LEARNABLE BEHAVIOR CONTROL: BREAKING ATARI HUMAN WORLD RECORDS VIA SAMPLE-EFFICIENT BEHAVIOR SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The exploration problem is one of the main challenges in deep reinforcement learning (RL). Recent promising works tried to handle the problem with population-based methods, which collect samples with diverse behaviors derived from a population of different exploratory policies. Adaptive policy selection has been adopted for behavior control. However, the behavior selection space is largely limited by the predefined policy population, which further limits behavior diversity. In this paper, we propose a general framework called **L**earnable **B**ehavioral **C**ontrol (LBC) to address the limitation, which a) enables a significantly enlarged behavior selection space via formulating a *hybrid behavior mapping* from all policies; b) constructs a unified *learnable process* for behavior selection. We introduce LBC into distributed off-policy actor-critic methods and achieve behavior control via optimizing the selection of the behavior mappings with bandit-based meta-controllers. Our agents have achieved 10077.52% mean human normalized score and surpassed 24 human world records within 1B training frames in the Arcade Learning Environment, which demonstrates our significant state-of-the-art (SOTA) performance without degrading the sample efficiency.

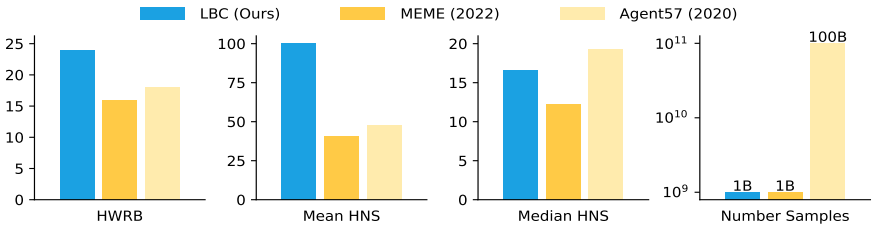


Figure 1: Performance on the 57 Atari games. Our method achieves the highest mean human normalized scores $\text{HNS} = \frac{\text{Agent}_{\text{score}} - \text{Random}_{\text{score}}}{\text{Human}_{\text{score}} - \text{Random}_{\text{score}}}$ (Bellemare et al., 2013), is the first to achieve 24 human world record breakthroughs $\text{HWRB} = \text{Agent}_{\text{score}} \geq \text{Human World Records}_{\text{score}}$ (Toromanoff et al., 2019), and demands the least training data.

1 INTRODUCTION

Reinforcement learning (RL) has led to tremendous progress in a variety of domains ranging from video games (Mnih et al., 2015) to robotics (Schulman et al., 2015; 2017). However, efficient exploration remains one of the significant challenges. Recent prominent works tried to address the problem with population-based training (Jaderberg et al., 2017, PBT) wherein a population of policies with different degrees of exploration is jointly trained to keep both the long-term and short-term exploration capabilities throughout the learning process. A set of actors is created to acquire diverse behaviors derived from the policy population (Badia et al., 2020b;a). Despite the significant improvement in the performance, these methods suffer from the aggravated high sample complexity due to the joint training on the whole population while keeping the diversity property. To acquire diverse behaviors, NGU (Badia et al., 2020b) uniformly selects policies in the population regardless of their contribution to the learning progress (Badia et al., 2020b). As an improvement, Agent57 adopts an adaptive policy selection mechanism that each behavior used for sampling is periodically

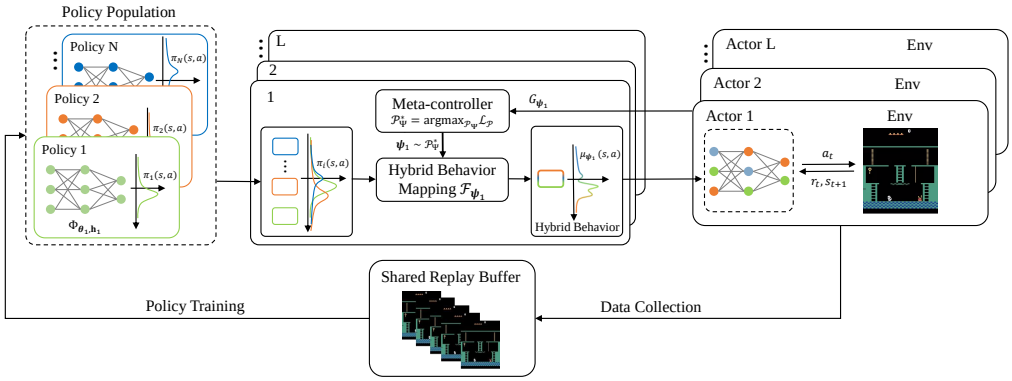


Figure 2: A General Architecture of Our Algorithm.

selected from the population according to a meta-controller (Badia et al., 2020a). Although Agent57 achieved significantly better results on the Arcade Learning Environment (ALE) benchmark, it costs tens of billions of environment interactions as much as NGU. To handle this drawback, GDI (Fan & Xiao, 2022) adaptively combines multiple advantage functions learned from each policy to obtain an enlarged behavior space without increasing policy population size. However, the behavior space may degenerate as the difference among the advantage functions learned from the same policy would gradually decrease (See App. N).

In this paper, we attempt to further improve the sample efficiency of population-based reinforcement learning methods by taking a step towards a more challenging setting to control behaviors with significantly enlarged behavior space with a population of different learned policies. Differing from all of the existing works where each behavior is derived from a single selected learned policy, we formulate the process of getting behaviors from all learned policies as *hybrid behavior mapping*, and the behavior control problem is directly transformed into selecting appropriate mapping functions. By combining all policies, the behavior selection space increases exponentially along with the population size. As a special case that *population size degrades to one*, diverse behaviors can also be obtained by choosing different behavior mappings. This two-fold mechanism enables tremendous larger space for behavior selection. By properly parameterizing the mapping functions, our method enables a unified learnable process, and we call this general framework **L**earnable **B**ehavior **C**ontrol.

We use the Arcade Learning Environment (ALE) to evaluate the performance of the proposed methods, which is an important testing ground that requires a broad set of skills such as perception, exploration, and control (Badia et al., 2020a). Previous works use the normalized human score to summarize the performance on ALE and claim superhuman performance (Bellemare et al., 2013). However, the human baseline is far from representative of the best human player, which greatly underestimates the ability of humanity. In this paper, we introduce a more challenging baseline, *i.e.*, the human world records baseline (see Toromanoff et al. (2019); Hafner et al. (2021) for more information on Atari human world records). We summarize the number of games that agents can outperform the human world records (*i.e.*, HWRB, see Figs. 1) to claim a real superhuman performance in these games, inducing a more challenging and fair comparison with human intelligence. Experimental results show that the sample efficiency of our method also outperforms the concurrent work MEME Kapturowski et al. (2022), which is 200x faster than Agent57. In summary, our contributions are as follows:

1. **A data-efficient RL framework named LBC**. We propose a general framework called **L**earnable **B**ehavior **C**ontrol (LBC), which enables a significantly enlarged behavior selection space without increasing the policy population size via formulating a hybrid behavior mapping from all policies, and constructs a unified learnable process for behavior selection.
2. **A family of LBC-based RL algorithms**. We provide a family of LBC-based algorithms by combining LBC with existing distributed off-policy RL algorithms, which shows the generality and scalability of the proposed method.
3. **The state-of-the-art performance with superior sample efficiency**. From Figs. 1, our method has achieved 10077.52% mean human normalized score (HNS) and surpassed 24 human world records within 1B training frames in the Arcade Learning Environment (ALE), which demonstrates our state-of-the-art (SOTA) sample efficiency.

2 BACKGROUND

2.1 REINFORCEMENT LEARNING

RL 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 $\rho_0(s) : \mathcal{S} \to \mathbb{P}(\mathcal{S})$, where we use $\mathbb{P}$ to represent the probability distribution. At each time t, the agent chooses an action $a_t \in \mathcal{A}$ according to the policy $\pi(a_t|s_t) : \mathcal{S} \to \mathbb{P}(\mathcal{A})$ at state $s_t \in \mathcal{S}$. The environment receives a_t, produces the reward $r_t \sim r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbf{R}$ and transfers to the next state s_{t+1} according to the transition distribution $p(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{P}(\mathcal{S})$. The process continues until the agent reaches a terminal state or a maximum time step. Define the discounted state visitation distribution as $d^{\pi}_{\rho_0}(s) = (1-\gamma)\mathbb{E}_{s_0 \sim \rho_0}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{P}(s_t = s|s_0)\right]$. Define return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ wherein $\gamma \in (0, 1)$ is the discount factor. The goal of reinforcement learning is to find the optimal policy π^* that maximizes the expected sum of discounted rewards G_t:

$$\pi^* := \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{s_t \sim d^{\pi}_{\rho_0}} \mathbb{E}_{\pi}\left[G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t\right], \tag{1}$$

2.2 BEHAVIOR CONTROL FOR REINFORCEMENT LEARNING

In value-based methods, a behavior policy can be derived from a state-action value function $Q^{\pi}_{\boldsymbol{\theta},\mathbf{h}}(s, a)$ via ϵ-greedy. In policy-based methods, a behavior policy can be derived from the policy logits $\Phi_{\boldsymbol{\theta},\mathbf{h}}$ (Li et al., 2018) via Boltzmann operator. For convenience, we define that a behavior policy can be derived from the learned policy model $\Phi_{\boldsymbol{\theta},\mathbf{h}}$ via a behavior mapping, which normally maps a single policy model to a behavior, *e.g.*, ϵ-greedy$(\Phi_{\boldsymbol{\theta},\mathbf{h}})$. In PBT-based methods, there would be a set of policy models $\{\Phi_{\boldsymbol{\theta}_1,\mathbf{h}_1}, ..., \Phi_{\boldsymbol{\theta}_{\mathrm{N}},\mathbf{h}_{\mathrm{N}}}\}$, each of which is parameterized by $\boldsymbol{\theta}_i$ and trained under its own hyper-parameters $\mathbf{h}_i$, wherein $\boldsymbol{\theta}_i \in \boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{\mathrm{N}}\}$ and $\mathbf{h}_i \in \mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_{\mathrm{N}}\}$.

The behavior control in population-based methods is normally achieved in two steps: i) select a policy model $\Phi_{\boldsymbol{\theta},\mathbf{h}}$ from the population. ii) applying a behavior mapping to the selected policy model. When the behavior mapping is rule-based for each actor (*e.g.*, ϵ-greedy with rule-based ϵ), the behavior control can be transformed into the policy model selection (See Proposition 1). Therefore, the optimization of the selection of the policy models becomes one of the critical problems in achieving effective behavior control. Following the literature on PBRL, NGU adopts a uniform selection, which is unoptimized and inefficient. Built upon NGU, Agent57 adopts a meta-controller to adaptively selected a policy model from the population to generate the behavior for each actor, which is implemented by a non-stationary multi-arm bandit algorithm. However, the policy model selection requires maintaining a large number of different policy models, which is particularly data-consuming since each policy model in the population holds heterogeneous training objectives.

To handle this problem, recent notable work GDI-H^3 (Fan & Xiao, 2022) enables to obtain an enlarged behavior space via adaptively controls the temperature of the softmax operation over the weighted advantage functions. However, since the advantage functions are derived from the same target policy under different reward scales, the distributions derived from them may tend to be similar (e.g., See App. N), thus would lead to degradation of the behavior space. Differing from all of the existing works where each behavior is derived from a single selected learned policy, in this paper, we try to handle this problem via three-fold: i) we bridge the relationship between the learned policies and each behavior via a hybrid behavior mapping, ii) we propose a general way to build a non-degenerate large behavior space for population-based methods in Sec. 4.1, iii) we propose a way to optimize the hybrid behavior mappings from a population of different learned models in Proposition. 2.

3 LEARNABLE BEHAVIOR CONTROL

In this section, we first formulate the behavior control problem and decouple it into two sub-problems: behavior space construction and behavior selection. Then, we discuss how to construct the behavior space and select behaviors based on the formulation. By integrating both, we can obtain a general framework to achieve behavior control in RL, called learnable behavior control (LBC).

3.1 BEHAVIOR CONTROL FORMULATION

Behavior Mapping Define *behavior mapping* $\mathcal{F}$ as a mapping from some policy model(s) to a behavior. In previous works, a behavior policy is typically obtained using a single policy model. In this paper, as a generalization, we define two kinds of $\mathcal{F}$ according to how many policy models they take as input to get a behavior. The first one, *individual behavior mapping*, is defined as a mapping from a single model to a behavior that is widely used in prior works, *e.g.*, ϵ-greedy and Boltzmann Strategy for discrete action space and Gaussian Strategy for continuous action space; And the second one, *hybrid behavior mapping*, is defined to map all policy models to a single behavior, *i.e.*, $\mathcal{F}(\Phi_{\boldsymbol{\theta}_1,\mathbf{h}_1}, ..., \Phi_{\boldsymbol{\theta}_\mathrm{N},\mathbf{h}_\mathrm{N}})$.[1] The hybrid behavior mapping enables us to get a hybrid behavior by combining all policies together, which provides a greater degree of freedom to acquire a larger behavior space. For any behavior mapping $\mathcal{F}_{\boldsymbol{\psi}}$ parameterized by $\boldsymbol{\psi}$, there exists a family of behavior mappings $\mathcal{F}_{\boldsymbol{\Psi}} = \{\mathcal{F}_{\boldsymbol{\psi}} | \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$ that hold the same parametrization form with $\mathcal{F}_{\boldsymbol{\psi}}$, where $\boldsymbol{\Psi} \subseteq \mathbf{R}^k$ is a parameter set that contains all possible parameter $\boldsymbol{\psi}$.

Behavior Formulation As described above, in our work, a behavior can be acquired by applying a behavior mapping $\mathcal{F}_{\boldsymbol{\psi}}$ to some policy model(s). For the individual behavior mapping case, a behavior can be formulated as $\mu_{\boldsymbol{\theta},\mathbf{h},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\boldsymbol{\Phi}_{\boldsymbol{\theta},\mathbf{h}})$, which is also the most used case in previous works. As for the hybrid behavior mapping case, a behavior is formulated as $\mu_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\boldsymbol{\Phi}_{\boldsymbol{\Theta},\mathbf{H}})$, wherein $\boldsymbol{\Phi}_{\boldsymbol{\Theta},\mathbf{H}} = \{\Phi_{\boldsymbol{\theta}_1,\mathbf{h}_1}, ..., \Phi_{\boldsymbol{\theta}_\mathrm{N},\mathbf{h}_\mathrm{N}}\}$ is a policy model set containing all policy models.

Behavior Control Formulation Behavior control can be decoupled into two sub-problems: 1) which behaviors can be selected for each actor at each training time, namely the *behavior space construction*. 2) how to select proper behaviors, namely the *behavior selection*. Based on the behavior formulation, we can formulate these sub-problems:

Definition 3.1 (Behavior Space Construction)**.** *Considering the RL problem that behaviors μ are generated from some policy model(s). We can acquire a family of realizable behaviors by applying a family of behavior mappings $\mathcal{F}_{\boldsymbol{\Psi}}$ to these policy model(s). Define the set that contains all of these realizable behaviors as the behavior space, which can be formulated as:*

$$\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}} = \begin{cases} \{\mu_{\boldsymbol{\theta},\mathbf{h},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\mathbf{h}}) | \boldsymbol{\theta} \in \boldsymbol{\Theta}, \mathbf{h} \in \mathbf{H}, \boldsymbol{\psi} \in \boldsymbol{\Psi}\}, & \textit{for individual behavior mapping} \\ \{\mu_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\boldsymbol{\Phi}_{\boldsymbol{\Theta},\mathbf{H}}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\}, & \textit{for hybrid behavior mapping} \end{cases} \tag{2}$$

Definition 3.2 (Behavior Selection)**.** *Behavior selection can be formulated as finding a optimal selection distribution $\mathcal{P}^*_{\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}}$ to select the behaviors μ from behavior space $\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}$ and maximizing some optimization target $\mathcal{L}_{\mathcal{P}}$, wherein $\mathcal{L}_{\mathcal{P}}$ is the optimization target of behavior selection:*

$$\mathcal{P}^*_{\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}} := \underset{\mathcal{P}_{\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}}}{\operatorname{argmax}} \mathcal{L}_{\mathcal{P}} \tag{3}$$

3.2 BEHAVIOR SPACE CONSTRUCTION

In this section, we further simplify the equation 2, and discuss how to construct the behavior space.

Assumption 1. *Assume all policy models share the same network structure, and $\mathbf{h}_i$ can uniquely index a policy model $\Phi_{\boldsymbol{\theta}_i,\mathbf{h}_i}$. Then, $\Phi_{\boldsymbol{\theta},\mathbf{h}}$ can be abbreviated as $\Phi_{\mathbf{h}}$.*

Unless otherwise specified, in this paper, we assume Assumption 1 holds. Under Assumption 1, the behavior space defined in equation 2 can be simplified as,

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \begin{cases} \{\mu_{\mathbf{h},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\mathbf{h}}) | \mathbf{h} \in \mathbf{H}, \boldsymbol{\psi} \in \boldsymbol{\Psi}\}, & \text{for individual behavior mapping} \\ \{\mu_{\mathbf{H},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\boldsymbol{\Phi}_{\mathbf{H}}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\}, & \text{for hybrid behavior mapping} \end{cases} \tag{4}$$

According to equation 4, four core factors need to be considered when constructing a behavior space: the network structure Φ, the form of behavior mapping $\mathcal{F}$, the hyper-parameter set $\mathbf{H}$ and the parameter set $\boldsymbol{\Psi}$. Many notable representation learning approaches have explored how to design the network structure (Chen et al., 2021; Irie et al., 2021), but it is not the focus of our work. In this

[1] An intuitive example for the hybrid behavior mapping is the mixture model Lindsay (1995).

paper, we do not make any assumptions about the model structure, which means it can be applied to *any* model structure. Hence, there remains three factors, which will be discussed below.

For cases that behavior space is constructed with *individual behavior mappings*, there are two things to be considered if one want to select a specific behavior from the behavior space: the policy model $\mathbf{\Phi_h}$ and behavior mapping $\mathcal{F}_{\boldsymbol{\psi}}$. Prior methods have tried to realize behavior control via selecting a policy model $\mathbf{\Phi}_{\mathbf{h}_i}$ from the population $\{\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}}\}$ (See Proposition 1). The main drawback of this approach is that only one policy model is considered to generate behavior, leaving other policy models in the population unused. In this paper, we argue that we can tackle this problem via *hybrid behavior mapping*, wherein the hybrid behavior is generated based on all policy models.

In this paper, we only consider the case that all of the N policy models are used for behavior generating, *i.e.*, $\mu_{\mathbf{H},\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\mathbf{\Phi_H})$. Now there is only one thing to be considered , *i.e.*, the behavior mapping function $\mathcal{F}_{\boldsymbol{\psi}}$, and the behavior control problem will be transformed into the optimization of the behavior mapping (See Proposition 2). We also do not make any assumptions about the form of the mapping. As an example, one could acquire a hybrid behavior from all policy models via network distillation, parameter fusion, mixture models, etc.

3.3 Behavior Selection

According to equation 4, each behavior can be indexed by $\mathbf{h}$ and $\boldsymbol{\psi}$ for individual behavior mapping cases, and when the $\boldsymbol{\psi}$ is not learned for each actor, the behavior selection can be cast to the selection of $\mathbf{h}$ (see Proposition 1). As for the hybrid behavior mapping cases, since each behavior can be indexed by $\boldsymbol{\psi}$, the behavior selection can be cast into the selection of $\boldsymbol{\psi}$ (see Proposition 2). Moreover, according to equation 3, there are two keys in behavior selection: **1)** Optimization Target $\mathcal{L}_{\mathcal{P}}$. **2)** The optimization algorithm to learn the selection distribution $\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}$ and maximize $\mathcal{L}_{\mathcal{P}}$. In this section, we will discuss them sequentially.

