (Mnih et al. 2015) have realized that RL agents hold the property of changing the data distribution and massive works have revealed the unfavorable aspect of the property. Among those algorithms, DQN (Mnih et al. 2015) firstly noticed the unique property of RL and considered it as one of the reasons for the training instability of DRL. After that, massive methods like replay buffer (Mnih et al. 2015), periodically updated target (Mnih et al. 2015) and importance sampling (Espeholt et al. 2018) have been proposed to mitigate the impact of the data distribution shift. However, after rethinking this property, we wonder whether changing the data distribution always brings unfavorable nature. What if we can control it? More precisely, what if we can control the ability to select superior data distribution for training automatically? Prior works in ML have revealed the great potential of this property. As (Cohn, Ghahramani, and Jordan 1996) put it, when training examples are appropriately selected, the data requirements for some problems decrease drastically, and some NP-complete learning problems become polynomial in computation time, which means that carefully selecting good training data benefits learning efficiency. Inspired by this perspective, instead of discussing how to ease the disadvantages caused by the change of data distribution like other prior works of RL, in this paper, we rethink the property distinguishing RL from SL and explore more effective aspects of it. One of the fundamental reasons RL holds the ability to change the data distribution is the change of behavior policies, which directly interact with the dynamic environments to obtain training data (Mnih et al. 2015). Therefore, the training experiences can be controlled by adjusting the behavior policies, which makes behavior selection the bridge between RL agents and training examples.

In the RL problem, the agent has to exploit what it already knows to obtain the reward, but it also has to explore to make better action selections in the future, which is called the exploration and exploitation dilemma (Sutton and Barto 2018). Therefore, diversity is one of the main factors that should be considered while selecting the training examples. In the recent advances of RL, some works have also noticed the importance of the diversity of training experiences (Badia et al. 2020a,b; Parker-Holder et al. 2020; Eysenbach et al. 2018), most of which have obtained diverse data via enriching the policy diversity. Among those algorithms, DIAYN (Eysenbach et al. 2018) focused entirely on the diversity of policy via learning skills without a reward function, which has revealed the effect of policy diversity but ignored its relationship with the RL objective. DvD (Parker-Holder et al. 2020) introduced a diversity-based regularizer into the RL objective to obtain more diverse data, which changed the optimal solution of the environment (Sutton and Barto 2018). Besides, training a population of agents to gather more diverse experiences seems to be a promising approach. Agent57 (Badia et al. 2020a) and NGU (Badia et al. 2020b) trained a family of policies with different degrees of exploratory behaviors using a shared network architecture. Both of them have obtained SOTA performance at the cost of increasing the uncertainty of environmental transition, which leads to extremely low learning efficiency. Through those successes, it is evident that the diversity of the training data benefit the RL training. However, why does it perform better and whether more diverse data always benefit RL training? In other words, we have to explore the following question:

Does diverse data always benefit effective learning?

To investigate this problem, we seek inspiration from the natural biological processes. In nature, the population evolves typically faster than individuals because the diversity of the populations boosts more **beneficial mutations** which provide more possibility for acquiring more adaptive direction of evolution (Pennisi 2016). Furthermore, beneficial mutations rapidly spread among the population, thus enhancing population adaptability (Pennisi 2016). Therefore, an appropriate diversity brings high-value individuals, and active learning among the population promotes its prosperity.[1] From this perspective, the RL agents have to pay more attention to **experiences worthy of learning from**. DisCor (Kumar, Gupta, and Levine 2020), which re-weighted the **existing data buffer** by the distribution that explicitly optimizes for corrective feedback, has also noticed the fact that **the choice of the sampling distribution is of crucial importance for the stability and efficiency of approximation dynamic programming algorithms**. Unfortunately, DisCor only changes the existing data distribution instead of directly controlling the source of the training experiences, which may be more important and also more complex. In conclusion, it seems that both **expanding the capacity of policy space for behaviors** and **selecting suitable behavior policies from a diverse behavior population** matter for efficient learning. This new perspective motivates us to investigate another critical problem:

How to select superior behaviors from the behavior policy space?

To address those problems, we proposed a novel RL paradigm called **G**eneralized **D**ata Distribution **I**teration (**GDI**), which consists of two major process, the policy iteration operator $\mathcal{T}$ and the data distribution iteration operator $\mathcal{E}$. Specifically, behaviors will be sampled from a policy space according to a selective distribution, which will be iteratively optimized through the operator $\mathcal{E}$. Simultaneously, elite training data will be used for policy iteration via the operator $\mathcal{T}$. More details about our methodology can see Sec. .

In conclusion, the main contributions of our work are:

- **A Novel RL Paradigm:** Rethinking the difference between RL and SL, we discover RL can ease the gap between the sampled data distribution and the ground truth data distribution via adjusting the behavior policies. Based on the perspective, we extend GPI into GDI, a more general version containing a data optimization process. This novel perspective allows us to unify massive RL algorithms, and various improvements can be considered a special case of data distribution optimization, detailed in Sec. .
- **Theoretical Proof of GDI:** We provide sufficient theoretical proof of GDI. The effectiveness of the data distribution optimization of GDI has been proved on both first-order optimization and second-order optimization, and the guarantee of monotonic improvement induced by the data distribution optimization operator $\mathcal{E}$ has also been proved. More details can see Sec. .
- **A General Practical Framework of GDI:** Based on GDI, we propose a general practical framework, wherein behavior policy belongs to a soft ϵ-greedy space which unifies ϵ-greedy policies (Watkins 1989) and Boltzmann policies (Wiering 1999). As a practical framework of GDI, a self-adaptable meta-controller is proposed to optimize the distribution of the behavior policies. More implementation details can see the appendix of (Fan, Xiao, and Huang 2021).
- **The State-Of-The-Art Performance:** From Figs. 1, our approach has achieved 9620.98% mean HNS and 1146.39% median HNS, which achieves new SOTA. More importantly, our learning efficiency has approached the human level as achieving the SOTA performance within less than 1.5 months of game time.
- **Human World Records Breakthrough:** As our algorithms have achieved SOTA on mean HNS, median HNS and learning efficiency, we aim to lead RL research on ALE to step into a new era of conquering human world records and seeking the real superhuman agents. Therefore, we propose several novel evaluation criteria and an open challenge on the Atari benchmark based on the

[1] According to (LaBar and Adami 2017), most mutations are deleterious and cause a reduction in population fitness known as the mutational load. Therefore, excessive and redundant diversity may be harmful.