Optimization Target Two core factors have to be considered for the optimization target: the diversity-based measurement V_μ^{TD} (Eysenbach et al., 2019) and the value-based measurement V_μ^{TV} (Parker-Holder et al., 2020). By integrating both, the optimization target can be formulated as:

$$\begin{aligned} \mathcal{L}_{\mathcal{P}} &= \mathcal{R}_{\mu\sim\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}} + c\cdot\mathcal{D}_{\mu\sim\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}} \\ &= \mathbb{E}_{\mu\sim\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}}[V_\mu^{\mathrm{TV}} + c\cdot V_\mu^{\mathrm{TD}}], \end{aligned} \tag{5}$$

wherein, $\mathcal{R}_{\mu\sim\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}}$ and $\mathcal{D}_{\mu\sim\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}}$ is the expectation of value and diversity of behavior μ over the selection distribution $\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}$. When $\mathcal{F}_{\boldsymbol{\psi}}$ is *unlearned* and *deterministic* for each actor, behavior selection for *each actor* can be simplified into the selection of the policy model:

Proposition 1 (Policy Model Selection)**.** *When $\mathcal{F}_{\boldsymbol{\psi}}$ is a deterministic and individual behavior mapping for each actor at each training step (wall-clock), e.g.,* ***Agent57****, the behavior for each actor can be uniquely indexed by* $\mathbf{h}$*, so equation 5 can be simplified into*

$$\mathcal{L}_{\mathcal{P}} = \mathbb{E}_{\mathbf{h}\sim\mathcal{P}_{\mathbf{H}}}\left[V_{\mu_{\mathbf{h}}}^{\mathrm{TV}} + c\cdot V_{\mu_{\mathbf{h}}}^{\mathrm{TD}}\right], \tag{6}$$

where $\mathcal{P}_{\mathbf{H}}$ is a selection distribution of $\mathbf{h}\in\mathbf{H}=\{\mathbf{h}_1,...,\mathbf{h}_\mathrm{N}\}$. For each actor, the behavior is generated from a selected policy model $\Phi_{\mathbf{h}_i}$ with a pre-defined behavior mapping $\mathcal{F}_{\boldsymbol{\psi}}$.

In Proposition 1, the behavior space size is controlled by the policy model population size (*i.e.*, $|\mathbf{H}|$). However, maintaining a large population of different policy models is data-consuming. Hence, we try to control behaviors via optimizing the selection of behavior mappings:

Proposition 2 (Behavior Mapping Optimization)**.** *When all the policy models are used to generate each behavior, e.g., $\mu_{\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\boldsymbol{\theta},\mathbf{h}})$ for single policy model cases or $\mu_{\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\boldsymbol{\theta}_1,\mathbf{h}_1},...,\Phi_{\boldsymbol{\theta}_\mathrm{N},\mathbf{h}_\mathrm{N}})$ for* N *policy models cases, each behavior can be uniquely indexed by $\mathcal{F}_{\boldsymbol{\psi}}$, and equation 5 can be simplified into:*

$$\mathcal{L}_{\mathcal{P}} = \mathbb{E}_{\boldsymbol{\psi}\sim\mathcal{P}_{\mathbf{\Psi}}}\left[V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TV}} + c\cdot V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TD}}\right], \tag{7}$$

where $\mathcal{P}_{\mathbf{\Psi}}$ is a selection distribution of $\boldsymbol{\psi}\in\mathbf{\Psi}$.

In Proposition 2, the behavior space is majorly controlled by $|\Psi|$, which could be a continuous parameter space. Hence, a larger behavior space can be enabled.

Selection Distribution Optimization Given the optimization target $\mathcal{L}_{\mathcal{P}}$, we seek to find the optimal behavior selection distribution $\mathcal{P}^*_{\mu}$ that maximizes $\mathcal{L}_{\mathcal{P}}$:

$$\begin{aligned}\mathcal{P}^*_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}} := \underset{\mathcal{P}_{\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}}}{\operatorname{argmax}} \mathcal{L}_{\mathcal{P}} &\overset{(1)}{=} \underset{\mathcal{P}_{\mathbf{H}}}{\operatorname{argmax}} \mathcal{L}_{\mathcal{P}} \\ &\overset{(2)}{=} \underset{\mathcal{P}_{\mathbf{\Psi}}}{\operatorname{argmax}} \mathcal{L}_{\mathcal{P}},\end{aligned} \tag{8}$$

where (1) and (2) hold because we have Proposition 1 and 2, respectively. This optimization problem can be solved with existing optimizers, *e.g.*, evolutionary algorithm (Jaderberg et al., 2017), multi-arm bandits (MAB) (Badia et al., 2020a), etc.

4 LBC-BM: A BOLTZMANN MIXTURE BASED IMPLEMENTATION FOR LBC

In this section, we provide an example of improving the behavior control of off-policy actor-critic methods (Espeholt et al., 2018) via optimizing the behavior mappings as Proposition 2. We provide a practical design of hybrid behavior mapping, inducing an implementation of LBC, which we call **B**oltzmann **M**ixture based **LBC**, namely LBC-$\mathcal{BM}$. By choosing different $\mathbf{H}$ and $\mathbf{\Psi}$, we can obtain a family of implementations of LBC-$\mathcal{BM}$ with different behavior spaces (see Sec. 5.4).

4.1 BOLTZMANN MIXTURE BASED BEHAVIOR SPACE CONSTRUCTION

In this section, we provide a general hybrid behavior mapping design including three sub-processes:

Generalized Policy Selection In Agent57, behavior control is achieved by selecting a single policy from the policy population at each iteration. Following this idea, we generalize the method to the case where multiple policies can be selected. More specifically, we introduce a importance weights vector $\boldsymbol{\omega}$ to describe how much each policy will contribute to the generated behavior, $\boldsymbol{\omega} = [\omega_1, ..., \omega_{\mathrm{N}}], \omega_i \geq 0, \sum_{i=1}^{\mathrm{N}} \omega_i = 1$, where ω_i represents the importance of ith policy in the population (*i.e.*, $\Phi_{\mathbf{h}_i}$). In particular, if $\boldsymbol{\omega}$ is a one-hot vector, *i.e.*, $\exists i \in \{1, 2, ..., \mathrm{N}\}, \omega_i = 1; \forall j \in \{1, 2, ..., \mathrm{N}\} \neq i, \omega_j = 0$, then the policy selection becomes a single policy selection as Proposition 1. Therefore, it can be seen as a generalization of single policy selection, and we call this process *generalized policy selection*.

Policy-Wise Entropy Control In our work, we propose to use entropy control (which is typically rule-based controlled in previous works) to make a better trade-off between exploration and exploitation. For a policy model $\Phi_{\mathbf{h}_i}$ from the population, we will apply a entropy control function $f_{\tau_i}(\cdot)$, *i.e.*, $\pi_{\mathbf{h}_i,\tau_i} = f_{\tau_i}(\Phi_{\mathbf{h}_i})$, where $\pi_{\mathbf{h}_i,\tau_i}$ is the new policy after entropy control, and $f_{\tau_i}(\cdot)$ is parameterized by τ_i. Here we should note that the entropy of all the policies from the population is controlled in a policy-wise manner. Thus there would be a set of entropy control functions to be considered, which is parameterized by $\boldsymbol{\tau} = [\tau_1, ..., \tau_{\mathrm{N}}]$.

Behavior Distillation from Multiple Policies Different from previous methods where only one policy is used to generate the behavior, in our approach, we combine N policies $[\pi_{\mathbf{h}_1,\tau_1}, ..., \pi_{\mathbf{h}_{\mathrm{N}},\tau_{\mathrm{N}}}]$, together with their importance weights $\boldsymbol{\omega} = [\omega_1, ..., \omega_{\mathrm{N}}]$. Specially, in order to make full use of these policies according to their importance, we introduce a *behavior distillation function* g which takes both the policies and importance weights as input, *i.e.*, $\mu_{\mathbf{H},\boldsymbol{\tau},\boldsymbol{\omega}} = g(\pi_{\mathbf{h}_1,\tau_1}, ..., \pi_{\mathbf{h}_{\mathrm{N}},\tau_{\mathrm{N}}}, \boldsymbol{\omega})$. The distillation function $g(\cdot, \boldsymbol{\omega})$ can be implemented in different ways, *e.g.*, knowledge distillation (supervised learning), parameters fusion, etc.

In conclusion, the behavior space can be constructed as,

$$\mathbf{M}_{\mathbf{H},\mathbf{\Psi}} = \{g\left(f_{\tau_1}(\Phi_{\mathbf{h}_1}), ..., f_{\tau_{\mathrm{N}}}(\Phi_{\mathbf{h}_{\mathrm{N}}}), \omega_1, ..., \omega_{\mathrm{N}}\right) | \boldsymbol{\psi} \in \mathbf{\Psi}\} \tag{9}$$

wherein $\mathbf{\Psi} = \{\boldsymbol{\psi} = (\tau_1, ..., \tau_{\mathrm{N}}, \omega_1, ..., \omega_{\mathrm{N}})\}$, $\mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_{\mathrm{N}}\}$. Note that this is a general approach which can be applied to different tasks and algorithms by simply selecting different entropy control function $f_{\tau_i}(\cdot)$ and behavior distillation function $g(\cdot, \boldsymbol{\omega})$. As an example, for Atari task, we model the policy as a Boltzmann distribution, *i.e.*, $\pi_{\mathbf{h}_i,\tau_i}(a|s) = e^{\tau_i \Phi_{\mathbf{h}_i}(a|s)} \sum_{a'} e^{\tau_i \Phi_{\mathbf{h}_i}(a'|s)}$, where $\tau_i \in (0, \infty)$. The entropy can thus be controlled by controlling the temperature (*e.g.*, $\tau_i = 0$ for full

exploitation, $\tau_i = \infty$ for full exploration). As for the behavior distillation function, we are inspired by the behavior design of GDI, which takes a weighted sum of two softmax distributions derived from two advantage functions. We can further extend this simple but efficient approach to the case to do a combination across different policies, *i.e.*, $\mu_{\mathbf{H},\boldsymbol{\tau},\boldsymbol{\omega}}(a|s) = \sum_{i=1}^{\mathrm{N}} \omega_i \pi_{\mathbf{h}_i,\tau_i}(a|s)$. This formula is actually a form of *mixture model*, where the importance weights play the role of mixture weights of the mixture model. Then the behavior space becomes,

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{\mu_{\mathbf{H},\boldsymbol{\psi}} = \sum_{i=1}^{\mathbf{N}} \omega_i \, \mathrm{softmax}_{\tau_i}(\Phi_{\mathbf{h}_i}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{10}$$

4.2 MAB Based Behavior Selection

According to Proposition 2, the behavior selection over behavior space 10 can be simplified to the selection of $\boldsymbol{\psi}$. In this paper, we use MAB-based meta-controller to select $\boldsymbol{\psi} \in \boldsymbol{\Psi}$. Since $\boldsymbol{\Psi}$ is a continuous multidimensional space, we discretize $\boldsymbol{\Psi}$ into K regions $\{\Psi_1, ..., \Psi_{\mathrm{K}}\}$, and each region corresponds to an arm of MAB. At the beginning of a trajectory i, l-th actor will use MAB to sample a region Ψ_k indexed by arm k according to $\mathcal{P}_\Psi = \mathrm{softmax}(\mathrm{Score}_{\Psi_k}) = \frac{e^{\mathrm{Score}_{\Psi_k}}}{\sum_j e^{\mathrm{Score}_{\Psi_j}}}$. We adopt UCB score as $\mathrm{Score}_{\Psi_k} = V_{\Psi_k} + c \cdot \sqrt{\frac{\log(1+\sum_{j\neq k}^{\mathrm{K}} N_{\Psi_j})}{1+N_{\Psi_k}}}$ to tackle the reward-diversity trade-off problem in equation 7 (Garivier & Moulines, 2011). N_{Ψ_k} means the number of the visit of Ψ_k indexed by arm k. V_{Ψ_k} is calculated by the expectation of the undiscounted episodic returns to measure the value of each Ψ_k, and the UCB item $\sqrt{\frac{\log(1+\sum_{j\neq k}^{\mathrm{K}} N_{\Psi_j})}{1+N_{\Psi_k}}}$ is used to avoid selecting the same arm repeatedly and ensure sufficient diverse behavior mappings can be selected to boost the behavior diversity.

After an Ψ_k is sampled, a $\boldsymbol{\psi}$ will be uniformly sampled from Ψ_k, corresponding to a behavior mapping $\mathcal{F}_{\boldsymbol{\psi}}$. With $\mathcal{F}_{\boldsymbol{\psi}}$, we can obtain a behavior $\mu_{\boldsymbol{\psi}}$ according to equation 10. Then, the l-th actor acts $\mu_{\boldsymbol{\psi}}$ to obtain a trajectory τ_i and the undiscounted episodic return G_i, then G_i is used to update the reward model V_{Ψ_k} of region Ψ_k indexed by arm k. For more details, see App. E.

5 Experiment

In this section, we design our experiment to answer the following questions:

- Whether our methods can outperform prior SOTA RL algorithms in both sample efficiency and final performance in Atari 1B Benchmarks (See Sec. 5.2 and Figs. 3)?
- Can our methods adaptively adjust the exploration-exploration trade-off (See Figs. 4)?
- How to enlarge or narrow down the behavior space? What is the performance of methods with different behavior spaces (See Sec. 5.4)?
- How much performance will be degraded without proper behavior selection (See Figs. 5)?

5.1 Experimental Setup

5.1.1 Experimental Details

We conduct our experiments in ALE (Bellemare et al., 2013). The standard pre-processing settings of Atari are identical to those of Agent57 (Badia et al., 2020a), and related parameters have been concluded in App. I. We employ a separate evaluation process to record scores continuously. We record the undiscounted episodic returns averaged over five seeds using a windowed mean over 32 episodes. To avoid any issues that aggregated metrics may have, App. J provides full learning curves for all games and detailed comparison tables of raw and normalized scores. Apart from the mean and median HNS, we also report how many human worlds records our agents have broken to emphasize the superhuman performance of our methods. For more experimental details, see App. H.

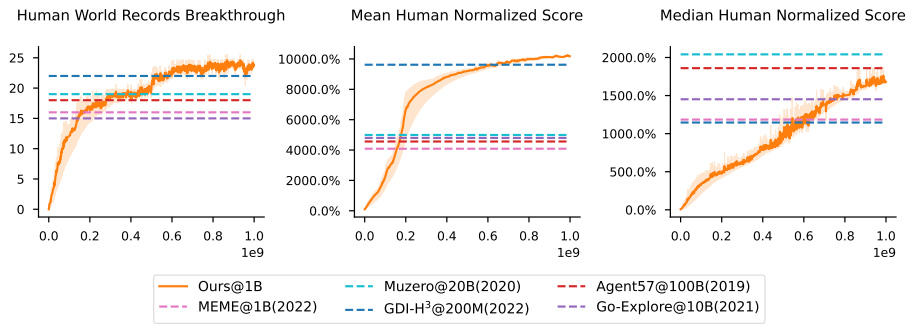


Figure 3: The learning curves in Atari. Curves are smoothed with a moving average over 5 points.

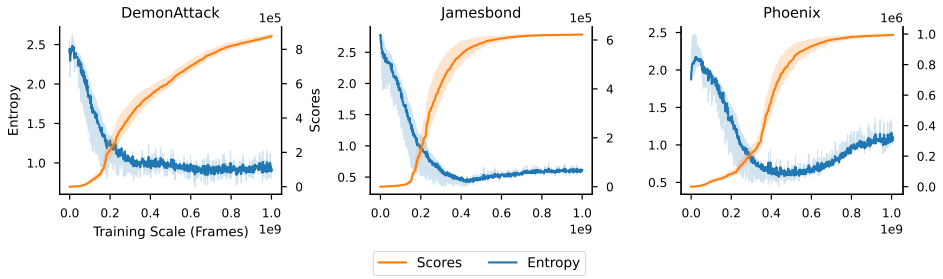


Figure 4: Behavior entropy and scores curve across training for different games where we achieved unprecedented performance. The names of the axes are the same as that of the leftmost figure.

5.1.2 Implementation Details

We jointly train three polices, and each policy can be indexed by the hyper-parameters $\mathbf{h}_i = (\gamma_i, \mathcal{RS}_i)$, wherein $\mathcal{RS}_i$ is a reward shaping method (Badia et al., 2020a), and γ_i is the discounted factor. Each policy model $\Phi_{\mathbf{h}_i}$ adopts the dueling network structure (Wang et al., 2016), where $\Phi_{\mathbf{h}_i} = A_{\mathbf{h}_i} = Q_{\mathbf{h}_i} - V_{\mathbf{h}_i}$. More details of the network structure can be found in App. L. To correct for harmful discrepancy of off-policy learning, we adopt V-Trace (Espeholt et al., 2018) and ReTrace (Munos et al., 2016) to learn $V_{\mathbf{h}_i}$ and $Q_{\mathbf{h}_i}$, respectively. The policy is learned by policy gradient (Schulman et al., 2017). Based on equation 10, we could build a behavior space with a hybrid mapping as $\mathbf{M}_{\mathbf{H},\mathbf{\Psi}} = \{\mu_{\mathbf{H},\boldsymbol{\psi}} = \sum_{i=1}^{3} \omega_i \operatorname{softmax}_{\tau_i}(\Phi_{\mathbf{h}_i})\}$, wherein $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3\}$, $\mathbf{\Psi} = \{\boldsymbol{\psi} = (\tau_1, \omega_1, \tau_2, \omega_2, \tau_3, \omega_3) | \tau_i \in (0, \tau^+), \sum_{j=1}^{3} \omega_j = 1\}$. The behavior selection is achieved by MAB described in Sec. 4.2, and more details can see App. E. Finally, we could obtain an implementation of LBC-$\mathcal{BM}$, which is our **main algorithm**. The target policy for A_1^π and A_2^π in GDI-H^3 is the same, while in our work the target policy for $A_i^{\pi_i}$ is $\pi_i = \operatorname{softmax}(A_i)$.

5.2 Summary of Results

Results on Atari Benchmark The aggregated results across games are reported in Figs. 3. On 200M training frames, GDI-H^3 obtains the highest mean HNS, median HNS and HWRB, which demonstrates its SOTA performance in the 200M Atari Benchmarks. Among the existing algorithms using more data than 1B, our agents achieve the best mean HNS and surpass the most human world records, which implies the effectiveness of our method.

Discussion of Overall Results With LBC, we can understand the mechanisms underlying the success of GDI-H^3 more clearly: **i)** GDI-H^3 has a high-capacity behavior space and a meta-controller to optimize the behavior selection **ii)** the target policy of A_1 and A_2 is the same, which enables stable learning and fast converge (See the case study of KL divergence in App. N). Compared to GDI-H^3, to ensure the behavior space will not degenerate, LBC maintains a population of diverse policies and, as a price, sacrifices some sample efficiency. Nevertheless, LBC can **continuously** maintain a significantly larger behavior space with hybrid behavior mapping, which enables RL agents to continuously explore and get improvement.

5.3 Case Study: Behavior Control

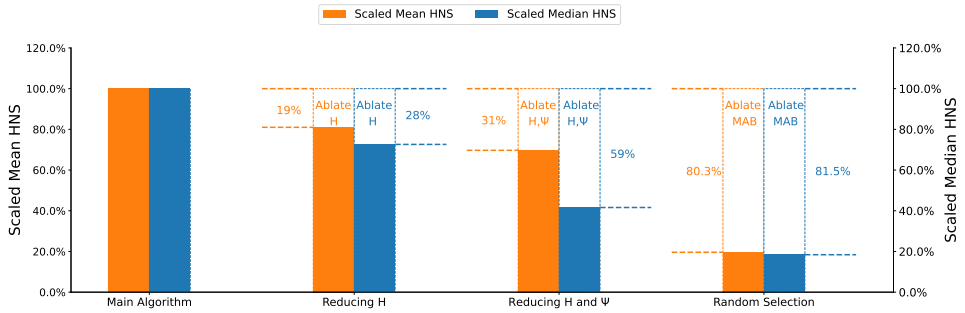


Figure 5: Ablation Results. All the results are scaled by the main algorithm to improve readability.

To further explore the mechanisms underlying the success of behavior control of our method, we adopt a case study to showcase our control process of behaviors. As shown in Figs. 4, in most tasks, our agents prefer exploratory behaviors first (i.e., high stochasticity policies with high entropy), and, as training progresses, the agents shift into producing experience from more exploitative behaviors. On the verge of peaking, the entropy of the behaviors could be maintained at a certain level (task-wise) instead of collapsing swiftly to zero to avoid converging prematurely to sub-optimal policies.

5.4 Ablation Study

In this section, we investigate several properties of our method. For more details, see App. K.

Behavior Space Decomposition To explore the effect of different behavior spaces, we decompose the behavior space of our main algorithm via reducing $\mathbf{H}$ and $\mathbf{\Psi}$:

1) Reducing $\mathbf{H}$. When we set all the policy models of our main algorithm the same, the behavior space transforms from $\mathcal{F}(\Phi_{\mathbf{h}_1}, \Phi_{\mathbf{h}_2}, \Phi_{\mathbf{h}_3})$ into $\mathcal{F}(\Phi_{\mathbf{h}_1}, \Phi_{\mathbf{h}_1}, \Phi_{\mathbf{h}_1})$. $\mathbf{H}$ degenerates from $\{\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3\}$ into $\{\mathbf{h}_1\}$. We can obtain a control group with a smaller behavior space by reducing $\mathbf{H}$.

2)Reducing $\mathbf{H}$ and $\mathbf{\Psi}$. Based on the control group reducing $\mathbf{H}$, we can further reduce $\mathbf{\Psi}$ to further narrow down the behavior space. Specially, we can directly adopt a individual behavior mapping to build the behavior space as $\mathbf{M}_{\mathbf{H},\mathbf{\Psi}} = \{\mu_\psi = \text{softmax}_\tau(\Phi_{\mathbf{h}_1})\}$, where $\mathbf{\Psi}$ degenerates from $\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3, \tau_1, \tau_2, \tau_3\}$ to $\{\tau\}$ and $\mathbf{H} = \{\mathbf{h}_1\}$. Then, we can obtain a control group with the smallest behavior space by reducing $\mathbf{H}$ and $\mathbf{\Psi}$.

The performance of these methods is illustrated in Figs. 5, and from left to right, the behavior space of the first three algorithms decreases in turn (According to Corollary 4 in App. C). It is evident that narrowing the behavior space via reducing $\mathbf{H}$ or $\mathbf{\Psi}$ will degrade the performance. On the contrary, the performance can be boosted by enlarging the behavior space, which could be a promising way to improve the performance of existing methods.

Behavior Selection To highlight the importance of an appropriate behavior selection, we replace the meta-controller of our main algorithm with a random selection. The ablation results are illustrated in Figs. 5, from which it is evident that, with the same behavior space, not learning an appropriate selection distribution of behaviors will significantly degrade the performance. We conduct a t-SNE analysis in App. K.3 to demonstrate that our methods can acquire more diverse behaviors than the control group with pre-defined behavior mapping. Another ablation study that removed the UCB item has been conducted in App. K.3 to demonstrate the behavior diversity may be boosted by the UCB item, which can encourage the agents to select more different behavior mappings.

6 Conclusion

We present the first deep reinforcement learning agent to break 24 human world records in Atari using only 1B training frames. To achieve this, we propose a general framework called LBC, which enables a significantly enlarged behavior selection space via formulating a hybrid behavior mapping from all policies, and constructs a unified learnable process for behavior selection. We introduced LBC into off-policy actor-critic methods and obtained a family of implementations. A large number of experiments on Atari have been conducted to demonstrate the effectiveness of our methods empirically. Apart from the full results, we do detailed ablation studies to examine the effectiveness of the proposed components.

REFERENCES

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 507–517. PMLR, 2020a. URL http://proceedings.mlr.press/v119/badia20a.html.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=Sye57xStvB.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.*, 47:253–279, 2013. doi: 10.1613/jair.3912. URL https://doi.org/10.1613/jair.3912.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1406–1415. PMLR, 2018. URL http://proceedings.mlr.press/v80/espeholt18a.html.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Jiajun Fan and Changnan Xiao. Generalized data distribution iteration. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6103–6184. PMLR, 2022. URL https://proceedings.mlr.press/v162/fan22c.html.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann (eds.), *Algorithmic Learning Theory - 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings*, volume 6925 of *Lecture Notes in Computer Science*, pp. 174–188. Springer, 2011. doi: 10.1007/978-3-642-24412-4_16. URL https://doi.org/10.1007/978-3-642-24412-4_16.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3215–3222. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17204.

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

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. *Advances in Neural Information Processing Systems*, 34:7703–7717, 2021.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *CoRR*, abs/1711.09846, 2017. URL http://arxiv.org/abs/1711.09846.

Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1lyTjAqYX.

Steven Kapturowski, Víctor Campos, Ray Jiang, Nemanja Rakićević, Hado van Hasselt, Charles Blundell, and Adrià Puigdomènech Badia. Human-level atari 200x faster. *CoRR*, abs/2209.07550, 2022. URL https://arxiv.org/abs/2209.07550.

Jinke Li, Ruonan Rao, and Jun Shi. Learning to trade with deep actor critic methods. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pp. 66–71. IEEE, 2018.

Bruce G Lindsay. Mixture models: theory, geometry, and applications. Ims, 1995.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *J. Artif. Intell. Res.*, 61:523–562, 2018. doi: 10.1613/jair.5699. URL https://doi.org/10.1613/jair.5699.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1046–1054, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c3992e9a68c5ae12bd18488bc579b30d-Abstract.html.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Marcin Choromanski, and Stephen J. Roberts. Effective diversity in population based reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d1dc3a8270a6f9394f88847d7f0050cf-Abstract.html.

Sepp Hochreiter; Jürgen Schmidhuber. Long short-term memory. *Neural Computation.*, 1997.

Simon Schmitt, Matteo Hessel, and Karen Simonyan. Off-policy actor-critic with shared experience replay. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8545–8554. PMLR, 2020. URL http://proceedings.mlr.press/v119/schmitt20a.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1889–1897. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL https://www.worldcat.org/oclc/37293240.

Marin Toromanoff, Émilie Wirbel, and Fabien Moutarde. Is deep reinforcement learning really superhuman on atari? *CoRR*, abs/1908.04683, 2019. URL http://arxiv.org/abs/1908.04683.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1995–2003. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/wangf16.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Appendix

Overview of the Appendix For ease of reading, we submit the main body of our paper together with the appendix as supplemental material. Below we will briefly introduce the structure of our appendix for easy reading. Readers can jump from the corresponding position in the body to the corresponding content in the appendix, or jump from the following table of contents to the corresponding content of interest in the appendix.

- In App. A, we briefly summarize some common notations in this paper for the convenience of readers.
- In App. B, we summarize and recall the background knowledge used in this paper.
- In App. C, we provide theoretical proofs.
- In App. E, we provide a detailed implementation of our MAB-based meta-controller.
- In App. F, we provide a detailed implementation of our core framework.
- In App. G, we use our framework to introduce an LBC-based version of well-known RL methods, which leads to a better understanding of their original counterparts.
- In App. H, we provide relevant details of our experiments.
- In App. I, we summarize the hyper-parameters used in our experiments.
- In App. J, we provides full learning curves for all games and detailed comparison tables of raw and normalized scores.
- In App. K, we provides the design of the ablation study and the overall results.
- In App. L, we summarize our model architecture in detail to for reproducibility.

A SUMMARY OF NOTATION AND ABBREVIATION

In this section, we briefly summarize some common notations in this paper for the convenience of readers, which are concluded in Tab. 1.

Table 1: Summary of Notation

Symbol	Description	Symbol	Description
s	State	$\mathcal{S}$	Set of all states
a	Action	$\mathcal{A}$	Set of all actions
$\mathbb{P}$	Probability distribution	μ	Behavior policy
π	Policy	G_t	Cumulative discounted reward at t
$d_{\rho_0}^{\pi}$	States visitation distribution of π	V_π	State value function of π
Q_π	State-action value function of π	A_π	Advantage function of π
γ	Discount-rate parameter	δ_t	Temporal-difference error at t
$\mathcal{F}$	Behavior mapping	Φ	Policy models
$\boldsymbol{\theta}$	Parameters of the policy network	$\boldsymbol{\Theta}$	Set of θ
$\mathbf{h}$	hyper-parameters of policy models	$\mathbf{H}$	Set of h
$\boldsymbol{\psi}$	Parameters to index $\mathcal{F}$	$\boldsymbol{\Psi}$	Set of $\boldsymbol{\psi}$
$\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}$	Behavior policy sets/space parameterized by $\theta, \mathbf{h}, \boldsymbol{\psi}$	$\mu_{\theta,\mathbf{h},\boldsymbol{\psi}}$	A behavior policy of $\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}$
$\lvert\mathbf{H}\rvert$	Size of set $\mathbf{H}$	$\lvert\boldsymbol{\Psi}\rvert$	Size of set $\boldsymbol{\Psi}$
$\mathcal{P}_{\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}}$	Behavior selection distribution over $\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}$	$\mathbb{1}_{\boldsymbol{\psi}=\boldsymbol{\psi}_0}$	One-point distribution $\mathbf{P}(\boldsymbol{\psi}=\boldsymbol{\psi}_0)=1$
V_μ^{TV}	Some measurement on the value of policy π	V_μ^{TD}	Some measurement on the diversity of policy π
$\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H}}$	Subspace of $\mathbf{M}_{\boldsymbol{\Theta},\mathbf{H},\boldsymbol{\Psi}}$	$\mathcal{RS}$	Reward shaping

B BACKGROUND

Most of the previous work has introduced the background knowledge of RL in detail. In this section, we only summarize and recall the background knowledge used in this paper. If you are interested in the relevant content, we recommend you read the relevant material (Badia et al., 2020a; Sutton & Barto, 1998; Badia et al., 2020b; Kapturowski et al., 2019; Espeholt et al., 2018). The relevant notations and abbreviations have been documented in App. A.

B.1 POLICY GRADIENT

Policy gradient (Williams, 1992, PG) is a policy-based RL methods, which updates the policy via optimizing the following target:

$$\mathcal{J}(\theta) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \log \pi_{\theta}\left(a_t \mid s_t\right) G(\tau)\right]$$

The policy gradient of Actor-Critic (Sutton & Barto, 1998, AC) methods is updated as follows:

$$\nabla_{\theta}\mathcal{J}(\theta) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \mathbf{\Phi}_t \nabla_{\theta} \log \pi_{\theta}\left(a_t \mid s_t\right)\right]$$

wherein $\mathbf{\Phi}_t$ could be $Q^{\pi}\left(s_t, a_t\right)$ or $A^{\pi}\left(s_t, a_t\right)$.

B.2 VTRACE

V-Trace is an off-policy technique proposed by IMPALA (Espeholt et al., 2018) to correct for the discrepancy between target policy and behavior policy. V-Trace estimates $V(s_t)$ by

$$V^{\tilde{\pi}}(s_t) = \mathbb{E}_{\mu}[V(s_t) + \sum_{k\geq 0} \gamma^k c_{[t:t+k-1]} \rho_{t+k} \delta^V_{t+k} V],$$

wherein $\delta^V_t V \stackrel{def}{=} r_t + \gamma V(s_{t+1}) - V(s_t)$ and $\rho_t = \min\{\frac{\pi_t}{\mu_t}, \bar{\rho}\}$.

B.3 RETRACE

ReTrace (Munos et al., 2016) estimates $Q(s_t, a_t)$ as follows:

$$Q^{\tilde{\pi}}(s_t, a_t) = \mathbb{E}_{\mu}[Q(s_t, a_t) + \sum_{k\geq 0} \gamma^k c_{[t+1:t+k]} \delta^Q_{t+k} Q],$$

where $\delta^Q_t Q \stackrel{def}{=} r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$.

C PROOF

*Proof of **Proposition** 1.* When $\mathcal{F}_{\boldsymbol{\psi}}$ is a pre-defined or rule-based mapping, $\mathcal{F}_{\boldsymbol{\psi}}$ of actor j at each training step (wall-clock) is deterministic, namely $\mathcal{F}_{\boldsymbol{\psi}} = \mathcal{F}_{\boldsymbol{\psi}_j}$, so each behavior of actor j can be uniquely indexed by $\mathbf{h} \in \mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_\mathrm{N}\}$, namely,

$$\mathbf{M}_{\mathbf{H}} = \{\mu_{\mathbf{h}} = \mathcal{F}_{\boldsymbol{\psi}_j}(\Phi_{\mathbf{h}}) | \mathbf{h} \in \mathbf{H}\}, \tag{11}$$

where $\boldsymbol{\psi}_j$ is the same for each behavior of actor j. Hence, the selection distribution of behavior can be simplified into the selection distribution over $\mathbf{h} \in \mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_N\}$ as:

$$\mathcal{P}_{\mu \in \mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}}} = \mathcal{P}_{\mu_{\mathbf{h}} \in \mathbf{M}_{\mathbf{H}}} = \mathcal{P}_{\mathbf{H}}, \tag{12}$$

where $\mathbf{M}_{\mathbf{H}} = \{\mathcal{F}_{\boldsymbol{\psi}_j}(\Phi_{\mathbf{h}}) | \mathbf{h} \in \mathbf{H}\}$ and $\mathcal{P}_{\mathbf{H}}$ is a selection distribution of $\mathbf{h} \in \mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_N\}$ and N is the number of policy models. Substituting equation 12 into equation 5, we can obtain

$$\begin{aligned}
\mathcal{L}_{\mathcal{P}} &= \mathbb{E}_{\mu \sim \mathcal{P}_{\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}}}}[V_{\mu}^{\mathrm{TV}} + c \cdot V_{\mu}^{\mathrm{TD}}] \\
&= \mathbb{E}_{\mu_{\mathbf{h}} \sim \mathcal{P}_{\mu_{\mathbf{H}}}}[V_{\mu_{\mathbf{h}}}^{\mathrm{TV}} + c \cdot V_{\mu_{\mathbf{h}}}^{\mathrm{TD}}] \\
&= \mathbb{E}_{\mathbf{h} \sim \mathcal{P}_{\mathbf{H}}}[V_{\mu_{\mathbf{h}}}^{\mathrm{TV}} + c \cdot V_{\mu_{\mathbf{h}}}^{\mathrm{TD}}]
\end{aligned}$$

□

Corollary 1 (Behavior Circling in Policy Model Selection)**.** *When the policy models overlap as $\Phi_{\mathbf{h}_1} \approx ... \approx \Phi_{\mathbf{h}_\mathrm{N}}$, all realizable behavior of actor j will overlap as $\mathcal{F}_{\boldsymbol{\psi}_j}(\Phi_{\mathbf{h}_1}) \approx ... \approx \mathcal{F}_{\boldsymbol{\psi}_j}(\Phi_{\mathbf{h}_\mathrm{N}})$. Behaviors from different model can not be distinguished by $\mathbf{h}_i$, the behavior selection via policy model selection becomes invalid.*

*Proof of **Proposition** 2.* When $\boldsymbol{\theta}, \mathbf{h}$ are shared among behaviors for each actor at each training step, such as $\mu = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\boldsymbol{\theta},\mathbf{h}})$ or $\mu = \mathcal{F}_{\boldsymbol{\psi}}(\Phi_{\boldsymbol{\theta}_1,\mathbf{h}_1}, ..., \Phi_{\boldsymbol{\theta}_\mathrm{N},\mathbf{h}_\mathrm{N}})$, each behavior for each behavior can be uniquely indexed by $\boldsymbol{\psi}$, namely,

$$\mathbf{M}_{\boldsymbol{\Psi}} = \{\mu_{\boldsymbol{\psi}} = \mathcal{F}_{\psi}(\boldsymbol{\Phi}_{1:\mathrm{N}}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\},$$

where $\boldsymbol{\Phi}_{1:\mathrm{N}}$ is the same among behaviors. Hence, the selection distribution of behavior can be simplified into the selection distribution of $\boldsymbol{\psi} \in \boldsymbol{\Psi}$ as

$$\mathcal{P}_{\mu \in \mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}}} = \mathcal{P}_{\mu_{\boldsymbol{\psi}} \in \mathbf{M}_{\boldsymbol{\Psi}}} = \mathcal{P}_{\boldsymbol{\Psi}}, \tag{13}$$

where $\mathcal{P}_{\boldsymbol{\Psi}}$ is a selection distribution of $\boldsymbol{\psi} \in \boldsymbol{\Psi}$. Substituting equation 13 into equation 5, we can obtain

$$\begin{aligned}
\mathcal{L}_{\mathcal{P}} &= \mathbb{E}_{\mu \sim \mathcal{P}_{\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}}}}[V_{\mu}^{\mathrm{TV}} + c \cdot V_{\mu}^{\mathrm{TD}}] \\
&= \mathbb{E}_{\mu_{\boldsymbol{\psi}} \sim \mathcal{P}_{\mathbf{M}_{\boldsymbol{\Psi}}}}[V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TV}} + c \cdot V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TD}}] \\
&= \mathbb{E}_{\boldsymbol{\psi} \sim \mathcal{P}_{\boldsymbol{\Psi}}}[V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TV}} + c \cdot V_{\mu_{\boldsymbol{\psi}}}^{\mathrm{TD}}]
\end{aligned}$$

□

The behavior mapping optimization may be a cure for the behavior circling:

Corollary 2 (Behavior Mapping Optimization Is An Antidote for Behavior Circling)**.** *As for an behavior mapping optimization method, the behavior of actor j is indexed by $\mathcal{F}_{\boldsymbol{\psi}}$. When all the policy models overlap as $\Phi_{\mathbf{h}_1} \approx ... \approx \Phi_{\mathbf{h}_\mathrm{N}}$, the realizable behavior of actor j are*

$$\mathcal{F}_{\boldsymbol{\psi}_1}(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}}), \mathcal{F}_{\boldsymbol{\psi}_2}(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}}), ..., \mathcal{F}_{\boldsymbol{\psi}_\infty}(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}}), \tag{14}$$

wherein $\boldsymbol{\psi}_i$ is a continuous parameter. Assuming $\mathcal{F}_{\psi}$ can be uniquely indexed by ψ and $\mathcal{F}_{\psi_i} \neq \mathcal{F}_{\psi_j}$, there are still infinite different behaviors that can be realized by actor j.