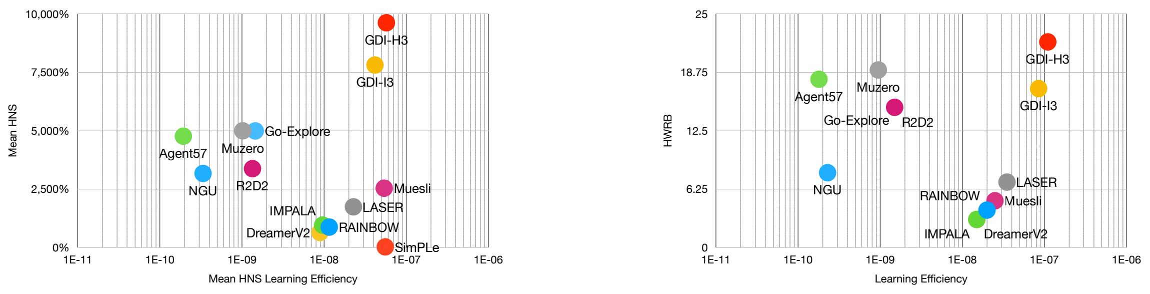

Figure 1: Performance of SOTA algorithms of Atari 57 games on mean HNS(%) with corresponding learning efficiency and human world record breakthrough with corresponding game time. Details on those evaluation criteria can see the appendix of (Fan, Xiao, and Huang 2021).

human world records. From Figs. 1, our method has surpassed 22 human world records, which has also surpassed all previous algorithms. The RL Benchmark on human world records normalized scores (HWRNS), SABER (Toromanoff, Wirbel, and Moutarde 2019) and HWRB can be found in the appendix of (Fan, Xiao, and Huang 2021), respectively. Relevant scores can see the appendix of (Fan, Xiao, and Huang 2021).

Preliminaries

The RL problem can be formulated as a Markov Decision Process (Howard 1960, MDP) defined by $(\mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0)$. Considering a discounted episodic MDP, the initial state s_0 is sampled from the initial distribution $\rho_0(s) : \mathcal{S} \rightarrow \Delta(\mathcal{S})$, where we use Δ to represent the probability simplex. At each time t, the agent chooses an action $a_t \in \mathcal{A}$ according to the policy $\pi(a_t|s_t) : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ at state $s_t \in \mathcal{S}$. The environment receives a_t, produces the reward $r_t \sim r(s,a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbf{R}$ and transfers to the next state s_{t+1} according to the transition distribution $p(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$. The process continues until the agent reaches a terminal state or a maximum time step. Define the discounted state visitation distribution as $d_{\rho_0}^{\pi}(s) = (1-\gamma)\mathbf{E}_{s_0 \sim \rho_0}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{P}(s_t = s|s_0)\right]$. The goal of reinforcement learning is to find the optimal policy π^* that maximizes the expected sum of discounted rewards, denoted by $\mathcal{J}$ (Sutton and Barto 2018):

$$\begin{aligned}
\pi^* &= \underset{\pi}{\operatorname{argmax}} \mathcal{J}_\pi \\
&= \underset{\pi}{\operatorname{argmax}} \mathbf{E}_{s_t \sim d_{\rho_0}^{\pi}} \mathbf{E}_\pi \left[G_t|s_t\right] \\
&= \underset{\pi}{\operatorname{argmax}} \mathbf{E}_{s_t \sim d_{\rho_0}^{\pi}} \mathbf{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}|s_t\right]
\end{aligned} \tag{1}$$

where $\gamma \in (0,1)$ is the discount factor.

RL algorithms can be divided into off-policy manners (Mnih et al. 2015, 2016; Haarnoja et al. 2018; Espeholt et al. 2018) and on-policy manners (Schulman et al. 2017). Off-policy algorithms select actions according to a behavior policy μ that can be different from the learning policy π. On-policy algorithms evaluate and improve the learning policy through data sampled from the same policy. RL algorithms can also be divided into value-based methods (Mnih et al. 2015; Van Hasselt, Guez, and Silver 2016; Wang et al. 2016; Hessel et al. 2017; Horgan et al. 2018) and policy-based methods (Schulman et al. 2017; Mnih et al. 2016; Espeholt et al. 2018; Schmitt, Hessel, and Simonyan 2020). In the value-based methods, agents learn the policy indirectly, where the policy is defined by consulting the learned value function, like ϵ-greedy, and the value function is learned by a typical GPI. In the policy-based methods, agents learn the policy directly, where the correctness of the gradient direction is guaranteed by the policy gradient theorem (Sutton and Barto 2018), and the convergence of the policy gradient methods is also guaranteed (Agarwal et al. 2019).

Methodology

Generalized Data Distribution Iteration

Let's abstract our notations first.

Define Λ to be an index set, $\Lambda \subseteq \mathbf{R}^k$. $\lambda \in \Lambda$ is an index in Λ. $(\Lambda, \mathcal{B}|_\Lambda, \mathcal{P}_\Lambda)$ is a probability space, where $\mathcal{B}|_\Lambda$ is a Borel σ-algebra restricted to Λ. Under the setting of meta-RL, Λ can be regarded as the set of all possible meta information. Under the setting of population-based training (PBT) (Jaderberg et al. 2017), Λ can be regarded as the set of the whole population.

Define Θ to be a set of all possible values of parameters. $\theta \in \Theta$ is some specific value of parameters. For each index λ, there exists a specific mapping between each parameter of θ and λ, denoted as θ_λ, to indicate the parameters in θ corresponding to λ. Under the setting of linear regression $y = w \cdot x$, $\Theta = \{w \in R^n\}$ and $\theta = w$. If λ represents using only the first half features to make regression, assume $w = (w_1, w_2)$, then $\theta_\lambda = w_1$. Under the setting of RL, θ_λ defines a parameterized policy indexed by λ, denoted as π_{θ_λ}.

Define $\mathcal{D} \overset{def}{=} \{d_{\rho_0}^{\pi} | \ \pi \in \Delta(\mathcal{A})^{\mathcal{S}}, \rho_0 \in \Delta(\mathcal{S})\}$ to be the set of all states visitation distributions. For the parameterized policies, denote $\mathcal{D}_{\Lambda,\Theta,\rho_0} \overset{def}{=} \{d_{\rho_0}^{\pi_{\theta_\lambda}} | \ \theta \in \Theta, \lambda \in \Lambda\}$. Note that $(\Lambda, \mathcal{B}|_\Lambda, \mathcal{P}_\Lambda)$ is a probability space on Λ, which induces a probability space on $\mathcal{D}_{\Theta,\Lambda,\rho_0}$, with the probability measure given by $\mathcal{P}_\mathcal{D}(\mathcal{D}_{\Lambda_0,\Theta,\rho_0}) = \mathcal{P}_\Lambda(\Lambda_0),\ \forall \Lambda_0 \in \mathcal{B}|_\Lambda$.