Proposition 3 (Comparison of Behavior Space)**.** *Given two behavior space $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ and $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$, if $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ is a sub-space of $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$, the space $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$ is not less than $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$. Furthermore, if $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ is a sub-space of $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$ and $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ is not equal to $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$, the space $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$ is larger than $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$.*

*Proof of **Proposition** 3.* Since $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ and $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$ are sets. When $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1} \subseteq \mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$, $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ is not larger than $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$. When $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1} \subset \mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$, $\mu_{\mathbf{\Theta}_1,\mathbf{H}_1,\mathbf{\Psi}_1}$ is smaller than $\mu_{\mathbf{\Theta}_2,\mathbf{H}_2,\mathbf{\Psi}_2}$ □

According to the behavior space construction formulation, we can draw the following Corollary:

Corollary 3. *Given the same policy model structure Φ and the same form of behavior mapping $\mathcal{F}$. Under Assumption 1, the behavior space can be fully determined by $\boldsymbol{H}$ and Ψ. For any two behavior space $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$ and $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$, if $\mathbf{H}_1 \subseteq \mathbf{H}_2$ and $\mathbf{\Psi}_1 \subseteq \mathbf{\Psi}_2$, the behavior space $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$ is a sub-space of $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$. Based on that, the space $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$ is not smaller than $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$.*

Corollary 4. *Given the same policy model structure Φ and the same form of behavior mapping $\mathcal{F}$. Under Assumption 1, the behavior space can be fully determined by $\boldsymbol{H}$ and Ψ. For any two behavior space $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$ and $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$, if at least one of the following conditions holds:*

- $\mathbf{H}_1 \subset \mathbf{H}_2$ *and* $\mathbf{\Psi}_1 \subseteq \mathbf{\Psi}_2$,
- $\mathbf{H}_1 \subseteq \mathbf{H}_2$ *and* $\mathbf{\Psi}_1 \subset \mathbf{\Psi}_2$,
- $\mathbf{H}_1 \subset \mathbf{H}_2$ *and* $\mathbf{\Psi}_1 \subset \mathbf{\Psi}_2$,

the behavior space $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$ is a sub-space of $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$ and $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$ is not equal to $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$. Based on that, the space $\mu_{\mathbf{H}_2,\mathbf{\Psi}_2}$ is larger than $\mu_{\mathbf{H}_1,\mathbf{\Psi}_1}$.

D BEHAVIOR SPACE CONSTRUCTION FOR MORE TASKS AND ALGORITHMS VIA LBC

Following the pipeline given in equation 9, different implementations of LBC can be acquired by simply selecting different entropy control function $f_{\tau_i}(\cdot)$ and behavior distillation function $g(\cdot, \boldsymbol{\omega})$ according to the corresponding RL algorithms and tasks.

D.1 SELECTION FOR ENTROPY CONTROL FUNCTION

Here we would give some examples for the selection of entropy control function $f_{\tau_i}(\cdot)$.

Continuous Control Tasks For tasks with continuous action spaces, the entropy control function can be selected as gaussian distribution, *i.e.*,

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{g\left(\mathrm{Normal}(\Phi_{\mathbf{h}_1}, \sigma_1), ..., \mathrm{Normal}(\Phi_{\mathbf{h}_\mathrm{N}}, \sigma_\mathrm{N}), \omega_1, ..., \omega_\mathrm{N}\right) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{15}$$

or uniform distribution, *i.e.*,

$$\begin{aligned}\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{g(\mathrm{Uniform}(\Phi_{\mathbf{h}_1} - b_1/2, \Phi_{\mathbf{h}_1} + b_1/2), ..., \mathrm{Uniform}(\Phi_{\mathbf{h}_\mathrm{N}} - b_\mathrm{N}/2, \Phi_{\mathbf{h}_\mathrm{N}} + b_\mathrm{N}/2), \\ \omega_1, ..., \omega_\mathrm{N}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\}\end{aligned} \tag{16}$$

where $\boldsymbol{\psi} = (\sigma_1, ..., \sigma_\mathrm{N}, \omega_1, ..., \omega_\mathrm{N})$ for gaussian distribution and $\boldsymbol{\psi} = (b_1, ..., b_\mathrm{N}, \omega_1, ..., \omega_\mathrm{N})$ for uniform distribution.

Discrete Control Tasks and Value-Based Algorithms

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{g\left(\epsilon_1\text{−greedy}(\Phi_{\mathbf{h}_1}), ..., \epsilon_\mathrm{N}\text{−greedy}(\Phi_{\mathbf{h}_\mathrm{N}}), \omega_1, ..., \omega_\mathrm{N}\right) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{17}$$

where $\boldsymbol{\psi} = (\epsilon_1, ..., \epsilon_\mathrm{N}, \omega_1, ..., \omega_\mathrm{N})$.

Discrete Control Tasks and Policy-Based Algorithms

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{g\left(\mathrm{softmax}_{\tau_1}(\Phi_{\mathbf{h}_1}), ..., \mathrm{softmax}_{\tau_\mathrm{N}}(\Phi_{\mathbf{h}_\mathrm{N}}), \omega_1, ..., \omega_\mathrm{N}\right) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{18}$$

where $\boldsymbol{\psi} = (\tau_1, ..., \tau_\mathrm{N}, \omega_1, ..., \omega_\mathrm{N})$.

D.2 SELECTION FOR BEHAVIOR DISTILLATION FUNCTION

Mixture Model

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{\sum_{i=1}^{\mathrm{N}} \omega_i f_{\tau_i}(\Phi_{\mathbf{h}_i}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{19}$$

Knowledge Distillation The knowledge distillation method can been seen as a derivative form of mixture model. The mixture model is simple and straightforward, but it requires more resources for model storage and inference. To address this disadvantage, we can distill the knowledge of multiple policies into a single network using knowledge distillation.

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{\mathrm{Distill}(f_{\tau_1}(\Phi_{\mathbf{h}_1}), ..., f_{\tau_\mathrm{N}}(\Phi_{\mathbf{h}_\mathrm{N}}), \omega_1, ..., \omega_\mathrm{N}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{20}$$

and the knowledge distillation process $\mathrm{Distill}(\cdot)$ can be realized by supervised learning.

Parameters Fusion Define μ_f as the generated behavior policy which shares the same network structure with the policy models in the population, and is parameterized by $\boldsymbol{\theta}_f$. Define $\boldsymbol{\theta}_{\mathbf{h}_i}$ as the parameters of policy $f_{\tau_i}(\Phi_{\mathbf{h}_i,\tau_i})$. Then we can define the parameters fusion function $\mathrm{Fusion}(\cdot, \boldsymbol{\omega})$,

$$\mathbf{M}_{\mathbf{H},\boldsymbol{\Psi}} = \{\mu_f = \mathrm{Fusion}(f_{\tau_1}(\Phi_{\mathbf{h}_1}), ..., f_{\tau_\mathrm{N}}(\Phi_{\mathbf{h}_\mathrm{N}}), \omega_1, ..., \omega_\mathrm{N}) | \boldsymbol{\psi} \in \boldsymbol{\Psi}\} \tag{21}$$

where $\boldsymbol{\theta}_f = \sum_{i=1}^{\mathrm{N}} \omega_i \boldsymbol{\theta}_{\mathbf{h}_i,\tau_i}$.

E ADAPTIVE CONTROL MECHANISM

In this paper, we cast the behavior control into the behavior mapping optimization, which can be further simplified into the selection of $\psi \in \Psi$. We formalize this problem via multi-armed bandits (MAB). In this section, we describes the multi-arm bandit (Sutton & Barto, 1998) paradigm and upper confidence bound (UCB) (Garivier & Moulines, 2011) algorithm. For a more thorough explanation and analysis, we refer the readers to (Garivier & Moulines, 2011).

E.1 DISCRETIZATION

Since Ψ is a continuous space, the optimization of $\psi \in \Psi$ is a continuous optimization problem. However, MAB usually only handle discrete control tasks. Hence, we have to discretize Ψ into K regions according to the discretization accuracy τ, wherein each arm of MAB corresponds to a region of the continuous space.

Remark. *The discretization accuracy τ is related to the accuracy of the algorithm. In general, a higher discretization accuracy indicates a higher accuracy of the algorithm, but correspondingly, a higher computational complexity of the algorithm.*

Example 1 (Example of Discretization)**.** *As for a ϵ-greedy behavior mapping, $\psi = \epsilon$ and $\Psi = \{\psi = \epsilon | \epsilon \in [0, 1]\}$. We can set the discretization accuracy $\tau = 0.1$, and we can discretize Ψ into 10 regions corresponding to $\mathrm{K} = 10$ arms. Each arm corresponds to an interval. For example, $k = 1$ corresponds to $[0, 0.1)$; $k = 1$ corresponds to $[0.1, 0.2)$...$k = 10$ corresponds to $[0.9, 1.0]$.*

E.2 SAMPLE AND UPDATE

We adopt the Thompson Sampling (Garivier & Moulines, 2011). $\mathcal{K} = \{1, ..., \mathrm{K}\}$ denote a set of arms available to the decision maker, who is interested in maximizing the expected cumulative return (Badia et al., 2020a; Parker-Holder et al., 2020). The optimal strategy for each actor is to pull the arm with the largest mean reward. At the beginning of each round, each actor will produce a sample mean from its mean reward model for each arm, and pulls the arm from which it obtained the largest sample. After observing the selected arm's reward, it updates its mean reward model.

In general, at each time t, MAB method will choose an arm k_t from all possible arms $\mathcal{K} = \{1, ..., \mathrm{K}\}$ according to a sampling distribution $\mathcal{P}_{\mathcal{K}}$, which is normally conditioned on the sequence of previous decisions and returns. Then we will uniformly sample the parameters ψ from this discretized regions. Based on the ψ, we can obtain the corresponding behavior according to $\mathcal{F}(\Phi_h)$ or $\mathcal{F}(\Phi_{h_1}, ..., \Phi_{h_N})$. Executing the behaviors in the environment, each actor will receive a excitation/reward signal $R_t(k_t) \in \mathbb{R}$, which will be used to update the MAB.

E.3 UPPER CONFIDENCE BOUND

The UCB (Garivier & Moulines, 2011) are often used to encourage MAB to try more the arms with a low frequency of use. Let's first define the number of times that the arm k has been selected within T rounds as follows:

$$N_T(x) = \sum_{t=0}^{T} \mathbb{1}_{k_t = x}. \tag{22}$$

Then we can obtain the empirical mean of the arm x within T rounds as follows:

$$V_T(x) = N_T(x) \sum_{t=0}^{T} R_t(x) \mathbb{1}_{k_t = x}. \tag{23}$$

The UCB methods encourage the decision maker (actor-wise) to maximize the UCB scores:

$$\mathrm{Score}_x = V_T(x) + c \cdot \sqrt{\frac{\log(1 + \sum_j \mathbf{N}_T(j))}{1 + N_T(x)}} \tag{24}$$

The optimal strategy for each actor is to pull the arm with the largest mean scores. At the beginning of each round, each actor will produce a sample mean from its mean reward model for each arm, and pulls the arm from which it obtained the largest sample. After observing the selected arm's scores, it updates its mean reward model.

Remark. *In practical, Z-score Normalization are normally used to normalized $V_T(x)$, namely $\frac{V_T(x)-\mathbb{E}[V_T]}{\boldsymbol{D}[V_T]}$, which can be formulated as*

$$\text{Score}_x = \frac{V_T(x) - \mathbb{E}[V_T(x)]}{\boldsymbol{D}[V_T(x)]} + c \cdot \sqrt{\frac{\log(1 + \sum_j \boldsymbol{N}_T(j))}{1 + N_T(x)}} \tag{25}$$

E.4 Population-Based MAB

In the non-stationary scenario, the distributions of $V_T(x)$ could be shifted in the course of the lifelong learning. The standard UCB-based MAB failed to adapt to the change of the reward distribution and thus we refer to a population-based MAB to handle this problem, which jointly train a population of MAB with different hyperparameters. The sampling and update procedure of MAB is slightly different from the origin MAB, which will be discussed in the following. The main implementation of our population-based MAB has been concluded in Algorithm 1.

E.4.1 MAB Population Formulation

Assuming there are N bandits $B_{\mathbf{h}_i}$ to from a population $\mathcal{B} = \{B_{\mathbf{h}_1}, ..., B_{\mathbf{h}_N}\}$, wherein each bandit can be uniquely indexed by its hyper-parameter $\mathbf{h}_i$ and keep other hyper-parameters remain the same such as the discretization. In this paper, $\mathbf{h}_i = c$, wherein c is the trade-off coefficient in equation 24, which is uniformly sampled from $[0.5, 1.5]$, *i.e.*, randomly select a $c \in [0.5, 1.5]$ while initializing each bandit.

E.4.2 Population-Based Sample

During the sampling procedure, each bandit $B_{\mathbf{h}_i}$ will sample D arm $k_i \in \mathcal{K}$ with the Top-D ucb-scores. After all the bandits sample D arm, there are $D \times N$ sampled arms. We summarize the number of times each arm is selected, and sorted in descending order by the number of times they are selected. Then, we can obtain an arm $x_{j,t}$ that is selected the most times, which is the sample output of the population-based MAB. Finally, we uniformly sample a $\psi_{j,t}$ from the region indexed by $x_{j,t}$.

Example 2. *Assuming there are 7 bandits, and each bandit will sample $D = 2$ arms from $\mathcal{K} = \{1, ..., 10\}$. Assuming that the sample output is as follows:*

$$1, 2, 1, 3, 2, 4, 5; 1, 1, 2, 2, 1, 1, 4.$$

Then, the arm $k = 1$ is the arm being selected the most times, so we can get the sampled arm $x_{j,t} = 1$.

Remark. *Noting that, if there are more than one arm that is selected the most times, we can uniformly sample one from these arms.*

E.4.3 Population-Based Update

With $x_{j,t}$, according to the behavior space equation 10 , we can obtain a behavior $\mu_{j,t} = \mathcal{F}_\psi(\Phi_{\mathbf{h}})$ or $\mu_{j,t} = \mathcal{F}_{\psi_{j,t}}(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\text{N}})$. Execute $\mu_{j,t}$ in the environment and we can obtain obtain the return $G_{j,t}$. With $G_{j,t}$, we can update each bandit in the population based on equation 22 - equation 24.

E.4.4 Bandit Replacement

Similar to the sliding windows (Badia et al., 2020a), to tackle the non-stationary problem, we have to track the changes of optimization objectives in a timely manner. To achieve this, we update the replace in the population regularly so that it captures short-term information to improve its tracking performance.

Noting that there are many methods to solve this non-stationary problem at present, such as the sliding windows (Badia et al., 2020a). Since this is not the main proposition of this paper, we just choose a feasible implementation to handle this problem.

Algorithm 1 Population-Based Multi-Arm Bandits (Actor-Wise)

// For Each Actor j
// Initialize Bandits Population
Initialize each bandit $B_{\mathbf{h}_i}$ in the population with different hyper-parameters c.
Incorporate each bandit together to form a population of bandits $\mathcal{B}$.
for each episode t **do**
  for each $B_{\mathbf{h}_i}$ in $\mathcal{B}$ **do**
    Sample D arms with Top-D UCB Score via equation 24.
  end for
  Summarize $N \times D$ arms and count the selected times of each arm.
  Uniformly sample an arm among arms that selected the most times to obtain arm $x_{j,t}$.
  Uniformly sample a $\psi_{j,t}$ from the region indexed by $x_{j,t}$.
  Obtain a behavior $\mu_{j,t} = \mathcal{F}_{\psi}(\Phi_{\mathbf{h}})$ or $\mu_{j,t} = \mathcal{F}_{\psi_{j,t}}(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}})$.
  Execute $\mu_{j,t}$ and obtain the return $G_{j,t}$.
  for each $B_{\mathbf{h}_i}$ in $\mathcal{B}$ **do**
    Update $B_{\mathbf{h}_i}$ via equation 22 and equation 23.
  end for
  Update each bandit in the population via equation 22 and equation 23.
  // Replace Bandit from The Population
  if t mod $T_{replace}$=0 **then**
    Remove one bandit from the bandit population Uniformly and recreate (reinitialize) one into it.
  end if
end for

F ALGORITHM PSEUDOCODE

We concluded our algorithm in in the Algorithm. 2. Apart from that, we also concluded our model architecture in App. L.

Algorithm 2 Learnable Behavior Control

Initialize the Data Buffer (DB), the Parameter Sever (PS), the Learner Push Parameter Interval d_{push} and the Actor Pull Paramter Interval d_{pull}.
// LEARNER i
Initialize the network parameter θ_i (for model structure, see App. L)
for Training Step t **do**
  Load data from DB.
  Estimate Q_{θ_i} by $Q^{\tilde{\pi}}(s_t, a_t) = \mathbb{E}_{\mu}[Q(s_t, a_t) + \sum_{k \geq 0} \gamma^k c_{[t+1:t+k]} \delta^Q_{t+k} Q]$, wherein the target policy $\pi_i = \mathrm{softmax}(A_i)$.
  Estimate V_{θ_i} by $V^{\tilde{\pi}}(s_t) = \mathbb{E}_{\mu}[V(s_t) + \sum_{k \geq 0} \gamma^k c_{[t:t+k-1]} \rho_{t+k} \delta^V_{t+k} V]$, wherein the target policy $\pi_i = \mathrm{softmax}(A_i)$.
  Update θ_i via $\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \mathbf{\Phi}_t \nabla_\theta \log \pi_{\theta_i}\left(a_t \mid s_t\right)\right]$.
  if t mod $d_{push} = 0$ **then**
    Push θ_i into PS.
  end if
end for
// ACTOR j
for kth episode at training step t **do**
  Sample a $\boldsymbol{\psi}_{j,k} = (\tau_1, ..., \tau_{\mathrm{N}}, \omega_1, ..., \omega_{\mathrm{N}})$ via the MAB-based meta-controller (see App. E).
  Generalized Policy Selection. Adjusting the contribution proportion of the each learned policies for the behavior via a importance weight $\boldsymbol{\omega} = (\omega_1, ..., \omega_{\mathbf{N}})$.
  Policy-Wise Entropy Control. Adjusting the entropy of each policy via a τ_i, (e.g., $\pi_{\tau_i} = \mathrm{softmax}_{\tau_i}(\Phi_i)$).
  Behavior Distillation from Multiple Policies. Distilling the entropy-controlled policies into a behavior policy $\mu_{j,k}$ via a mixture model $\mu = \sum_i^{\mathbf{N}} \pi_{\tau_i}$.
  Obtaining episode $\tau_{j,k}$ and reward $G_{j,k}$ via executing $\mu_{j,k}$, and push $\tau_{j,k}$ into DB.
  Update the meta-controller with $(\mu_{j,k}, G_{j,k})$.
  if t mod $d_{pull} = 0$ **then**
    Pull $\{\theta_1, ..., \theta_{\mathbf{N}}\}$ from PS.
  end if
end for

G AN LBC-BASED VERSION OF RL

The behavior space is vital for RL methods, which can be used to categorize RL algorithms. Given the model structure Φ, and the form of $\mathcal{F}$, the behavior space can be fully determined by $\mathbf{H}$ and $\mathbf{\Psi}$, which can be used to categorize RL methods. We say one algorithm belongs to LBC-$\mathbf{H}_{\mathrm{N}}^{\mathrm{C}}$-$\mathbf{\Psi}_{\mathrm{L}}^{\mathrm{K}}$ when **1)** the hyper-parameters $\mathbf{h}$ is a C-D vector and $\mathbf{h}$ has N possible values corresponding to N different policy models, and **2)** $\boldsymbol{\psi}$ is a K-D vector and $\boldsymbol{\psi}$ has L possible behavior mappings at each training step. Based on that, we can offer a general view to understand prior methods from the perspective of behavior control, which is illustrated in Tab. 2.

Table 2: An LBC-based Version of RL Methods.

Algorithm	PBT	Agent57	DvD	LBC-$\mathcal{BM}$ (Ours)
$\mathcal{F}$	$\epsilon - greedy$	$\epsilon - greedy$	identical mapping	$\sum_{i=1}^{\mathrm{N}} \omega_i \operatorname{Softmax}_{\tau_i}(\Phi_{\mathbf{h}_i})$
$\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}$	$\{\mathcal{F}_\psi(\Phi_{\mathbf{h}_j})\vert\mathbf{h}_j \in \mathbf{H}, \psi \in \mathbf{\Psi}\}$	$\{\mathcal{F}_\psi(\Phi_{\mathbf{h}_j})\vert\mathbf{h}_j \in \mathbf{H}, \psi \in \mathbf{\Psi}\}$	$\{\mathcal{F}_\psi(\Phi_{\mathbf{h}_j})\vert\mathbf{h}_j \in \mathbf{H}, \psi \in \mathbf{\Psi}\}$	$\{\mathcal{F}_\psi(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_\mathrm{N}})\vert\mathbf{h}_j \in \mathbf{H}, \psi \in \mathbf{\Psi}\}$
$\mathbf{H}$	$\{\mathbf{h}_i = (h_1, .., h_\mathrm{C})\vert i = 1, ..., \mathrm{N}\}$	$\{(\gamma_i, \beta_i)\vert i = 1, ..., \mathrm{N}\}$	$\{(\lambda_i)\vert i = 1, ..., \mathrm{N}\}$	$\{\mathbf{h}_i = (h_1, .., h_\mathrm{C})\vert i = 1, ..., \mathrm{N}\}$
$\mathbf{\Psi}$	$\{(\epsilon_l)\vert l = 1, ..., \mathrm{L}\}$	$\{(\epsilon_l)\vert l = 1, ..., \mathrm{L}\}$	$\{1\}$	$\{(\omega_1, \tau_1, ..., \omega_\mathrm{N}, \tau_\mathrm{N})\}$
Category	LBC-$\mathrm{H}_{\mathrm{N}}^{\mathrm{C}}$-$\mathbf{\Psi}_{\mathrm{L}}^{1}$	LBC-$\mathrm{H}_{\mathrm{N}}^{2}$-$\mathbf{\Psi}_{\mathrm{L}}^{1}$	LBC-$\mathrm{H}_{\mathrm{N}}^{1}$-$\mathbf{\Psi}_{1}^{1}$	LBC-$\mathrm{H}_{\mathrm{N}}^{\mathrm{C}}$-$\mathbf{\Psi}_{\infty}^{\mathrm{K}=2\mathrm{N}}$
$\vert\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}\vert$	$\mathrm{N} \times \mathrm{L}$	$\mathrm{N} \times \mathrm{L}$	N	∞
Meta-Controller ($\mathbf{H}$)	ES	MAB	MAB	Ensemble equation 10
Meta-Controller ($\mathbf{\Psi}$)	Rule-Based	Rule-Based	Rule-Based	MAB

H EXPERIMENT DETAILS

H.1 IMPLEMENTATION DETAILS

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). The burn-in technique is adopted to deal with the representational drift (Kapturowski et al., 2019), and we train each sample twice. A complete description of the hyper-parameters can see App. I.

H.2 EXPERIMENTAL SETUP

We employ additional environments to evaluate the scores during model training, and we record the undiscounted episode returns averaged over 5 seeds using a windowed mean over 32 episodes. We evaluated all agents on 57 Atari 2600 games from the arcade learning environment (Bellemare et al., 2013, ALE) by recording the average score of the population of agents during model training. Noting that episodes will be truncated at 100K frames as other baseline algorithms (Hessel et al., 2018; Badia et al., 2020a; Schmitt et al., 2020; Badia et al., 2020b; Kapturowski et al., 2019).

H.3 RESOURCES USED

All the experiment is accomplished using 10 workers with 72 cores CPU and 3 learners with 3 Tesla-V100-SXM2-32GB GPU.

I HYPER-PARAMETERS

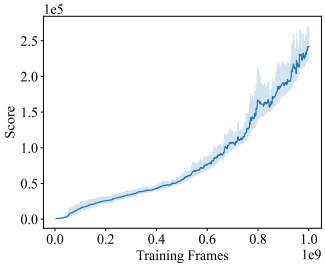
The hyper-parameters that we used in all experiments are like those of NGU Badia et al. (2020b) and Agent57 (Badia et al., 2020a). However, for completeness and readability, we detail them below in Tab. 3. We also include the hyper-parameters we used in the population-based MAB. For more details on the parameters in ALE, we refer the readers to see (Machado et al., 2018).

Table 3: Hyper-Parameters for Atari Experiments.

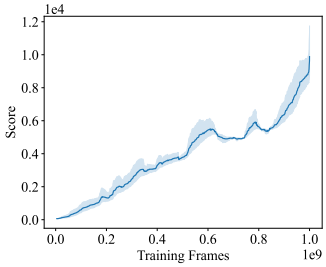
Parameter	**Value**	**Parameter**	**Value**
Burn-in	40	Replay	2
Seq-length	80	Burn-in Stored Recurrent State	Yes
Bootstrap	Yes	Batch size	64
V-loss Scaling (ξ)	1.0	Q-loss Scaling (α)	5.0
π-loss Scaling (β)	5.0	Importance sampling clip $\bar{c}$	1.05
Importance Sampling Clip $\bar{\rho}$	1.05	LSTM Units	256
Weight Decay Rate	0.01	Optimizer	Adam weight decay
Learning Rate	5.3e-4	Weight Decay Schedule	Anneal linearly to 0
Warmup Steps	4000	Learning Rate Schedule	Anneal linearly to 0
AdamW β_1	0.9	Auxiliary Forward Dynamic Task	Yes
AdamW ϵ	1e-6	Learner Push Model Every d_{push} Steps	25
AdamW β_2	0.98	Auxiliary Inverse Dynamic Task	Yes
AdamW Clip Norm	50.0	Actor Pull Model Every d_{pull} Steps	64
γ_1	0.997	$\mathcal{RS}_1$	$\text{sign}(x) \cdot (\sqrt{\lvert x\rvert + 1} - 1) + 0.001 \cdot x$
γ_2	0.999	$\mathcal{RS}_2$ (log scaling)	$\log(\lvert x\rvert + 1) \cdot (2 \cdot \mathbb{1}_{r \geq 0} - \mathbb{1}_{r \leq 0})$
γ_3	0.99	$\mathcal{RS}_3$	$0.3 \cdot min(\tanh x, 0) + 5 \cdot max(\tanh x, 0)$
Population Num.	7	UCB c	Uniformly sampled from $[0.5, 1.5]$
D of Top-D	4	Replacement Interval $T_{replace}$	50
Range of τ_i	$[0, \exp 4]$	Range of ω_i	$[0, 1]$
Discrete Accuracy of τ_i	0.2	Discrete Accuracy of ω_i	0.1
Max episode length	30 min	Image Size	(84, 84)
Grayscaled/RGB	Grayscaled	Life information	Not allowed
Action Space	Full	Sticky action probability	0.0
Num. Action Repeats	4	Random noops range	30
Num. Frame Stacks	4	Num. Atari Games	57 (Full)