We use x to represent one sample, which contains all necessary information for learning. For DQN, $x = (s_t, a_t, r_t, s_{t+1})$. For R2D2, $x = (s_t, a_t, r_t, \dots, s_{t+N}, a_{t+N}, r_{t+N}, s_{t+N+1})$. For IMPALA, x also contains the distribution of the behavior policy. The content of x depends on the algorithm, but it's sufficient for learning. We use $\mathcal{X}$ to represent the set of samples. At training stage t, given the parameter $\theta = \theta^{(t)}$, the distribution of the index set $\mathcal{P}_\Lambda = \mathcal{P}_\Lambda^{(t)}$ and the distribution of the initial state ρ_0, we denote the set of samples as

$$\begin{aligned}\mathcal{X}_{\rho_0}^{(t)} &\overset{def}{=} \bigcup_{d_{\rho_0}^{\pi} \sim \mathcal{P}_\mathcal{D}^{(t)}} \{x | x \sim d_{\rho_0}^{\pi}\} \\ &= \bigcup_{\lambda \sim \mathcal{P}_\Lambda^{(t)}} \{x | x \sim d_{\rho_0}^{\pi_\theta}, \theta = \theta_\lambda^{(t)}\} \triangleq \bigcup_{\lambda \sim \mathcal{P}_\Lambda^{(t)}} \mathcal{X}_{\rho_0,\lambda}^{(t)}.\end{aligned}$$

Now we introduce our main algorithm:

Algorithm 1: Generalized Data Distribution Iteration (GDI).

```
Initialize $\Lambda$, $\Theta$, $\mathcal{P}_\Lambda^{(0)}$, $\theta^{(0)}$.
**for** $t = 0, 1, 2, \dots$ **do**
    Sample $\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}}$.
    $\theta^{(t+1)} = \mathcal{T}(\theta^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$.
    $\mathcal{P}_\Lambda^{(t+1)} = \mathcal{E}(\mathcal{P}_\Lambda^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$.
**end for**
```

$\mathcal{T}$ defined as $\theta^{(t+1)} = \mathcal{T}(\theta^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$ is a typical optimization operator of RL algorithms, which utilizes the collected samples to update the parameters for maximizing some function $L_\mathcal{T}$. For instance, $L_\mathcal{T}$ may contain the policy gradient and the state value evaluation for the policy-based methods, may contain generalized policy iteration for the value-based methods, may also contain some auxiliary tasks or intrinsic rewards for special designed methods.

$\mathcal{E}$ defined as $\mathcal{P}_\Lambda^{(t+1)} = \mathcal{E}(\mathcal{P}_\Lambda^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$ is a data distribution optimization operator. It uses the samples $\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}}$ to maximize some function $L_\mathcal{E}$, namely,

$$\mathcal{P}_\Lambda^{(t+1)} = \arg\max_{\mathcal{P}_\Lambda} L_\mathcal{E}(\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda}).$$

When $\mathcal{P}_\Lambda$ is parameterized, we abuse the notation and use $\mathcal{P}_\Lambda$ to represent the parameter of $\mathcal{P}_\Lambda$. If $\mathcal{E}$ is a first order optimization operator, then we can write $\mathcal{E}$ explicitly as

$$\mathcal{P}_\Lambda^{(t+1)} = \mathcal{P}_\Lambda^{(t)} + \eta \nabla_{\mathcal{P}_\Lambda^{(t)}} L_\mathcal{E}(\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}}).$$

If $\mathcal{E}$ is a second order optimization operator, like natural gradient, we can write $\mathcal{E}$ formally as

$$\begin{aligned}\mathcal{P}_\Lambda^{(t+1)} &= \mathcal{P}_\Lambda^{(t)} + \eta \mathbf{F}(\mathcal{P}_\Lambda^{(t)})^\dagger \nabla_{\mathcal{P}_\Lambda^{(t)}} L_\mathcal{E}(\{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}}), \\ \mathbf{F}(\mathcal{P}_\Lambda^{(t)}) &= \left[\nabla_{\mathcal{P}_\Lambda^{(t)}} \log \mathcal{P}_\Lambda^{(t)}\right] \cdot \left[\nabla_{\mathcal{P}_\Lambda^{(t)}} \log \mathcal{P}_\Lambda^{(t)}\right]^\top,\end{aligned}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix.

Systematization of GDI

We can further divide all algorithms into two categories, GDI-I^n and GDI-H^n. n represents the degree of freedom of Λ. I represents Isomorphism. We say one algorithm belongs to GDI-I^n, if $\theta = \theta_\lambda, \forall \lambda \in \Lambda$. H represents Heterogeneous. We say one algorithm belongs to GDI-H^n, if $\theta_{\lambda_1} \neq \theta_{\lambda_2}, \exists \lambda_1, \lambda_2 \in \Lambda$. We say one algorithm is "w/o $\mathcal{E}$" if it doesn't have the operator $\mathcal{E}$, in another word, its $\mathcal{E}$ is an identical mapping.

Now we discuss the connections between GDI and some algorithms.

For DQN, RAINBOW, PPO and IMPALA, they are in GDI-I^0 w/o $\mathcal{E}$. Let $|\Lambda| = 1$, WLOG, assume $\Lambda = \{\lambda_0\}$. The probability measure $\mathcal{P}_\Lambda$ collapses to $\mathcal{P}_\Lambda(\lambda_0) = 1$. $\Theta = \{\theta_{\lambda_0}\}$. $\mathcal{E}$ is an identical mapping of $\mathcal{P}_\Lambda^{(t)}$. $\mathcal{T}$ is the first order operator that optimizes the loss functions, respectively.