J EXPERIMENTAL RESULTS

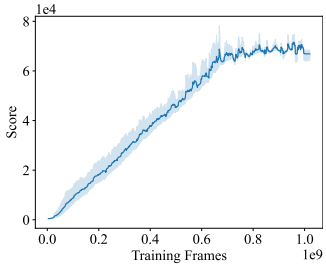
J.1 ATARI GAMES LEARNING CURVES



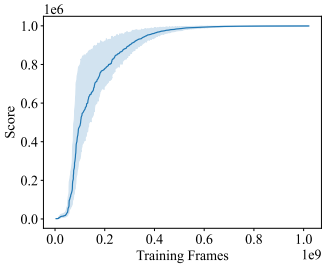
1. Alien



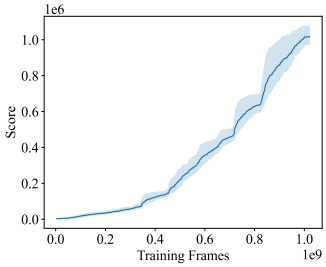
2. Amidar



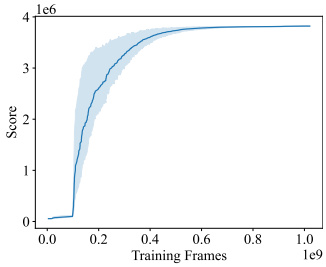
3. Assault



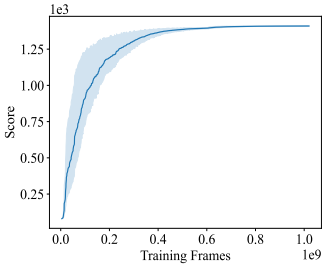
4. Asterix



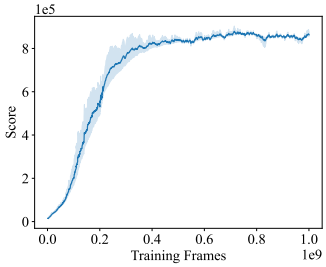
5. Asteroids



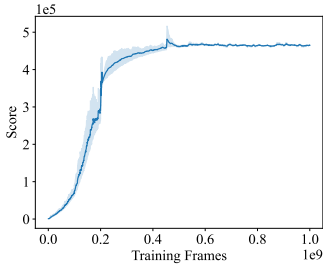
6. Atlantis



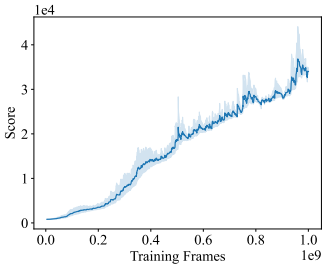
7. Bank_Heist



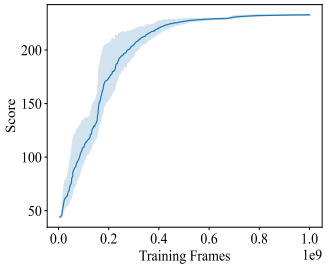
8. Battle_Zone



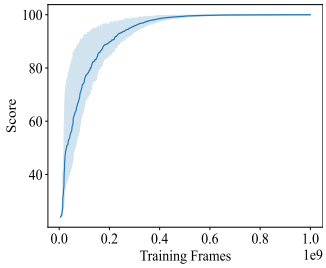
9. Beam_Rider



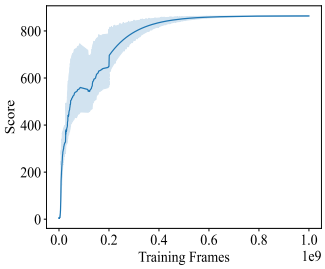
10. Berzerk



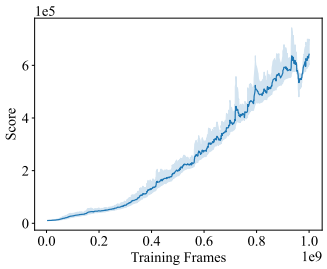
11. Bowling



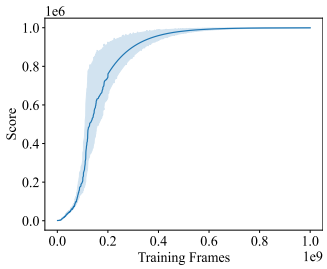
12. Boxing



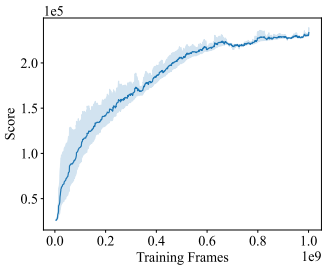
13. Breakout



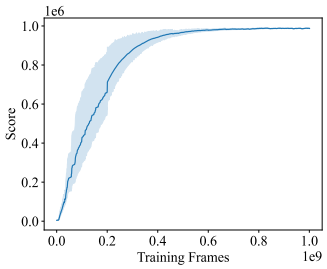
14. Centipede



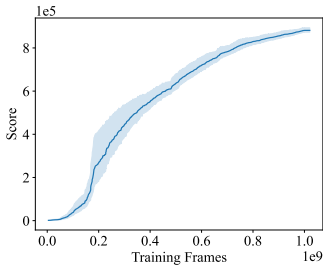
15. Chopper_Command



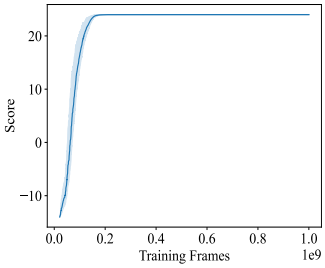
16. Crazy_Climber



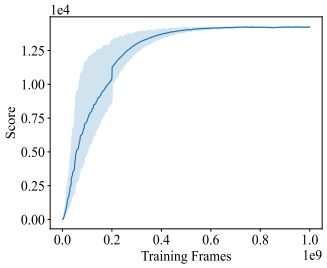
17. Defender



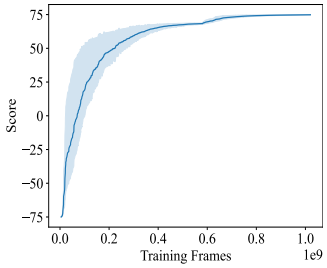
18. Demon_Attack



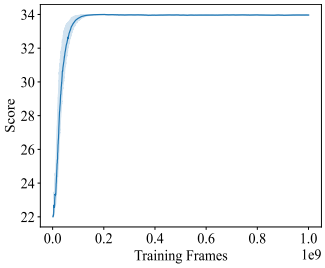
19. Double_Dunk



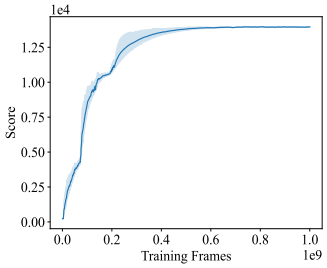
20. Enduro



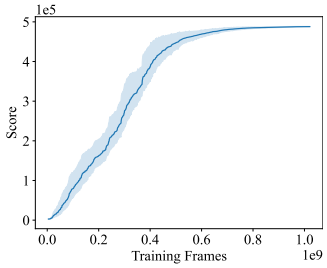
21. Fishing_Derby



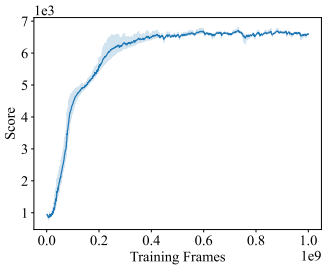
22. Freeway



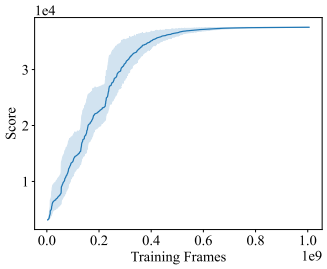
23. Frostbite



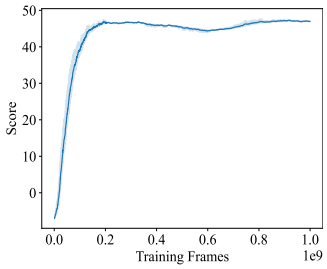
24. Gopher



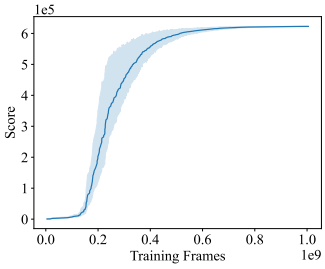
25. Gravitar



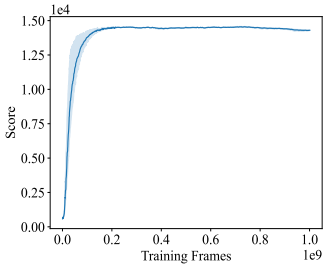
26. Hero



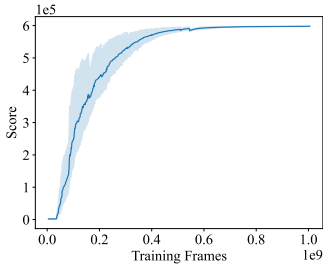
27. Ice_Hockey



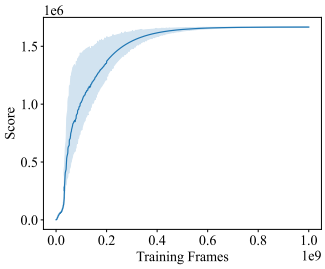
28. Jamesbond



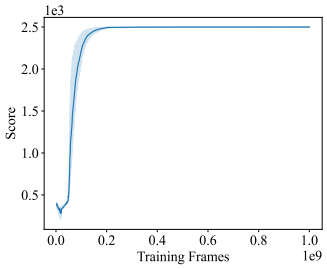
29. Kangaroo



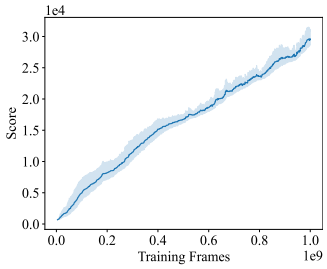
30. Krull



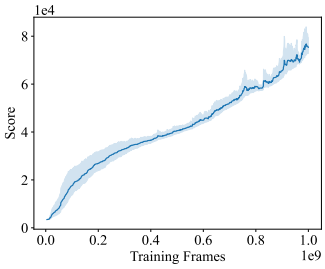
31. Kung_Fu_Master



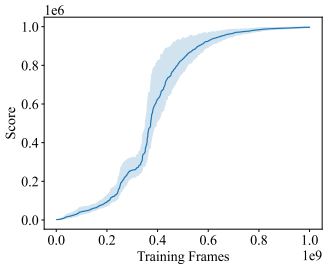
32. Montezuma_Revenge



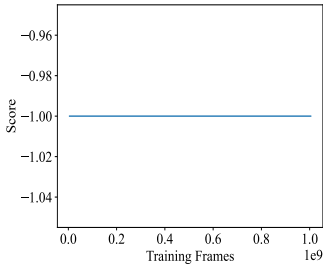
33. Ms_Pacman



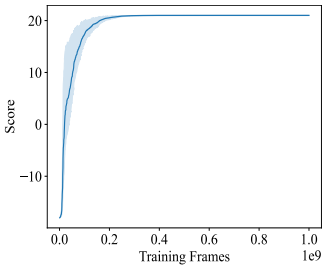
34. Name_This_Game



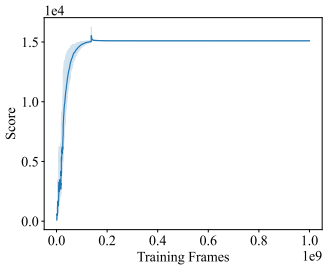
35. Phoenix



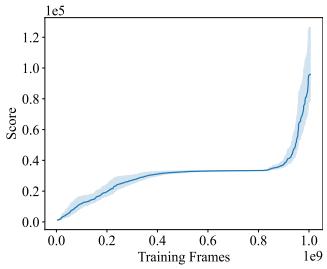
36. Pitfall



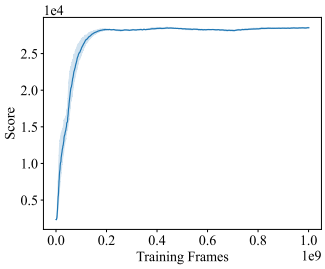
37. Pong



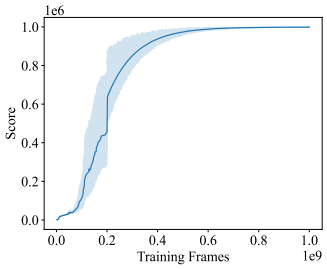
38. Private_Eye



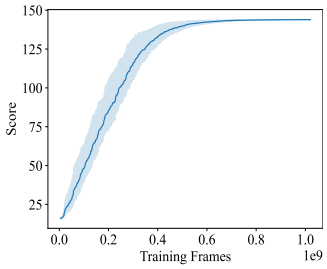
39. Qbert



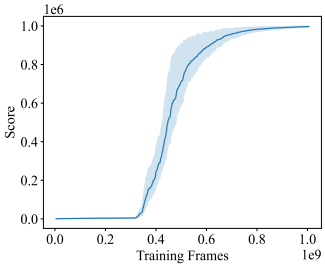
40. Riverraid



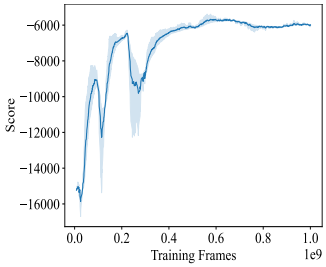
41. Road_Runner



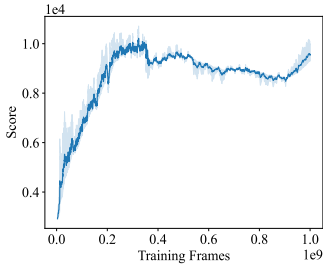
42. Robotank



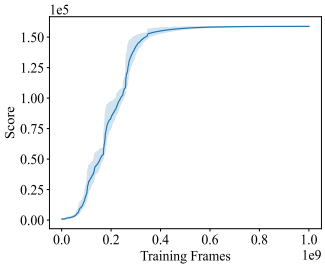
43. Seaquest



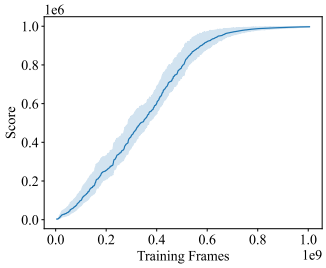
44. Skiing



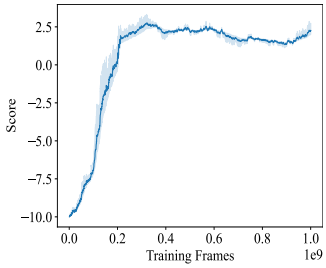
45. Solaris



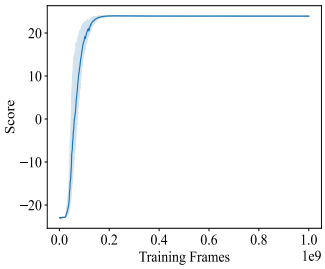
46. Space_Invaders



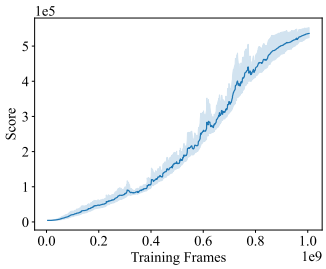
47. Star_Gunner



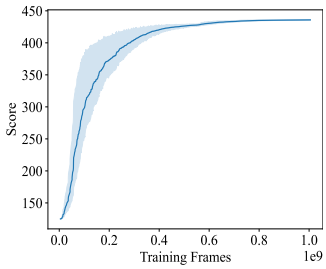
48. Surround



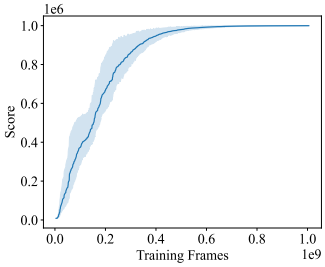
49. Tennis



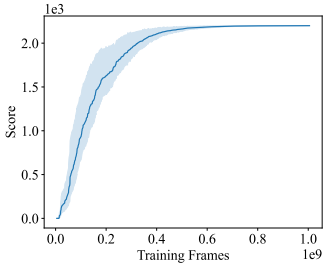
50. Time_Pilot



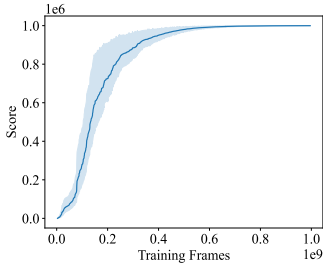
51. Tutankham



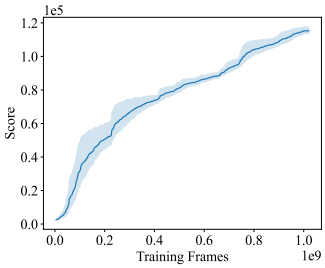
52. Up_N_Down



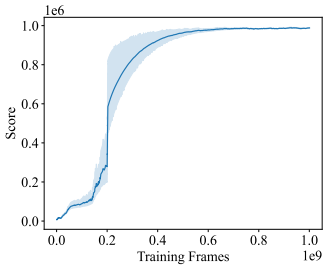
53. Venture



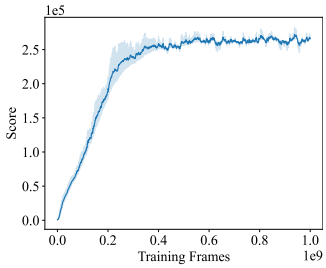
54. Video_Pinball



55. Wizard_of_Wor



56. Yars_Revenge



57. Zaxxon

J.2 ATARI GAMES TABLE OF SCORES BASED ON HUMAN AVERAGE SCORES

In this part, we detail the raw score of several representative SOTA algorithms , including the SOTA model-free algorithms, SOTA model-based algorithms, and other SOTA algorithms. Apart from the raw score, we also report the Human Average Scores for each game and Human Normalized Score (HNS) for each algorithm in Atari 57 games. More details on these algorithms can see Machado et al. (2018); Toromanoff et al. (2019).

Table 4: Score table of SOTA model-free algorithms on HNS(%).

Games Scale	RND	Average Human	AGENT57 100B	HNS(%)	Ours 1B	HNS(%)	MEME 1B	HNS(%)
Alien	227.8	7127.8	**297638.17**	**4310.30%**	279703.5	4050.37%	83683.43	1209.50%
Amidar	5.8	1719.5	**29660.08**	**1730.42%**	12996.3	758.04%	14368.9	838.13%
Assault	222.4	742	**67212.67**	**12892.66%**	62025.7	11894.40%	46635.86	8932.54%
Asterix	210	8503.3	991384.42	11951.51%	**999999**	**12055.38%**	769803.92	9279.71%
Asteroids	719	47388.7	150854.61	321.70%	**1106603.5**	**2369.60%**	364492.07	779.46%
Atlantis	12850	29028.1	1528841.76	9370.64%	**3824506.3**	**23560.59%**	1669226.33	10238.39%
Bank Heist	14.2	753.1	23071.5	3120.49%	1410	188.90%	**87792.55**	**11879.60%**
Battle Zone	236	37187.5	**934134.88**	**2527.36%**	857369	2319.62%	776770	2101.50%
Beam Rider	363.9	16926.5	300509.8	1812.19%	**457321**	**2758.97%**	51870.2	310.98%
Berzerk	123.7	2630.4	**61507.83**	**2448.80%**	35340	1404.89%	38838.35	1544.45%
Bowling	23.1	160.7	251.18	165.76%	233.1	152.62%	**261.74**	**173.43%**
Boxing	0.1	12.1	**100**	**832.50%**	**100**	**832.50%**	99.85	831.25%
Breakout	1.7	30.5	790.4	2738.54%	**864**	**2994.10%**	831.08	2879.79%
Centipede	2090.9	12017	412847.86	4138.15%	**728080**	**7313.94%**	245892.18	2456.16%
Chopper Command	811	7387.8	999900	15191.11%	**999999**	**15192.62%**	912225	13858.02%
Crazy Climber	10780.5	36829.4	**565909.85**	**2131.10%**	233090	853.43%	339274.67	1261.07%
Defender	2874.5	18688.9	677642.78	4266.80%	**995950**	**6279.56%**	543979.5	3421.60%
Demon Attack	152.1	1971	143161.44	7862.41%	**900170**	**49481.44%**	142176.58	7808.26%
Double Dunk	-18.6	-16.4	23.93	1933.18%	**24**	**1936.36%**	23.7	1922.73%
Enduro	0	860.5	2367.71	275.16%	**14332.5**	**1665.60%**	2360.64	274.33%
Fishing Derby	-91.7	-38.8	**86.97**	**337.75%**	75	315.12%	77.05	319.00%
Freeway	0	29.6	32.59	110.10%	**34**	**114.86%**	33.97	114.76%
Frostbite	65.2	4334.7	**541280.88**	**12676.32%**	13792.4	321.52%	526239.5	12324.03%
Gopher	257.6	2412.5	117777.08	5453.59%	**488900**	**22675.87%**	119457.53	5531.58%
Gravitar	173	3351.4	19213.96	599.07%	6372.5	195.05%	**20875**	**651.33%**
Hero	1027	30826.4	114736.26	381.58%	37545.6	122.55%	**199880.6**	**667.31%**
Ice Hockey	-11.2	0.9	**63.64**	**618.51%**	47.53	485.37%	47.22	482.81%
Jamesbond	29	302.8	135784.96	49582.16%	**623300.5**	**227637.51%**	117009.92	42724.95%
Kangaroo	52	3035	**24034.16**	**803.96%**	14372.6	480.07%	17311.17	578.58%
Krull	1598	2665.5	251997.31	23456.61%	**593679.5**	**55464.31%**	155915.32	14455.96%
Kung Fu Master	258.5	22736.3	206845.82	919.07%	**1666665**	**7413.57%**	476539.53	2118.90%
Montezuma Revenge	0	4753.3	9352.01	196.75%	2500	52.60%	**12437**	**261.65%**
Ms Pacman	307.3	6951.6	**63994.44**	**958.52%**	31403	468.01%	29747.91	443.10%
Name This Game	2292.3	8049	54386.77	904.94%	**81473**	**1375.45%**	40077.73	656.37%
Phoenix	761.5	7242.6	908264.15	14002.29%	**999999**	**15417.71%**	849969.25	13102.83%
Pitfall	-229.4	6463.7	18756.01	283.66%	-1	3.41%	**46734.79**	**701.68%**
Pong	-20.7	14.6	20.67	117.20%	**21**	**118.13%**	19.31	113.34%
Private Eye	24.9	69571.3	79716.46	114.59%	15100	21.68%	**100798.9**	**144.90%**
Qbert	163.9	13455	**580328.14**	**4365.06%**	151730	1140.36%	238453.5	1792.85%
Riverraid	1338.5	17118	63318.67	392.79%	27964.3	168.74%	**90333.12**	**563.99%**
Road Runner	11.5	7845	243025.8	3102.24%	**999999**	**12765.53%**	399511.83	5099.90%
Robotank	2.2	11.9	127.32	1289.90%	**144**	**1461.86%**	114.46	1157.32%
Seaquest	68.4	42054.7	999997.63	2381.56%	**1000000**	**2381.57%**	960181.39	2286.73%
Skiing	-17098	-4336.9	-4202.6	101.05%	-5903.34	87.72%	**-3273.43**	**108.33%**
Solaris	1236.3	12326.7	**44199.93**	**387.39%**	10732.5	85.63% 28175.53	242.91%	
Space Invaders	148	1668.7	48680.86	3191.48%	**159999.6**	**10511.71%**	57828.45	3793.02%
Star Gunner	664	10250	839573.53	8751.40%	**999999**	**10424.94%**	264286.33	2750.08%
Surround	-10	6.5	9.5	118.18%	2.726	77.13%	**9.82**	**120.12%**
Tennis	-23.8	-8.3	23.84	307.35%	**24**	**308.39%**	22.79	300.58%
Time Pilot	3568	5229.2	405425.31	24190.78%	**531614**	**31787.02%**	404751.67	24150.23%
Tutankham	11.4	167.6	**2354.91**	**1500.33%**	436.2	271.96%	1030.27	652.29%
Up N Down	533.4	11693.2	623805.73	5584.98%	**999999**	**8955.95%**	524631	4696.30%
Venture	0	1187.5	2623.71	220.94%	2200	185.26%	**2859.83**	**240.83%**
Video Pinball	0	17667.9	992340.74	5616.63%	**999999**	**5659.98%**	617640.95	3495.84%
Wizard of Wor	563.5	4756.5	**157306.41**	**3738.20%**	118900	2822.24%	71942	1702.33%
Yars Revenge	3092.9	54576.9	998532.37	1933.49%	**998970**	**1934.34%**	633867.66	1225.19%
Zaxxon	32.5	9173.3	**249808.9**	**2732.54%**	241570.6	2642.42%	77942.17	852.33%
Mean HNS				4762.17%		**10077.52%**		4081.14
Median HNS				**1933.49%**		1665.60%		1225.19

Table 5: Score table of SOTA model-based algorithms on HNS(%).

Games Scale	MuZero 20B	HNS(%)	EfficientZero 100K	HNS(%)	Ours 1B	HNS(%)
Alien	**741812.63**	**10747.61%**	808.5	8.42%	279703.5	4050.37%
Amidar	**28634.39**	**1670.57%**	148.6	8.33%	12996.3	758.04%
Assault	**143972.03**	**27665.44%**	1263.1	200.29%	62025.7	11894.40%
Asterix	998425	12036.40%	25557.8	305.64%	**999999**	**12055.38%**
Asteroids	678558.64	1452.42%	N/A	N/A	**1106603.5**	**2369.60%**
Atlantis	1674767.2	10272.64%	N/A	N/A	**3824506.3**	**23560.59%**
Bank Heist	1278.98	171.17%	351	45.58%	**1410**	**188.90%**
Battle Zone	848623	2295.95%	13871.2	36.90%	**857369**	**2319.62%**
Beam Rider	454993.53	2744.92%	N/A	N/A	**457321**	**2758.97%**
Berzerk	**85932.6**	**3423.18%**	N/A	N/A	35340	1404.89%
Bowling	**260.13**	**172.26%**	N/A	N/A	233.1	152.62%
Boxing	**100**	**832.50%**	52.7	438.33%	**100**	**832.50%**
Breakout	**864**	**2994.10%**	414.1	1431.94%	**864**	**2994.10%**
Centipede	**1159049.27**	**11655.72%**	N/A	N/A	728080	7313.94%
Chopper Command	991039.7	15056.39%	1117.3	4.66%	**999999**	**15192.62%**
Crazy Climber	**458315.4**	**1718.06%**	83940.2	280.86%	233090	853.43%
Defender	839642.95	5291.18%	N/A	N/A	**995950**	**6279.56%**
Demon Attack	143964.26	7906.55%	13003.9	706.57%	**900170**	**49481.44%**
Double Dunk	23.94	1933.64%	N/A	N/A	**24**	**1936.36%**
Enduro	2382.44	276.87%	N/A	N/A	**14332.5**	**1665.60%**
Fishing Derby	**91.16**	**345.67%**	N/A	N/A	75	315.12%
Freeway	33.03	111.59%	21.8	73.65%	**34**	**114.86%**
Frostbite	**631378.53**	**14786.59%**	296.3	5.41%	13792.4	321.52%
Gopher	130345.58	6036.85%	3260.3	139.34%	**488900**	**22675.87%**
Gravitar	**6682.7**	**204.81%**	N/A	N/A	6372.5	195.05%
Hero	**49244.11**	**161.81%**	3915.9	9.69%	37545.6	122.55%
Ice Hockey	**67.04**	**646.61%**	N/A	N/A	47.53	485.37%
Jamesbond	41063.25	14986.94%	517	178.23%	**623300.5**	**227637.51%**
Kangaroo	**16763.6**	**560.23%**	724.1	22.53%	14372.6	480.07%
Krull	269358.27	25082.93%	5663.3	380.82%	**593679.5**	**55464.31%**
Kung Fu Master	204824	910.08%	30944.8	136.52%	**1666665**	**7413.57%**
Montezuma Revenge	0	0.00%	N/A	N/A	**2500**	**52.60%**
Ms Pacman	**243401.1**	**3658.68%**	1281.2	14.66%	31403	468.01%
Name This Game	**157177.85**	**2690.53%**	N/A	N/A	81473	1375.45%
Phoenix	955137.84	14725.53%	N/A	N/A	**999999**	**15417.71%**
Pitfall	**0**	**3.43%**	N/A	N/A	-1	3.41%
Pong	**21**	**118.13%**	20.1	115.58%	**21**	**118.13%**
Private Eye	**15299.98**	**21.96%**	96.7	0.10%	15100	21.68%
Qbert	72276	542.56%	14448.5	107.47%	**151730**	**1140.36%**
Riverraid	**323417.18**	**2041.12%**	N/A	N/A	27964.3	168.74%
Road Runner	613411.8	7830.48%	17751.3	226.46%	**999999**	**12765.53%**
Robotank	131.13	1329.18%	N/A	N/A	**144**	**1461.86%**
Seaquest	999976.52	2381.51%	1100.2	2.46%	**1000000**	**2381.57%**
Skiing	-29968.36	-100.86%	N/A	N/A	**-5903.34**	**87.72%**
Solaris	56.62	-10.64%	N/A	N/A	**10732.5**	**85.63%**
Space Invaders	74335.3	4878.50%	N/A	N/A	**159999.6**	**10511.71%**
Star Gunner	549271.7	5723.01%	N/A	N/A	**999999**	**10424.94%**
Surround	**9.99**	**121.15%**	N/A	N/A	2.726	77.13%
Tennis	0	153.55%	N/A	N/A	**24**	**308.39%**
Time Pilot	**476763.9**	**28485.19%**	N/A	N/A	531614	31787.02%
Tutankham	**491.48**	**307.35%**	N/A	N/A	436.2	271.96%
Up N Down	715545.61	6407.03%	17264.2	149.92%	**999999**	**8955.95%**
Venture	0.4	0.03%	N/A	N/A	**2200**	**185.26%**
Video Pinball	981791.88	5556.92%	N/A	N/A	**999999**	**5659.98%**
Wizard of Wor	**197126**	**4687.87%**	N/A	N/A	118900	2822.24%
Yars Revenge	553311.46	1068.72%	N/A	N/A	**998970**	**1934.34%**
Zaxxon	**725853.9**	**7940.46%**	N/A	N/A	241570.6	2642.42%
Mean HNS		4994.97%		194.3%		**10077.52%**
Median HNS		**2041.12%**		109%		1665.60%

Table 6: Score table of SOTA exploration-based algorithms on HNS(%).

Games Scale	Go-Explore 10B	HNS	Ours 1B	HNS
Alien	**959312**	**13899.77%**	279703.5	4050.37%
Amidar	**19083**	**1113.22%**	12996.3	758.04%
Assault	30773	5879.64%	**62025.7**	**11894.40%**
Asterix	999500	12049.37%	**999999**	**12055.38%**
Asteroids	112952	240.48%	**1106603.5**	**2369.60%**
Atlantis	286460	1691.24%	**3824506.3**	**23560.59%**
Bank Heist	**3668**	**494.49%**	1410	188.90%
Battle Zone	**998800**	**2702.36%**	857369	2319.62%
Beam Rider	371723	2242.15%	**457321**	**2758.97%**
Berzerk	**131417**	**5237.69%**	35340	1404.89%
Bowling	**247**	**162.72%**	233.1	152.62%
Boxing	91	757.50%	**100**	**832.50%**
Breakout	774	2681.60%	**864**	**2994.10%**
Centipede	613815	6162.78%	**728080**	**7313.94%**
Chopper Command	996220	15135.16%	**999999**	**15192.62%**
Crazy Climber	**235600**	**863.07%**	233090	853.43%
Defender	N/A	N/A	**995950**	**6279.56%**
Demon Attack	239895	13180.65%	**900170**	**49481.44%**
Double Dunk	**24**	**1936.36%**	**24**	**1936.36%**
Enduro	1031	119.81%	**14332.5**	**1665.60%**
Fishing Derby	67	300.00%	**75**	**315.12%**
Freeway	**34**	**114.86%**	**34**	**114.86%**
Frostbite	**999990**	**23420.19%**	13792.4	321.52%
Gopher	134244	6217.75%	**488900**	**22675.87%**
Gravitar	**13385**	**415.68%**	6372.5	195.05%
Hero	**37783**	**123.34%**	37545.6	122.55%
Ice Hockey	33	365.29%	**47.53**	**485.37%**
Jamesbond	200810	73331.26%	**623300.5**	**227637.51%**
Kangaroo	**24300**	**812.87%**	14372.6	480.07%
Krull	63149	5765.90%	**593679.5**	**55464.31%**
Kung Fu Master	24320	107.05%	**1666665**	**7413.57%**
Montezuma Revenge	**24758**	**520.86%**	2500	52.60%
Ms Pacman	**456123**	**6860.25%**	31403	468.01%
Name This Game	**212824**	**3657.16%**	81473	1375.45%
Phoenix	19200	284.50%	**999999**	**15417.71%**
Pitfall	**7875**	**121.09%**	-1	3.41%
Pong	**21**	**118.13%**	**21**	**118.13%**
Private Eye	**69976**	**100.58%**	15100	21.68%
Qbert	**999975**	**7522.41%**	151730	1140.36%
Riverraid	**35588**	**217.05%**	27964.3	168.74%
Road Runner	999900	12764.26%	**999999**	**12765.53%**
Robotank	143	1451.55%	**144**	**1461.86%**
Seaquest	539456	1284.68%	**1000000**	**2381.57%**
Skiing	**-4185**	**101.19%**	-5903.34	87.72%
Solaris	**20306**	**171.95%**	10732.5	85.63%
Space Invaders	93147	6115.54%	**159999.6**	**10511.71%**
Star Gunner	609580	6352.14%	**999999**	**10424.94%**
Surround	N/A	N/A	**2.726**	**77.13%**
Tennis	**24**	**308.39%**	**24**	**308.39%**
Time Pilot	183620	10838.67%	**531614**	**31787.02%**
Tutankham	**528**	**330.73%**	436.2	271.96%
Up N Down	553718	4956.94%	**999999**	**8955.95%**
Venture	**3074**	**258.86%**	2200	185.26%
Video Pinball	**999999**	**5659.98%**	**999999**	**5659.98%**
Wizard of Wor	**199900**	**4754.03%**	118900	2822.24%
Yars Revenge	**999998**	**1936.34%**	998970	1934.34%
Zaxxon	18340	200.28%	**241570.6**	**2642.42%**
Mean HNS		4989.31%		**10077.52%**
Median HNS		1451.55%		**1665.60%**

Table 7: Score Table of GDI-H^3 and LBC-$\mathcal{BM}$ (Ours) on HNS(%).

Games Scale	GDI-H^3 200M	HNS	Ours 1B	HNS
Alien	48735	703.00%	**279703.5**	**4050.37%**
Amidar	1065	61.81%	**12996.3**	**758.04%**
Assault	**97155**	**18655.23%**	62025.7	11894.40%
Asterix	**999999**	**12055.38%**	**999999**	**12055.38%**
Asteroids	760005	1626.94%	**1106603.5**	**2369.60%**
Atlantis	**3837300**	**23639.67%**	3824506.3	23560.59%
Bank Heist	1380	184.84%	**1410**	**188.90%**
Battle Zone	824360	2230.29%	**857369**	**2319.62%**
Beam Rider	422390	2548.07%	**457321**	**2758.97%**
Berzerk	14649	579.46%	**35340**	**1404.89%**
Bowling	205.2	132.34%	**233.1**	**152.62%**
Boxing	**100**	**832.50%**	**100**	**832.50%**
Breakout	**864**	**2994.10%**	**864**	**2994.10%**
Centipede	195630	1949.80%	**728080**	**7313.94%**
Chopper Command	**999999**	**15192.62%**	**999999**	**15192.62%**
Crazy Climber	**241170**	**919.76%**	233090	853.43%
Defender	970540	6118.89%	**995950**	**6279.56%**
Demon Attack	787985	43313.70%	**900170**	**49481.44%**
Double Dunk	**24**	**1936.36%**	**24**	**1936.36%**
Enduro	14300	1661.82%	**14332.5**	**1665.60%**
Fishing Derby	65	296.22%	**75**	**315.12%**
Freeway	**34**	**114.86%**	**34**	**114.86%**
Frostbite	11330	263.84%	**13792.4**	**321.52%**
Gopher	473560	21964.01%	**488900**	**22675.87%**
Gravitar	5915	180.66%	**6372.5**	**195.05%**
Hero	**38225**	**124.83%**	37545.6	122.55%
Ice Hockey	47.11	481.90%	**47.53**	**485.37%**
Jamesbond	620780	226716.95%	**623300.5**	**227637.51%**
Kangaroo	14636	488.90%	14372.6	480.07%
Krull	594540	55544.92%	593679.5	55464.31%
Kung Fu Master	**1666665**	**7413.57%**	**1666665**	**7413.57%**
Montezuma Revenge	**2500**	**52.60%**	**2500**	**52.60%**
Ms Pacman	**11573**	**169.55%**	**31403**	**468.01%**
Name This Game	36296	590.68%	**81473**	**1375.45%**
Phoenix	959580	14794.07%	**999999**	**15417.71%**
Pitfall	-4.3	3.36	**-1**	**3.41%**
Pong	21	118.13%	**21**	**118.13%**
Private Eye	15100	21.68%	15100	21.68%
Qbert	28657	214.38%	**151730**	**1140.36%**
Riverraid	**28349**	**171.17%**	27964.3	168.74%
Road Runner	999999	12765.53%	**999999**	**12765.53%**
Robotank	113.4	1146.39%	**144**	**1461.86%**
Seaquest	**1000000**	**2381.57%**	**1000000**	**2381.57%**
Skiing	**-6025**	**86.77%**	-5903.34	87.72%
Solaris	9105	70.95%	**10732.5**	**85.63%**
Space Invaders	154380	10142.17%	**159999.6**	**10511.71%**
Star Gunner	677590	7061.61%	**999999**	**10424.94%**
Surround	2.606	76.40%	**2.726**	**77.13%**
Tennis	**24**	**308.39%**	**24**	**308.39%**
Time Pilot	450810	26924.45%	**531614**	**31787.02%**
Tutankham	418.2	260.44%	**436.2**	**271.96%**
Up N Down	966590	8656.58%	**999999**	**8955.95%**
Venture	2000	168.42%	**2200**	**185.26%**
Video Pinball	978190	5536.54%	**999999**	**5659.98%**
Wizard of Wor	63735	1506.59%	**118900**	**2822.24%**
Yars Revenge	968090	1874.36%	**998970**	**1934.34%**
Zaxxon	216020	2362.89%	**241570.6**	**2642.42%**
Mean HNS		4989.31%		**10077.52%**
Median HNS		1451.55%		**1665.60%**

J.3 ATARI GAMES TABLE OF SCORES BASED ON HUMAN WORLD RECORDS

In this part, we detail the raw score of several representative SOTA algorithms, including the SOTA model-free algorithms, SOTA model-based algorithms, and other SOTA algorithms. Apart from the raw score, we also report the Human World Records for each game and **H**uman **W**orld **R**ecords **B**reakthrough (HWRB) for each algorithm in Atari 57 games. More details on these algorithms can see Machado et al. (2018); Toromanoff et al. (2019).

Table 8: Score table of SOTA model-free algorithms on HWRB.

Games Scale	RND	Human World Records	AGENT57 100B	HWRB	Ours 1B	HWRB	MEME 1B	HWRB
Alien	227.8	251916	297638.17	1	279703.5	1	83683.43	0
Amidar	5.8	104159	29660.08	0	12996.3	0	14368.9	0
Assault	222.4	8647	67212.67	1	62025.7	1	46635.86	1
Asterix	210	1000000	991384.42	0	999999	0	769803.92	0
Asteroids	719	10506650	150854.61	0	1106603.5	0	364492.07	0
Atlantis	12850	10604840	1528841.76	0	3824506.3	0	1669226.33	0
Bank Heist	14.2	82058	23071.5	0	1410	0	87792.55	1
Battle Zone	236	801000	934134.88	1	857369	1	776770	0
Beam Rider	363.9	999999	300509.8	0	457321	0	51870.2	0
Berzerk	123.7	1057940	61507.83	0	35340	0	38838.35	0
Bowling	23.1	300	251.18	0	233.1	0	261.74	0
Boxing	0.1	100	100	1	100	1	99.85	0
Breakout	1.7	864	790.4	0	864	1	831.08	0
Centipede	2090.9	1301709	412847.86	0	728080	0	245892.18	0
Chopper Command	811	999999	999900	0	999999	1	912225	0
Crazy Climber	10780.5	219900	565909.85	1	233090	1	339274.67	1
Defender	2874.5	6010500	677642.78	0	995950	0	543979.5	0
Demon Attack	152.1	1556345	143161.44	0	900170	0	142176.58	0
Double Dunk	-18.6	21	23.93	1	24	1	23.7	1
Enduro	0	9500	2367.71	0	14332.5	1	2360.64	0
Fishing Derby	-91.7	71	86.97	1	75	1	77.05	1
Freeway	0	38	32.59	0	34	0	33.97	0
Frostbite	65.2	454830	541280.88	1	13792.4	0	526239.5	1
Gopher	257.6	355040	117777.08	0	488900	1	119457.53	0
Gravitar	173	162850	19213.96	0	6372.5	0	20875	0
Hero	1027	1000000	114736.26	0	37545.6	0	199880.6	0
Ice Hockey	-11.2	36	63.64	1	47.53	1	47.22	1
Jamesbond	29	45550	135784.96	1	623300.5	1	117009.92	1
Kangaroo	52	1424600	24034.16	0	14372.6	0	17311.17	0
Krull	1598	104100	251997.31	1	593679.5	1	155915.32	1
Kung Fu Master	258.5	1000000	206845.82	0	1666665	1	476539.53	0
Montezuma Revenge	0	1219200	9352.01	0	2500	0	12437	0
Ms Pacman	307.3	290090	63994.44	0	31403	0	29747.91	0
Name This Game	2292.3	25220	54386.77	1	81473	1	40077.73	1
Phoenix	761.5	4014440	908264.15	0	999999	0	849969.25	0
Pitfall	-229.4	114000	18756.01	0	-1	0	46734.79	0
Pong	-20.7	21	20.67	0	21	1	19.31	0
Private Eye	24.9	101800	79716.46	0	15100	0	100798.9	0
Qbert	163.9	2400000	580328.14	0	151730	0	238453.5	0
Riverraid	1338.5	1000000	63318.67	0	27964.3	0	90333.12	0
Road Runner	11.5	2038100	243025.8	0	999999	0	399511.83	0
Robotank	2.2	76	127.32	1	144	1	114.46	1
Seaquest	68.4	999999	999997.63	0	1000000	1	960181.39	0
Skiing	-17098	-3272	-4202.6	0	-5903.34	0	-3273.43	0
Solaris	1236.3	111420	44199.93	0	10732.5	0	28175.53	0
Space Invaders	148	621535	48680.86	0	159999.6	0	57828.45	0
Star Gunner	664	77400	839573.53	1	999999	1	264286.33	1
Surround	-10	9.6	9.5	0	2.726	0	9.82	1
Tennis	-23.8	21	23.84	1	24	1	22.79	1
Time Pilot	3568	65300	405425.31	1	531614	1	404751.67	1
Tutankham	11.4	5384	2354.91	0	436.2	0	1030.27	0
Up n Down	533.4	82840	623805.73	1	999999	1	524631	1
Venture	0	38900	2623.71	0	2200	0	2859.83	0
Video Pinball	0	89218328	992340.74	0	999999	0	617640.95	0
Wizard of Wor	563.5	395300	157306.41	0	118900	0	71942	0
Yars Revenge	3092.9	15000105	998532.37	0	998970	0	633867.66	0
Zaxxon	32.5	83700	249808.9	1	241570.6	1	77942.17	0
$\sum$ HWRB				18		**24**		16

Table 9: Score table of SOTA model-based algorithms on HWRB.

Games Scale	MuZero 20B	HWRB	EfficientZero 100K	HWRB	Ours 1B	HWRB
Alien	741812.63	1	808.5	0	279703.5	1
Amidar	28634.39	0	148.6	0	12996.3	0
Assault	143972.03	1	1263.1	0	62025.7	1
Asterix	998425	0	25557.8	0	999999	0
Asteroids	678558.64	0	N/A	N/A	1106603.5	0
Atlantis	1674767.2	0	N/A	N/A	3824506.3	0
Bank Heist	1278.98	0	351	0	1410	0
Battle Zone	848623	1	13871.2	0	857369	1
Beam Rider	454993.53	0	N/A	N/A	457321	0
Berzerk	85932.6	0	N/A	N/A	35340	0
Bowling	260.13	0	N/A	N/A	233.1	0
Boxing	100	1	52.7	0	100	1
Breakout	864	1	414.1	0	864	1
Centipede	1159049.27	0	N/A	N/A	728080	0
Chopper Command	991039.7	0	1117.3	0	999999	1
Crazy Climber	458315.4	1	83940.2	0	233090	1
Defender	839642.95	0	N/A	N/A	995950	0
Demon Attack	143964.26	0	13003.9	0	900170	0
Double Dunk	23.94	1	N/A	N/A	24	1
Enduro	2382.44	0	N/A	N/A	14332.5	1
Fishing Derby	91.16	1	N/A	N/A	75	1
Freeway	33.03	0	21.8	0	34	0
Frostbite	631378.53	1	296.3	0	13792.4	0
Gopher	130345.58	0	3260.3	0	488900	1
Gravitar	6682.7	0	N/A	N/A	6372.5	0
Hero	49244.11	0	3915.9	0	37545.6	0
Ice Hockey	67.04	1	N/A	N/A	47.53	1
Jamesbond	41063.25	0	517	0	623300.5	1
Kangaroo	16763.6	0	724.1	0	14372.6	0
Krull	269358.27	1	5663.3	0	593679.5	1
Kung Fu Master	204824	0	30944.8	0	1666665	1
Montezuma Revenge	0	0	N/A	N/A	2500	0
Ms Pacman	243401.1	0	1281.2	0	31403	0
Name This Game	157177.85	1	N/A	N/A	81473	1
Phoenix	955137.84	0	N/A	N/A	999999	0
Pitfall	0	0	N/A	N/A	-1	0
Pong	21	1	20.1	0	21	1
Private Eye	15299.98	0	96.7	0	15100	0
Qbert	72276	0	14448.5	0	151730	0
Riverraid	323417.18	0	N/A	N/A	27964.3	0
Road Runner	613411.8	0	17751.3	0	999999	0
Robotank	131.13	1	N/A	N/A	144	1
Seaquest	999976.52	0	1100.2	0	1000000	1
Skiing	-29968.36	0	N/A	N/A	-5903.34	0
Solaris	56.62	0	N/A	N/A	10732.5	0
Space Invaders	74335.3	0	N/A	N/A	159999.6	0
Star Gunner	549271.7	1	N/A	N/A	999999	1
Surround	9.99	1	N/A	N/A	2.726	0
Tennis	0	0	N/A	N/A	24	1
Time Pilot	476763.9	1	N/A	N/A	531614	1
Tutankham	491.48	0	N/A	N/A	436.2	0
Up N Down	715545.61	1	17264.2	0	999999	1
Venture	0.4	0	N/A	N/A	2200	0
Video Pinball	981791.88	0	N/A	N/A	999999	0
Wizard of Wor	197126	0	N/A	N/A	118900	0
Yars Revenge	553311.46	0	N/A	N/A	998970	0
Zaxxon	725853.9	1	N/A	N/A	241570.6	1
$\sum$ HWRB		19		0		**24**

Table 10: Score table of SOTA exploration-based algorithms on HWRB.

Games Scale	Go-Explore 10B	HWRB	Ours 1B	HWRB
Alien	959312	1	279703.5	1
Amidar	19083	0	12996.3	0
Assault	30773	1	62025.7	1
Asterix	999500	0	999999	0
Asteroids	112952	0	1106603.5	0
Atlantis	286460	0	3824506.3	0
Bank Heist	3668	0	1410	0
Battle Zone	998800	1	857369	1
Beam Rider	371723	0	457321	0
Berzerk	131417	0	35340	0
Bowling	247	0	233.1	0
Boxing	91	0	100	1
Breakout	774	0	864	1
Centipede	613815	0	728080	0
Chopper Command	996220	0	999999	1
Crazy Climber	235600	1	233090	1
Defender	N/A	N/A	995950	0
Demon Attack	239895	0	900170	0
Double Dunk	24	1	24	1
Enduro	1031	0	14332.5	1
Fishing Derby	67	0	75	1
Freeway	34	0	34	0
Frostbite	999990	1	13792.4	0
Gopher	134244	0	488900	1
Gravitar	13385	0	6372.5	0
Hero	37783	0	37545.6	0
Ice Hockey	33	0	47.53	1
Jamesbond	200810	1	623300.5	1
Kangaroo	24300	0	14372.6	0
Krull	63149	0	593679.5	1
Kung Fu Master	24320	0	1666665	1
Montezuma Revenge	24758	0	2500	0
Ms Pacman	456123	1	31403	0
Name This Game	212824	1	81473	1
Phoenix	19200	0	999999	0
Pitfall	7875	0	-1	0
Pong	21	1	21	1
Private Eye	69976	0	15100	0
Qbert	999975	0	151730	0
Riverraid	35588	0	27964.3	0
Road Runner	999900	0	999999	0
Robotank	143	1	144	1
Seaquest	539456	0	1000000	1
Skiing	-4185	0	-5903.34	0
Solaris	20306	0	10732.5	0
Space Invaders	93147	0	159999.6	0
Star Gunner	609580	1	999999	1
Surround	N/A	N/A	2.726	0
Tennis	24	1	24	1
Time Pilot	183620	1	531614	1
Tutankham	528	0	436.2	0
Up N Down	553718	1	999999	1
Venture	3074	0	2200	0
Video Pinball	999999	0	999999	0
Wizard of Wor	199900	0	118900	0
Yars Revenge	999998	0	998970	0
Zaxxon	18340	0	241570.6	1
$\sum$ HWRB		15		**24**

Table 11: Score Table of GDI-H^3 and LBC-$\mathcal{BM}$ (Ours) on HWRB.

Games	GDI-H^3	HWRB	Ours	HWRB
Scale	200M		1B	
Alien	48735	0	**279703.5**	1
Amidar	1065	0	**12996.3**	0
Assault	**97155**	1	62025.7	1
Asterix	**999999**	0	**999999**	0
Asteroids	760005	0	**1106603.5**	0
Atlantis	**3837300**	0	3824506.3	0
Bank Heist	1380	0	**1410**	0
Battle Zone	824360	1	**857369**	1
Beam Rider	422390	0	**457321**	0
Berzerk	14649	0	**35340**	0
Bowling	205.2	0	**233.1**	0
Boxing	**100**	1	**100**	1
Breakout	**864**	1	**864**	1
Centipede	195630	0	**728080**	0
Chopper Command	**999999**	1	**999999**	1
Crazy Climber	**241170**	1	233090	1
Defender	970540	0	**995950**	0
Demon Attack	787985	0	**900170**	0
Double Dunk	**24**	1	**24**	1
Enduro	14300	1	**14332.5**	1
Fishing Derby	65	0	**75**	1
Freeway	**34**	0	**34**	0
Frostbite	11330	0	**13792.4**	0
Gopher	473560	1	**488900**	1
Gravitar	5915	0	**6372.5**	0
Hero	**38225**	0	37545.6	0
Ice Hockey	47.11	0	**47.53**	1
Jamesbond	620780	1	**623300.5**	1
Kangaroo	14636	0	14372.6	0
Krull	594540	1	593679.5	1
Kung Fu Master	**1666665**	1	**1666665**	1
Montezuma Revenge	**2500**	0	**2500**	0
Ms Pacman	**11573**	0	**31403**	0
Name This Game	36296	1	**81473**	1
Phoenix	959580	0	**999999**	0
Pitfall	-4.3	0	**-1**	0
Pong	21	1	**21**	1
Private Eye	15100	0	15100	0
Qbert	28657	0	**151730**	0
Riverraid	**28349**	0	27964.3	0
Road Runner	999999	0	**999999**	0
Robotank	113.4	0	**144**	0
Seaquest	**1000000**	1	**1000000**	1
Skiing	**-6025**	0	-5903.34	0
Solaris	9105	0	**10732.5**	0
Space Invaders	154380	0	**159999.6**	0
Star Gunner	677590	1	**999999**	1
Surround	2.606	0	**2.726**	0
Tennis	**24**	1	**24**	1
Time Pilot	450810	1	**531614**	1
Tutankham	418.2	0	**436.2**	0
Up N Down	966590	1	**999999**	1
Venture	2000	0	**2200**	0
Video Pinball	978190	0	**999999**	0
Wizard of Wor	63735	0	**118900**	0
Yars Revenge	968090	0	**998970**	0
Zaxxon	216020	1	**241570.6**	1
$\sum$ HWRB		22		**24**

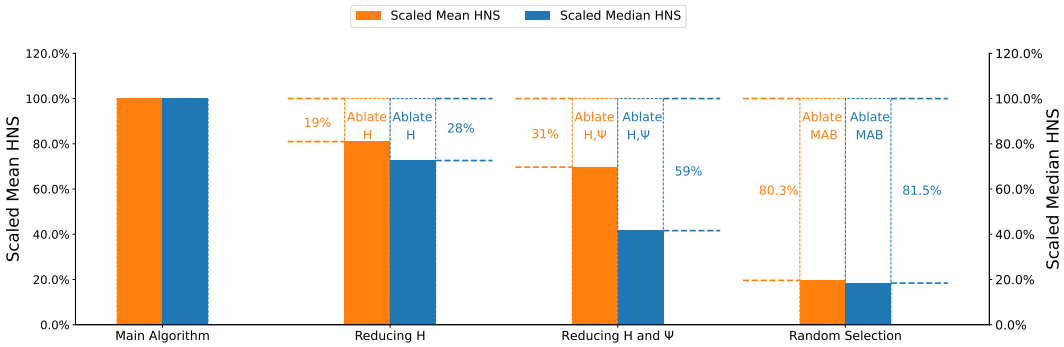


Figure 6: Ablation Results on Atari Benchmark (Machado et al., 2018). All the results are scaled by that of our main algorithm to improve readability. In these figures, we sequentially demonstrate how much performance (%) will degrade after ablating each component of LBC.