For Ape-X and R2D2, they are in GDI-I^1 w/o $\mathcal{E}$. Let $\Lambda = \{\epsilon_l | \ l = 1, \dots, 256\}$. $\mathcal{P}_\Lambda$ is uniform, $\mathcal{P}_\Lambda(\epsilon_l) = |\Lambda|^{-1}$. Since all actors and the learner share parameters, we have $\theta_{\epsilon_1} = \theta_{\epsilon_2}$ for $\forall \epsilon_1, \epsilon_2 \in \Lambda$, hence $\Theta = \bigcup_{\epsilon \in \Lambda} \{\theta_\epsilon\} = \{\theta_{\epsilon_l}\}$, $\forall\ l = 1, \dots, 256$. $\mathcal{E}$ is an identical mapping, because $\mathcal{P}_\Lambda^{(t)}$ is always a uniform distribution. $\mathcal{T}$ is the first order operator that optimizes the loss functions.

For LASER, it's in GDI-H^1 w/o $\mathcal{E}$. Let $\Lambda = \{i | \ i = 1, \dots, K\}$ to be the number of learners. $\mathcal{P}_\Lambda$ is uniform, $\mathcal{P}_\Lambda(i) = |\Lambda|^{-1}$. Since different learners don't share parameters, $\theta_{i_1} \cap \theta_{i_2} = \emptyset$ for $\forall i_1, i_2 \in \Lambda$, hence $\Theta = \bigcup_{i \in \Lambda} \{\theta_i\}$. $\mathcal{E}$ is an identical mapping. $\mathcal{T}$ can be formulated as a union of $\theta_i^{(t+1)} = \mathcal{T}_i(\theta_i^{(t)}, \{\mathcal{X}_{\rho_0,\lambda}^{(t)}\}_{\lambda \sim \mathcal{P}_\Lambda^{(t)}})$, which represents optimizing θ_i of ith learner with shared samples from other learners.

For PBT, it's in GDI-H^{n+1}, where n is the number of searched hyperparameters. Let $\Lambda = \{h\} \times \{i | i = 1, \dots, K\}$, where h represents the hyperparameters being searched and K is the population size. $\Theta = \bigcup_{i=1,\dots,K} \{\theta_{i,h}\}$, where $\theta_{i,h_1} = \theta_{i,h_2}$ for $\forall (h_1, i), (h_2, i) \in \Lambda$. $\mathcal{E}$ is the meta-controller that adjusts h for each i, which can be formally written as $\mathcal{P}_\Lambda^{(t+1)}(\cdot, i) = \mathcal{E}_i(\mathcal{P}_\Lambda^{(t)}(\cdot, i), \{\mathcal{X}_{\rho_0,(h,i)}^{(t)}\}_{h \sim \mathcal{P}_\Lambda^{(t)}(\cdot, i)})$, which optimizes $\mathcal{P}_\Lambda$ according to the performance of all agents in the population. $\mathcal{T}$ can also be formulated as a union of $\mathcal{T}_i$, but is $\theta_i^{(t+1)} = \mathcal{T}_i(\theta_i^{(t)}, \{\mathcal{X}_{\rho_0,(h,i)}^{(t)}\}_{h \sim \mathcal{P}_\Lambda^{(t)}(\cdot, i)})$, which represents optimizing the ith agent with only samples from the ith agent.

For NGU and Agent57, it's in GDI-I^2. Let $\Lambda = \{\beta_i | i = 1, \dots, m\} \times \{\gamma_j | j = 1, \dots, n\}$, where β is the weight of the intrinsic value function and γ is the discount factor. Since all actors and the learner share variables, $\Theta = \bigcup_{(\beta,\gamma)\in\Lambda}\{\theta_{(\beta,\gamma)}\} = \{\theta_{(\beta,\gamma)}\}$ for $\forall(\beta,\gamma)\in\Lambda$. $\mathcal{E}$ is an optimization operator of a multi-arm bandit controller with UCB, which aims to maximize the expected cumulative rewards by adjusting $\mathcal{P}_\Lambda$. Different from above, $\mathcal{T}$ is identical to our general definition $\theta^{(t+1)} = \mathcal{T}(\theta^{(t)}, \{\mathcal{X}^{(t)}_{\rho_0,\lambda}\}_{\lambda\sim\mathcal{P}^{(t)}_\Lambda})$, which utilizes samples from different λs to update the shared θ.

For Go-Explore, it's in GDI-H[1]. Let $\Lambda = \{\tau\}$, where τ represents the stopping time of switching between robustification and exploration. $\Theta = \{\theta_r\}\cup\{\theta_e\}$, where θ_r is the robustification model and θ_e is the exploration model. $\mathcal{E}$ is a search-based controller, which defines the next $\mathcal{P}_\Lambda$ for a better exploration. $\mathcal{T}$ can be decomposed into $(\mathcal{T}_r, \mathcal{T}_e)$.

Monotonic Data Distribution Optimization

We see massive algorithms can be formulated as a special case of GDI. For the algorithms without a meta-controller, whose data distribution optimization operator $\mathcal{E}$ is trivially an identical mapping, the guarantee that the learned policy could converge to the optimal policy has been wildly studied, for instance, GPI in (Sutton and Barto 2018) and policy gradient in (Agarwal et al. 2019). But for the algorithms with a meta-controller, whose data distribution optimization operator $\mathcal{E}$ is non-identical, though most algorithms in this class show superior performance, it still lacks a general study on why the data distribution optimization operator $\mathcal{E}$ helps. In this section, with a few assumptions, we show that given the same optimization operator $\mathcal{T}$, a GDI with a non-identical data distribution optimization operator $\mathcal{E}$ is always superior to a GDI w/o $\mathcal{E}$.

For brevity, we denote the expectation of $L_\mathcal{E}, L_\mathcal{T}$ for each $\lambda\in\Lambda$ as $\mathcal{L}_\mathcal{E}(\lambda,\theta_\lambda) = \mathbf{E}_{x\sim\pi_{\theta_\lambda}}[L_\mathcal{E}(\{\mathcal{X}_{\rho_0,\lambda}\})]$, $\mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda) = \mathbf{E}_{x\sim\pi_{\theta_\lambda}}[L_\mathcal{T}(\{\mathcal{X}_{\rho_0,\lambda}\})]$, and denote the expectation of $L_\mathcal{E}, L_\mathcal{T}$ for any $\mathcal{P}_\Lambda$ as $\mathcal{L}_\mathcal{E}(\mathcal{P}_\Lambda,\theta) = \mathbf{E}_{\lambda\sim\mathcal{P}_\Lambda}[\mathcal{L}_\mathcal{E}(\lambda,\theta_\lambda)]$, $\mathcal{L}_\mathcal{T}(\mathcal{P}_\Lambda,\theta) = \mathbf{E}_{\lambda\sim\mathcal{P}_\Lambda}[\mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda)]$.