K Ablation Study

In this section, we will demonstrate the settings of our ablation studies first . Then we will introduce the algorithms of the ablation study, which has been concluded in Tab. 12. After that, we will introduce the ablation study results and case studies of t-SNE, including the results on the Atari benchmark in Fig. 6 and t-SNE analysis in App. K.3.

K.1 Ablation Study Setup

We summarized all the algorithms of the ablation study in Tab. 12. All algorithms are tested in the same experimental setup. More details on these experimental setups can see App. H. The hyper-parameters can see App. I.

Table 12: Algorithms of Ablation Study.

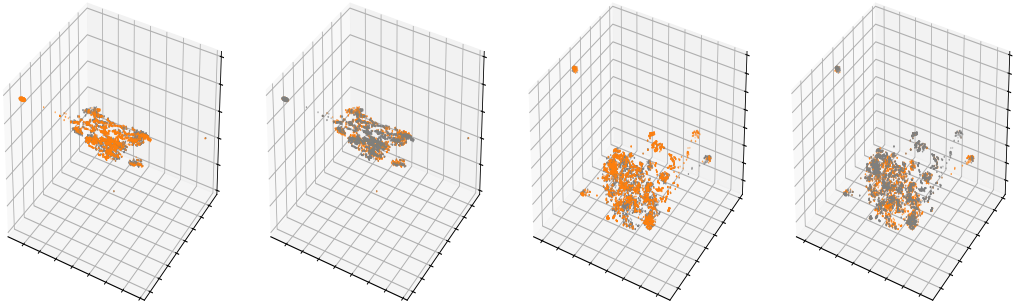
Algorithm	Main Algorithm	Reducing $\mathbf{H}$	Reducing $\mathbf{H}$ and $\mathbf{\Psi}$	Random Selection
Ablation Variables	Baseline	$\mathbf{H}$	$\mathbf{\Psi}$ and $\mathbf{H}$	Meta-Controller ($\mathbf{\Psi}$)
$\mathcal{F}_\psi$	$\sum_{i=1}^{3} \omega_i \operatorname{Softmax}_{\tau_i}(\Phi_{\mathbf{h}_i})$	$\sum_{i=1}^{3} \omega_i \operatorname{Softmax}_{\tau_i}(\Phi_{\mathbf{h}_i})$	$\operatorname{Softmax}_{\tau}(\Phi_{\mathbf{h}})$	$\sum_{i=1}^{3} \omega_i \operatorname{Softmax}_{\tau_i}(\Phi_{\mathbf{h}_i})$
$\mathbf{\Phi_H}$	$(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_3})$	$\Phi_{\mathbf{h}_1}$	$\Phi_{\mathbf{h}_1}$	$(\Phi_{\mathbf{h}_1}, ..., \Phi_{\mathbf{h}_3})$
$\mathbf{h}_i$	$(\gamma_i, \mathcal{RS}_i)$	$(\gamma_i, \mathcal{RS}_i)$	$(\gamma_i, \mathcal{RS}_i)$	$(\gamma_i, \mathcal{RS}_i)$
$\mathbf{H}$	$\{\mathbf{h}_i \vert i = 1, 2, 3\}$	$\{\mathbf{h}_i \vert i = 1\}$	$\{\mathbf{h}_i \vert i = 1\}$	$\{\mathbf{h}_i \vert i = 1, 2, 3\}$
$\boldsymbol{\psi}_i$	$(\omega_1, \tau_1, \omega_2, \tau_2, \omega_3, \tau_3)$	$(\omega_1, \tau_1, \omega_2, \tau_2, \omega_3, \tau_3)$	(τ)	$(\omega_1, \tau_1, \omega_2, \tau_2, \omega_3, \tau_3)$
$\mathbf{\Psi}$	$\{\boldsymbol{\psi}_i \vert i = 1, ..., \infty\}$	$\{\boldsymbol{\psi}_i \vert i = 1, ..., \infty\}$	$\{\boldsymbol{\psi}_i \vert i = 1, ..., \infty\}$	$\{\boldsymbol{\psi}_i \vert i = 1, ..., \infty\}$
$\vert\mathbf{H}\vert$	3	1	1	3
$\vert\mathbf{\Psi}\vert$	∞	∞	∞	∞
Category	LBC-H_3^2-$\mathbf{\Psi}_\infty^6$	LBC-H_1^2-$\mathbf{\Psi}_\infty^6$	LBC-H_1^2-$\mathbf{\Psi}_\infty^1$	LBC-H_3^2-$\mathbf{\Psi}_\infty^6$
$\vert\mathbf{M}_{\mathbf{H},\mathbf{\Psi}}\vert$	∞	∞	∞	∞
Meta-Controller ($\mathbf{\Psi}$)	MAB	MAB	MAB	Random Selection

K.2 Ablation Study Results

The ablation study results can be found in Fig. 6. From left to right, the behavior space of the first three algorithms decreases in turn, and the final performance of these three algorithms decreases in turn. We can draw the following corollary:

Corollary 5 (Smaller Behavior Space, Lower Final Performance)**.** *Given any RL methods, assuming each behavior can be visited infinitely, decreasing the behavior space and keeping other conditions unchanged will degrade the final performance of the algorithm, and vice versa.*

The behavior space of Random Selection is the same as our main algorithm. Obviously, the appropriate behaviors fail to be selected with a random selection, resulting in a great decrease of the performance in limited training frames.



(a) Main Algorithm (b) w/o UCB Item (c) Main Algorithm (d) Rule-based $\mathcal{F}$

Figure 7: Visualizing Behavior Diversity via t-SNE. (a) and (b) are drawn from the t-SNE analysis of visited states (points highlighted with ●) in Chopper Command, and (c) and (d) are drawn the t-SNE analysis of visited states in Atlantis.

K.3 T-SNE ANALYSIS

In this paper, we adopt the ucb-score to encourage the actors to try more different behavior. To demonstrate the effectiveness of the ucb-score, we conduct the t-SNE analysis of the methods removing the ucb-score in 1 and 2 of Fig. 7.

To demonstrate that the behavior diversity can be boosted by our algorithm, we conducted the t-SNE analysis of the methods with rule-based $\mathcal{F}$ in 2 and 3 of Fig. Fig. 7.

From (a) and (b) of Fig. 7, we find that removing the UCB item (i.e., $\sqrt{\frac{\log(1+\sum_{j\neq k}^{\mathrm{K}} N_{\Psi_j})}{1+N_{\Psi_k}}}$) from the optimization target of behavior selection, the behavior diversity fade away. It can prove the effectiveness of the diversity control of our methods.

From (c) and (d) of Fig. 7, we find that compared with the rule-based $\mathcal{F}$, our method can acquire a diverse set of behaviors though we do not contain a diversity-based multi-objective model training which confirms the Corollary 2.

L MODEL ARCHITECTURE

Since the network structure is not the focus of our work, we keep most of the components of the network, e.g., the LSTM Core, RL Head and Convolutional Layers the same as that of Agent57 (Badia et al., 2020a). To improve the reproducibility of our work, we still summarize our model architecture of our main algorithm in detail in Fig. 8. Wherein, $\mathbf{h}_j^t$ is the the hyper-parameters (e.g., discounted factor γ) of policy model j and $\boldsymbol{\psi}_t$ is the parameters of the constructed behavior space (e.g., ϵ in ϵ-greedy). More details on the hyper-parameters of each policy model can see App. I. $\boldsymbol{\psi}_t$ will be adaptively selected to control the behaviors across learning process via MAB. More implementation details on the MAB can see App. E. It is worth noting that our framework is not limited to an implementation of report in our body. In Fig. 8, we show a general way of integrating multiple policy models and automatically adjusting the proportion of any multiple policy models in the ensembled behavior policy.

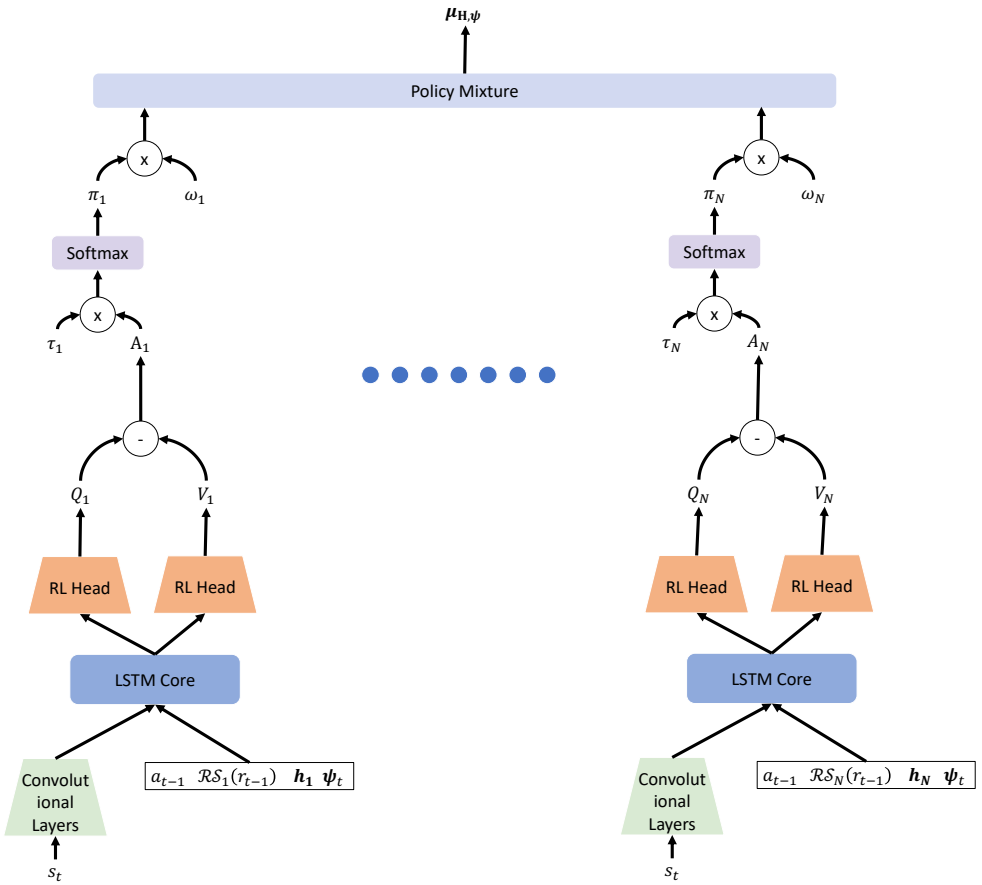


Figure 8: Model Architecture of our main algorithm.

M SUPPLEMENTARY MATERIAL

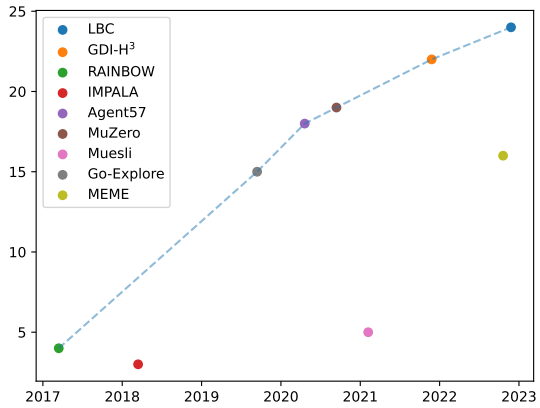


Figure 9: Human World Records Breakthrough of Atari RL Benchmarks.

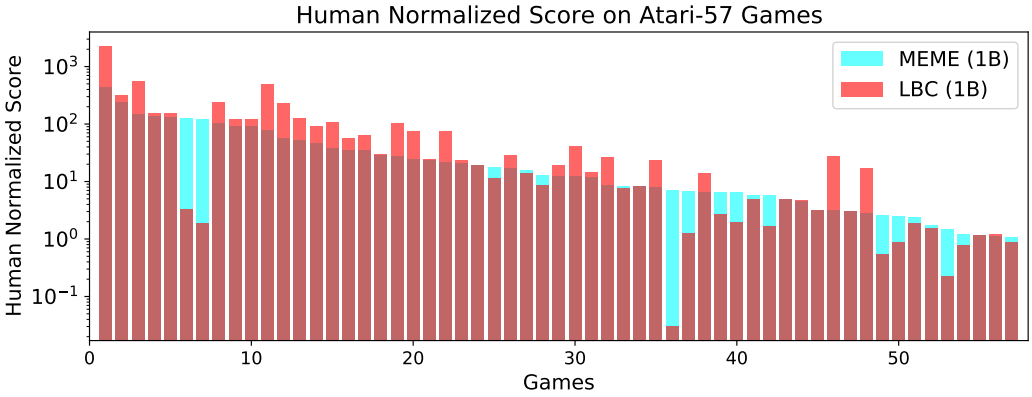


Figure 10: Performance Comparison between MEME and LBC based on HNS.(log scale)

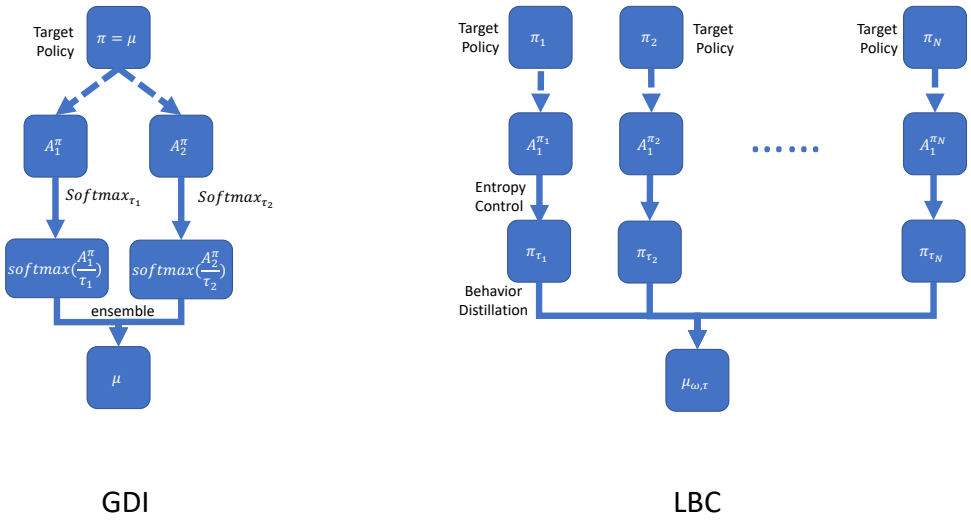
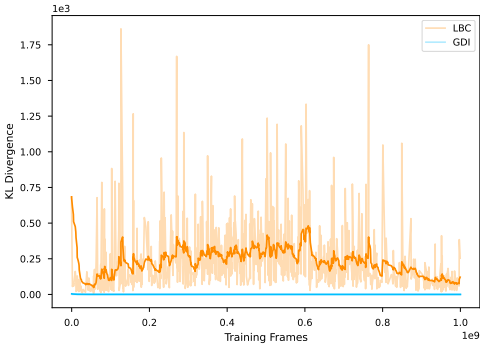


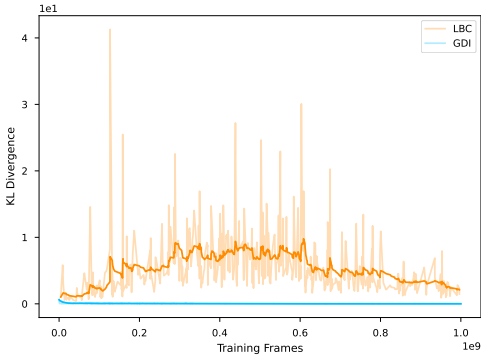
Figure 11: Different Learning Framework of GDI and LBC

N CASE STUDY: KL DIVERGENCE

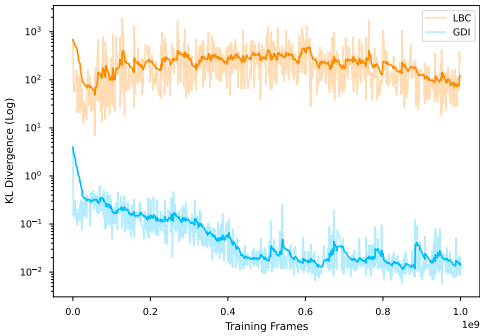
In this section, to further investigate the cause of the degradation phenomenon of the behavior space in GDI (i.e., due to a same learned policy π for A_1^π and A_2^π under different reward shaping) and demonstrate that in our behavior space (i.e., different learned policies for $A_1^{\pi_1}$ and $A_2^{\pi_2}$), the diversity can be maintained since different policy models are distinguishable across learning. For fairness, we designed two implementations to explore the degradation phenomenon in the behavior space of GDI including: i) a implementation with two different learned policies under different reward shaping (yellow in Fig. 12) ii) a implementation that learns two advantage functions of a same target policy (i.e., the behavior policy) under different reward shaping (blue in Fig. 12). The learning framework of this two implementation can be found in 11 .



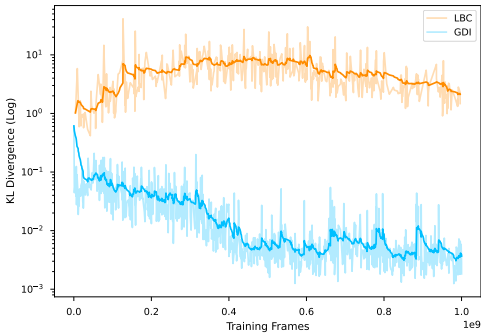
(a) $\text{Softmax}_{\tau_1}(A_1)$ and $\text{Softmax}_{\tau_2}(A_2)$



(b) $\text{Softmax}(A_1)$ and $\text{Softmax}(A_2)$



(c) $\text{Softmax}_{\tau_1}(A_1)$ and $\text{Softmax}_{\tau_2}(A_2)$



(d) $\text{Softmax}(A_1)$ and $\text{Softmax}(A_2)$

Figure 12: KL Divergence of GDI and LBC in Chopper Command (Smoothed by 0.9 for the ease of reading).

From Fig. 12, we can find the distance between $\text{Softmax}_{\tau_1}(A_1^\pi)$ and $\text{Softmax}_{\tau_2}(A_2^\pi)$ of A_1^π and A_2^π decrease rapidly in GDI while LBC can maintain a more diverse set of policies. The optional behaviors for each actor gradually diminish, and the behavior space of GDI degenerates across the learning process. In contrast, LBC can maintain the capacity of the behavior space and avoid degradation since LBC maintains a population of different policy models (Corresponding to a population of different policies.)