Assumption 1 (Uniform Continuous Assumption). *For $\forall\epsilon > 0,\ \forall s\in\mathcal{S},\ \exists\delta > 0,\ s.t.|V^{\pi_1}(s) - V^{\pi_2}(s)| < \epsilon,\ \forall d_\pi(\pi_1,\pi_2) < \delta$, where d_π is a metric on $\Delta(\mathcal{A})^\mathcal{S}$. If π is parameterized by θ, then for $\forall\epsilon > 0,\ \forall s\in\mathcal{S},\ \exists\delta > 0,\ s.t.|V^{\pi_{\theta_1}}(s) - V^{\pi_{\theta_2}}(s)| < \epsilon,\ \forall||\theta_1-\theta_2|| < \delta$.*

Remark. *(Dadashi et al. 2019) shows V^π is infinitely differentiable everywhere on $\Delta(\mathcal{A})^\mathcal{S}$ if $|\mathcal{S}| < \infty, |\mathcal{A}| < \infty$. (Agarwal et al. 2019) shows V^π is β-smooth, namely bounded second order derivative, for direct parameterization. If $\Delta(\mathcal{A})^\mathcal{S}$ is compact, continuity implies uniform continuity.*

Assumption 2 (Formulation of $\mathcal{E}$ Assumption). *Assume $\mathcal{P}^{(t+1)}_\Lambda = \mathcal{E}(\mathcal{P}^{(t)}_\Lambda, \{\mathcal{X}^{(t)}_{\rho_0,\lambda}\}_{\lambda\sim\mathcal{P}^{(t)}_\Lambda})$ can be written as $\mathcal{P}^{(t+1)}_\Lambda(\lambda) = \mathcal{P}^{(t)}_\Lambda(\lambda)\frac{\exp(\eta\mathcal{L}_\mathcal{E}(\lambda,\theta^{(t)}_\lambda))}{Z^{(t+1)}}$, $Z^{(t+1)} = \boldsymbol{E}_{\lambda\sim\mathcal{P}^{(t)}_\Lambda}[\exp(\eta\mathcal{L}_\mathcal{E}(\lambda,\theta^{(t)}_\lambda))]$.*

Remark. *The assumption is actually general. Regarding Λ as an action space and $r_\lambda = \mathcal{L}_\mathcal{E}(\lambda,\theta^{(t)}_\lambda)$, when solving $\arg\max_{\mathcal{P}_\Lambda}\boldsymbol{E}_{\lambda\sim\mathcal{P}_\Lambda}[\mathcal{L}_\mathcal{E}(\lambda,\theta^{(t)}_\lambda)] = \arg\max_{\mathcal{P}_\Lambda}\boldsymbol{E}_{\lambda\sim\mathcal{P}_\Lambda}[r_\lambda]$, the data distribution optimization operator $\mathcal{E}$ is equivalent to solving a multi-arm bandit (MAB) problem. For the first order optimization, (Schulman, Chen, and Abbeel 2017) shows that the solution of a KL-regularized version, $\arg\max_{\mathcal{P}_\Lambda}\boldsymbol{E}_{\lambda\sim\mathcal{P}_\Lambda}[r_\lambda] - \eta KL(\mathcal{P}_\Lambda||\mathcal{P}^{(t)}_\Lambda)$, is exactly the assumption. For the second order optimization, let $\mathcal{P}_\Lambda = softmax(\{r_\lambda\})$, (Agarwal et al. 2019) shows that the natural policy gradient of a softmax parameterization also induces exactly the assumption.*

Assumption 3 (First Order Optimization Co-Monotonic Assumption). *For $\forall\lambda_1,\lambda_2\in\Lambda$, we have $[\mathcal{L}_\mathcal{E}(\lambda_1,\theta_{\lambda_1}) - \mathcal{L}_\mathcal{E}(\lambda_2,\theta_{\lambda_2})]\cdot[\mathcal{L}_\mathcal{T}(\lambda_1,\theta_{\lambda_1}) - \mathcal{L}_\mathcal{T}(\lambda_2,\theta_{\lambda_2})] \geq 0$.*

Assumption 4 (Second Order Optimization Co-Monotonic Assumption). *For $\forall\lambda_1,\lambda_2\in\Lambda$, $\exists\eta_0 > 0$, s.t. $\forall 0 < \eta < \eta_0$, we have $[\mathcal{L}_\mathcal{E}(\lambda_1,\theta_{\lambda_1}) - \mathcal{L}_\mathcal{E}(\lambda_2,\theta_{\lambda_2})]\cdot[G^\eta\mathcal{L}_\mathcal{T}(\lambda_1,\theta_{\lambda_1}) - G^\eta\mathcal{L}_\mathcal{T}(\lambda_2,\theta_{\lambda_2})] \geq 0$, where $\theta^\eta_\lambda = \theta_\lambda + \eta\nabla_{\theta_\lambda}\mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda)$ and $G^\eta\mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda) = \frac{1}{\eta}[\mathcal{L}_\mathcal{T}(\lambda,\theta^\eta_\lambda) - \mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda)]$.*

Under Assumption (1) (2) (3), if $\mathcal{T}$ is a first order operator, namely a gradient accent operator, to maximize $\mathcal{L}_\mathcal{T}$, GDI can be guaranteed to be superior to that w/o $\mathcal{E}$. Under Assumption (1) (2) (4), if $\mathcal{T}$ is a second order operator, namely a natural gradient operator, to maximize $\mathcal{L}_\mathcal{T}$, GDI can also be guaranteed to be superior to that w/o $\mathcal{E}$.

Theorem 1 (Upper Triangular Transport Inequality for Co–Monotonic Functions in $\mathbf{R}^p$). *Assume μ is a continuous probability measure supported on $[0,1]^p$. Denote $\boldsymbol{x} \overset{def}{=} (x^1,\ldots,x^p)$. Let $f,g:[0,1]^p\to\boldsymbol{R}$ to be two co-monotonic functions that satisfy*

$$(f(\boldsymbol{x}) - f(\boldsymbol{y}))\cdot(g(\boldsymbol{x}) - g(\boldsymbol{y})) \geq 0,\ \forall\boldsymbol{x},\boldsymbol{y}\in[0,1]^p.$$

f is continuous. Define

$$\beta(\boldsymbol{x}) = \mu(\boldsymbol{x})\exp(g(\boldsymbol{x}))/Z,\ Z = \int_{[0,1]^p}\mu(\boldsymbol{x})\exp(g(\boldsymbol{x})).$$

Let $f,g:[0,1]^p\to\boldsymbol{R}$ to be two co-monotonic functions that satisfy

$$(f(\boldsymbol{x}) - f(\boldsymbol{y}))\cdot(g(\boldsymbol{x}) - g(\boldsymbol{y})) \geq 0,\ \forall\boldsymbol{x},\boldsymbol{y}\in[0,1]^p.$$

Then we have

$$\boldsymbol{E}_\mu[f] \leq \boldsymbol{E}_\beta[f].$$

Theorem 2 (First Order Optimization with Superior Target). *Under Assumption (1) (2) (3), we have $\mathcal{L}_\mathcal{T}(\mathcal{P}^{(t+1)}_\Lambda,\theta^{(t+1)}) = \boldsymbol{E}_{\lambda\sim\mathcal{P}^{(t+1)}_\Lambda}[\mathcal{L}_\mathcal{T}(\lambda,\theta^{(t+1)}_\lambda)] \geq \boldsymbol{E}_{\lambda\sim\mathcal{P}^{(t)}_\Lambda}[\mathcal{L}_\mathcal{T}(\lambda,\theta^{(t+1)}_\lambda)] = \mathcal{L}_\mathcal{T}(\mathcal{P}^{(t)}_\Lambda,\theta^{(t+1)})$.*

Proof. *By **Theorem 1** (see the appendix of (Fan, Xiao, and Huang 2021)), the upper triangular transport inequality, let $f(\lambda) = \mathcal{L}_\mathcal{T}(\lambda,\theta_\lambda)$ and $g(\lambda) = \mathcal{L}_\mathcal{E}(\lambda,\theta_\lambda)$, the proof is done.*

Remark (Why Superior Target). *In Algorithm 1, if $\mathcal{E}$ updates $\mathcal{P}^{(t)}_\Lambda$ at time t, then the operator $\mathcal{T}$ at time $t+1$ can be written as $\theta^{(t+2)} = \theta^{(t+1)} + \eta\nabla_{\theta^{(t+1)}}\mathcal{L}_\mathcal{T}(\mathcal{P}^{(t+1)}_\Lambda,\theta^{(t+1)})$. If $\mathcal{P}^{(t)}_\Lambda$ hasn't been updated at time t, then the operator*

*$\mathcal{T}$ at time $t+1$ can be written as $\theta^{(t+2)} = \theta^{(t+1)} + \eta\nabla_{\theta^{(t+1)}}\mathcal{L}_{\mathcal{T}}(\mathcal{P}_{\Lambda}^{(t)}, \theta^{(t+1)})$. **Theorem** 2 shows that the target of $\mathcal{T}$ at time $t+1$ becomes higher if $\mathcal{P}_{\Lambda}^{(t)}$ is updated by $\mathcal{E}$ at time t.*

Remark (Practical Implementation)**.** *We provide one possible practical setting of GDI. Let $\mathcal{L}_{\mathcal{E}}(\lambda, \theta_\lambda) = \mathcal{J}_{\pi_{\theta_\lambda}}$ and $\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda) = \mathcal{J}_{\pi_{\theta_\lambda}}$. $\mathcal{E}$ can update $\mathcal{P}_\Lambda$ by the Monte-Carlo estimation of $\mathcal{J}_{\pi_{\theta_\lambda}}$. $\mathcal{T}$ is to maximize $\mathcal{J}_{\pi_{\theta_\lambda}}$, which can be any RL algorithms.*

Theorem 3 (Second Order Optimization with Superior Improvement)**.** *Under Assumption (1) (2) (4), we have $\boldsymbol{E}_{\lambda\sim\mathcal{P}_{\Lambda}^{(t+1)}}[G^\eta\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1)})] \geq \boldsymbol{E}_{\lambda\sim\mathcal{P}_{\Lambda}^{(t)}}[G^\eta\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1)})]$, more specifically, $\boldsymbol{E}_{\lambda\sim\mathcal{P}_{\Lambda}^{(t+1)}}[\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1),\eta}) - \mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1)})] \geq \boldsymbol{E}_{\lambda\sim\mathcal{P}_{\Lambda}^{(t)}}[\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1),\eta}) - \mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda^{(t+1)})]$.*

Proof. *By **Theorem** 1 (see the appendix of (Fan, Xiao, and Huang 2021)), the upper triangular transport inequality, let $f(\lambda) = G^\eta\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda)$ and $g(\lambda) = \mathcal{L}_{\mathcal{E}}(\lambda, \theta_\lambda)$, the proof is done.*

Remark (Why Superior Improvement)**.** ***Theorem** 3 shows that, if $\mathcal{P}_\Lambda$ is updated by $\mathcal{E}$, the expected improvement of $\mathcal{T}$ is higher.*

Lemma 1 (Performance Difference Lemma)**.** *For any policies π, π' and any state s_0, we have*

$$V^\pi(s_0) - V^{\pi'}(s_0) = \frac{1}{1-\gamma}\boldsymbol{E}_{s\sim d_{s_0}^\pi}\boldsymbol{E}_{a\sim\pi(\cdot|s)}\left[A^{\pi'}(s,a)\right].$$

Remark (Practical Implementation). *Let $\mathcal{L}_{\mathcal{E}}(\lambda, \theta_\lambda) = \boldsymbol{E}_{s\sim d_{\rho_0}^\pi}\boldsymbol{E}_{a\sim\pi(\cdot|s)\exp(\epsilon A^\pi(s,\cdot))/Z}[A^\pi(s,a)]$, where $\pi = \pi_{\theta_\lambda}$. Let $\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda) = \mathcal{J}_{\pi_{\theta_\lambda}}$. If we optimize $\mathcal{L}_{\mathcal{T}}(\lambda, \theta_\lambda)$ by natural gradient, (Agarwal et al. 2019) shows that, for direct parameterization, the natural policy gradient gives $\pi^{(t+1)} \propto \pi^{(t)}\exp(\epsilon A^{\pi^{(t)}})$, by **Lemma** 1 (see the appendix of (Fan, Xiao, and Huang 2021)), the performance difference lemma, $V^\pi(s_0) - V^{\pi'}(s_0) = \frac{1}{1-\gamma}\boldsymbol{E}_{s\sim d_{s_0}^\pi}\boldsymbol{E}_{a\sim\pi(\cdot|s)}[A^{\pi'}(s,a)]$, hence if we ignore the gap between the states visitation distributions of $\pi^{(t)}$ and $\pi^{(t+1)}$, $\mathcal{L}_{\mathcal{E}}(\lambda, \theta_\lambda^{(t)}) \approx \frac{1}{1-\gamma}\boldsymbol{E}_{s\sim d_{\rho_0}^\pi}[V^{\pi^{(t+1)}}(s) - V^{\pi^{(t)}}(s)]$, where $\pi^{(t)} = \pi_{\theta_\lambda^{(t)}}$. Hence, $\mathcal{E}$ is actually putting more measure on λ that can achieve more improvement.*

Experiment

We begin this section by describing our experimental setup. Then we report and analyze our SOTA results on ALE, specifically, 57 games, which are summarized and illustrated in the appendix of (Fan, Xiao, and Huang 2021). To further investigate the mechanism of our algorithm, we study the effect of several major components.

Experimental Setup

The overall training architecture is on the top of the Learner-Actor framework (Espeholt et al. 2018), which supports large-scale training. Additionally, the recurrent encoder with LSTM (Schmidhuber 1997) is used to handle the partially observable MDP problem (Bellemare et al. 2013). *burn-in* technique is adopted to deal with the representational drift as (Kapturowski et al. 2018), and we train each sample twice. A complete description of the hyperparameters can be found in the appendix of (Fan, Xiao, and Huang 2021). We employ additional environments to evaluate the scores during training, and the undiscounted episode returns averaged over 32 environments with different seeds have been recorded. Details on ALE and relevant evaluation criteria can be found in the appendix of (Fan, Xiao, and Huang 2021).

To illustrate the generality and efficiency of GDI, we propose one implementation of GDI-I^3 and GDI-H^3, respectively. Let $\Lambda = \{\lambda|\lambda = (\tau_1, \tau_2, \epsilon)\}$. The behavior policy belongs to a soft ϵ-greedy policy space, which contains ϵ-greedy policy and Boltzmann policy. We define the behavior policy π_{θ_λ} as

$$\lambda = (\tau_1, \tau_2, \epsilon),\ \pi_{\theta_\lambda} = \varepsilon\cdot\text{Softmax}\left(\frac{A_1}{\tau_1}\right) + (1-\varepsilon)\cdot\text{Softmax}\left(\frac{A_2}{\tau_2}\right) \tag{2}$$

For GDI-I^3, A_1 and A_2 are identical, so it is estimated by an isomorphic family of trainable variables. The learning policy is also π_{θ_λ}. For GDI-H^3, A_1 and A_2 are different, and they are estimated by two different families of trainable variables. Since GDI needn't assume A_1 and A_2 are learned from the same MDP, so we use two kinds of reward shaping to learn A_1 and A_2 respectively, which can be found in the appendix of (Fan, Xiao, and Huang 2021). Full algorithm can be found in the appendix of (Fan, Xiao, and Huang 2021).

The operator $\mathcal{T}$ is achieved by policy gradient, V-Trace and ReTrace (Espeholt et al. 2018; Munos et al. 2016), which meets Theorem 2 by first order optimization.

The operator $\mathcal{E}$, which optimizes $\mathcal{P}_\Lambda$, is achieved by a variant of Multi-Arm Bandits (Sutton and Barto 2018, MAB), where Assumption 2 holds naturally. More details can be found in the appendix of (Fan, Xiao, and Huang 2021).

Summary of Results

We construct a multivariate evaluation system to emphasize the superiority of our algorithm in all aspects, and more discussions on those evaluation criteria are in the appendix of (Fan, Xiao, and Huang 2021) and details are in the appendix of (Fan, Xiao, and Huang 2021). Furthermore, to avoid any issues that aggregated metrics may have, the appendix of (Fan, Xiao, and Huang 2021) provides full learning curves for all games, as well as detailed comparison tables of raw and normalized scores.

The aggregated results across games are reported in Tab. 1. Our agents obtain the highest mean HNS with an extraordinary learning efficiency from this table. Furthermore, our agents have achieved 22 human world record breakthroughs and more than 90 times the average human score of Atari games via playing from scratch for less than 1.5 months. Although Agent57 obtains the highest median HNS, it costs each of the agents more than 57 years to obtain such performance, revealing its low learning efficiency. It is obvious that there is no such world record achieved by a human who played for over 57 years. This is due to the fact that Agent57 fails to handle the balance between exploration and

	GDI-H^3	GDI-I^3	Muesli	RAINBOW	LASER	R2D2	NGU	Agent57
Num. Frames	**2E+8**	**2E+8**	**2E+8**	**2E+8**	**2E+8**	1E+10	3.5E+10	1E+11
Game Time (year)	**0.114**	**0.114**	**0.114**	**0.114**	**0.114**	5.7	19.9	57
HWRB	**22**	17	5	4	7	15	8	18
Mean HNS(%)	**9620.98**	7810.6	2538.66	873.97	1741.36	3374.31	3169.90	4763.69
Median HNS(%)	1146.39	832.5	1077.47	230.99	454.91	1342.27	1208.11	**1933.49**
Mean HWRNS(%)	**154.27**	117.99	75.52	28.39	45.39	98.78	76.00	125.92
Median HWRNS(%)	**50.63**	35.78	24.86	4.92	8.08	33.62	21.19	43.62
Mean SABER(%)	71.26	61.66	48.74	28.39	36.78	60.43	50.47	**76.26**
Median SABER(%)	**50.63**	35.78	24.68	4.92	8.08	33.62	21.19	43.62

Table 1: Experiment results of Atari. Muesli's scores are from (Hessel et al. 2021). RAINBOW's scores are from (Espeholt et al. 2018). LASER's scores are from (Schmitt, Hessel, and Simonyan 2020), no sweep at 200M. R2D2's scores are from (Kapturowski et al. 2018). NGU's scores are from (Badia et al. 2020b). Agent57's scores are from (Badia et al. 2020a). Full comparison among all algorithms can see the appendix of (Fan, Xiao, and Huang 2021).

exploitation, thus collecting a large number of inferior samples, which further hinders the efficient-learning and makes it harder for policy improvement. Other algorithms gain higher learning efficiency than Agent57 but relatively lower final performance, such as NGU and R2D2, which acquire over 10B frames. Except for median HNS, our performance is better on all criteria than NGU and R2D2. In addition, other algorithms with 200M training frames are struggling to match our performance.

These results come from the following aspects:

1. Several games have been solved completely, achieving the historically highest score, such as RoadRunner, Seaquest, Jamesbond.
2. Massive games show enormous potentialities for improvement but fail to converge for lack of training, such as BeamRider, BattleZone, SpaceInvaders.
3. This paper aims to illustrate that GDI is general for seeking a suitable balance between exploration and exploitation, so we refuse to adopt any handcrafted and domain-specific tricks such as the intrinsic reward. Therefore, we suffer from the hard exploration problem, such as PrivateEye, Surround, Amidar.

Therefore, there are several aspects of potential improvement. For example, a more extensive training scale may benefit higher performance. More exploration techniques can be incorporated into GDI to handle those hard-exploration problems through guiding the direction of the acquired samples.

Ablation Study

In the ablation study, we further investigate the effects of several properties of GDI. We set GDI-I^3 and GDI-H^3 as our baseline control group. To prove the effects of the data distribution optimization operator $\mathcal{E}$, we set two ablation groups, which are Fixed Selection from GDI-I^0 w/o $\mathcal{E}$ and Random Selection from GDI-I^3 w/o $\mathcal{E}$. To prove the capacity of the behavior policy space matters in GDI, we set two ablation groups, which are ϵ-greedy Selection $\Lambda = \{\lambda|\lambda = (\epsilon)\}$ and Boltzmann Selection $\Lambda = \{\lambda|\lambda = (\tau)\}$. Both ϵ-greedy Selection and Boltzmann Selection implement $\mathcal{E}$ by the same MAB as our baselines'. More details on ablation study can see the appendix of (Fan, Xiao, and Huang 2021).

From results in the appendix of (Fan, Xiao, and Huang 2021), it is evident that both the data distribution optimization operator $\mathcal{E}$ and the capacity of the behavior policy space are critical. This is since if they lack the cognition to identify suitable experiences from various data, high variance and massive poor experiences will hinder the policy improvement, and if the RL agents lack the vision to find more examples to learn, they may ignore some shortcuts. To further prove the capacity of the policy space does bring more diverse data, we draw the t-SNE of GDI-I^3, GDI-H^3 and Boltzmann Selection in the appendix of (Fan, Xiao, and Huang 2021), from which we see GDI-I^3 and GDI-H^3 can explore more high-value states that Boltzmann selection has less chance to find. We also evaluate Fixed Selection and Boltzmann Selection in all 57 Atari games, and recorded the comparison tables of raw and normalized scores in the appendix of (Fan, Xiao, and Huang 2021).

Conclusion

This paper proposes a novel RL paradigm to effectively and adaptively trade-off the exploration and exploitation, integrating the data distribution optimization into the generalized policy iteration paradigm. Under this paradigm, we propose feasible implementations, which both have achieved new SOTA among all 200M scale algorithms on all evaluation criteria and obtained the best mean final performance and learning efficiency compared with all 10B+ scale algorithms. Furthermore, we have achieved 22 human world record breakthroughs within less than 1.5 months of game time. It implies that our algorithm obtains both superhuman learning performance and human-level learning efficiency. In the experiment, we discuss the potential improvement of our method in future work.

References

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2019. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *arXiv preprint arXiv:1908.00261.*

Badia, A. P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; and Blundell, C. 2020a. Agent57:

Outperforming the atari human benchmark. *arXiv preprint arXiv:2003.13350*.

Badia, A. P.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; Piot, B.; Kapturowski, S.; Tieleman, O.; Arjovsky, M.; Pritzel, A.; Bolt, A.; et al. 2020b. Never Give Up: Learning Directed Exploration Strategies. *arXiv preprint arXiv:2002.06038*.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4: 129–145.

Dadashi, R.; Taiga, A. A.; Le Roux, N.; Schuurmans, D.; and Bellemare, M. G. 2019. The value function polytope in reinforcement learning. In *International Conference on Machine Learning*, 1486–1495. PMLR.

Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2018. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Fan, J.; Xiao, C.; and Huang, Y. 2021. GDI: Rethinking What Makes Reinforcement Learning Different From Supervised Learning. *arXiv preprint arXiv:2106.06232*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Hessel, M.; Danihelka, I.; Viola, F.; Guez, A.; Schmitt, S.; Sifre, L.; Weber, T.; Silver, D.; and van Hasselt, H. 2021. Muesli: Combining Improvements in Policy Optimization. *arXiv preprint arXiv:2104.06159*.

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

Horgan, D.; Quan, J.; Budden, D.; Barth-Maron, G.; Hessel, M.; van Hasselt, H.; and Silver, D. 2018. Distributed Prioritized Experience Replay. In *International Conference on Learning Representations*.

Howard, R. A. 1960. *Dynamic programming and markov processes*. John Wiley.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W. M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.

Kapturowski, S.; Ostrovski, G.; Quan, J.; Munos, R.; and Dabney, W. 2018. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*.

Kumar, A.; Gupta, A.; and Levine, S. 2020. Discor: Corrective feedback in reinforcement learning via distribution correction. *arXiv preprint arXiv:2003.07305*.

LaBar, T.; and Adami, C. 2017. Evolution of drift robustness in small populations. *Nature Communications*, 8(1): 1–12.

Mitchell, T. M.; et al. 1997. *Machine learning*. McGraw-hill New York.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Munos, R.; Stepleton, T.; Harutyunyan, A.; and Bellemare, M. 2016. Safe and Efficient Off-Policy Reinforcement Learning. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 1054–1062. Curran Associates, Inc.

Parker-Holder, J.; Pacchiano, A.; Choromanski, K.; and Roberts, S. 2020. Effective diversity in population-based reinforcement learning. *arXiv preprint arXiv:2002.00632*.

Pennisi, E. 2016. Tracking how humans evolve in real time. *Science*, 352(6288): 876–877.

Schmidhuber, S. H. J. 1997. Long short-term memory. *Neural Computation*.

Schmitt, S.; Hessel, M.; and Simonyan, K. 2020. Off-policy actor-critic with shared experience replay. In *International Conference on Machine Learning*, 8545–8554. PMLR.

Schulman, J.; Chen, X.; and Abbeel, P. 2017. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Toromanoff, M.; Wirbel, E.; and Moutarde, F. 2019. Is deep reinforcement learning really superhuman on atari? leveling the playing field. *arXiv preprint arXiv:1908.04683*.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; and Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, 1995–2003.

Watkins, C. J. C. H. 1989. *Learning from delayed rewards*. King's College, Cambridge United Kingdom.

Wiering, M. A. 1999. *Explorations in efficient reinforcement learning*. Ph.D. thesis, University of Amsterdam.