Bayesian Soft Actor-Critic: A Directed Acyclic Strategy Graph Based Deep Reinforcement Learning

Qi Yang∗
Computer Science and Information Systems Department,
Bradley University
Peoria, Illinois, U.S.

Ramviyas Parasuraman
Department of Computer Science, University of Georgia
Athens, Georgia, U.S.

ABSTRACT
Adopting reasonable strategies is challenging but crucial for an intelligent agent with limited resources working in hazardous, unstructured, and dynamic environments to improve the system’s utility, decrease the overall cost, and increase mission success probability. This paper proposes a novel directed acyclic strategy graph decomposition approach based on Bayesian chaining to separate an intricate policy into several simple sub-policies and organize their relationships as Bayesian strategy networks (BSN). We integrate this approach into the state-of-the-art DRL method — soft actor-critic (SAC), and build the corresponding Bayesian soft actor-critic (BSAC) model by organizing several sub-policies as a joint policy. We compare our method against the state-of-the-art deep reinforcement learning algorithms on the standard continuous control benchmarks in the OpenAI Gym environment. The results demonstrate that the promising potential of the BSAC method significantly improves training efficiency.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

KEYWORDS
Strategy; Bayesian Networks; Deep Reinforcement Learning; Soft Actor-Critic; Utility; Expectation

ACM Reference Format:
Qi Yang and Ramviyas Parasuraman. 2024. Bayesian Soft Actor-Critic: A Directed Acyclic Strategy Graph Based Deep Reinforcement Learning. In Proceedings of ACM SAC Conference (SAC’24). ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.1145/3605098.3636113

1 INTRODUCTION
A strategy is a rule used by agents to select an action to pursue goals, which is equivalent to a policy in a Markov Decision Process (MDP) [7]. It exhibits the fundamental properties of agents’ perception, reasoning, planning, decision-making, learning, problem-solving.

∗Corresponding author: Dr. Yang; Email: RickYang2014@gmail.com

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC ’24, April 8–12, 2024, Avila, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 978-8-4007-0243-3/24/04...
https://doi.org/10.1145/3605098.3636113

Figure 1: An example of the robot dog model’s Bayesian Strategy Network, showing a decomposed strategy based on action dependencies.

Figure 2: An example of the proposed BSN based implementation of the Actor-Critic DRL architecture model.

and communication in interaction with dynamic and complex environments [4, 10]. Especially in the field of real-time strategy (RTS) game [1, 14, 15] and real-world implementation scenarios like robot-aided urban search and rescue (USAR) missions [6, 12], agents need to dynamically change the strategies adapting to the current situations based on the environments and their expected utilities or needs [11, 13]. This paper first introduces the Bayesian Strategy Network (BSN) based on the Bayesian net to decompose a complex strategy or intricate behavior into several simple tactics or actions. An example of a BSN-based strategy decomposition (or action dependencies) of a robot dog is shown in Fig. 1. Furthermore, we propose a new DRL model termed Bayesian Soft Actor-Critic (BSAC), which integrates the Bayesian Strategy Networks (BSN) and the state-of-the-art SAC method. We demonstrate the effectiveness of the BSAC against the other state-of-the-art approaches on the standard continuous control benchmark domains in the OpenAI Gym environment.

2 METHODOLOGY
2.1 Bayesian Strategy Networks (BSN)
Supposing that the strategy \( T \) consists of \( m \) tactics \( (t_1, \ldots, t_m) \) and their specific relationships can be described as the BSN. We
consider the probability distribution $P_t$ as the policy for tactic $t_i$. Then, according to the Bayesian chain rule, the joint policy $\pi(a_t \in \mathcal{T}, s)$ can be described as the joint probability function (Eq. (1)) through each sub-policy $\pi_i(t_i, s)$, correspondingly. An overview of the example BSN implementation in actor-critic architecture is presented in Fig. 2.

$$\pi_F(t_1, \ldots, t_m) = \pi_i(t_1) \prod_{i=2}^{m} \pi_i(t_i|t_1, \ldots, t_i), \ m \in \mathbb{Z}^+.$$ (1)

### 2.2 Derivation of Sub-Policy Iteration

Considering that the agent interacts with an environment through a sequence of observations, strategy (action combinations), and rewards, we can describe the relationships between actions in the strategy as a BSN, represented in Eq. (1) accordingly. More formally, we can use the corresponding deep convolution neural networks to approximate the strategy Policy Network (Actor) in the Eq. (1) as Eq. (2) \footnote{Here, $\mathcal{A}$ is the joint action or strategy space for the policy $\pi$, $\theta_1$ and $a_t$ are the parameters and action space of each sub-policy network $\pi_i$.}. The Value Network (Critic) $q(s, \mathcal{A}_t; w)$ evaluate the performance of the specific joint action $\mathcal{A}$ using the value function in Eq. (3) with a parameter $w$. Then, we can calculate the corresponding parameters’ gradient descent using Eq. (4). Through this process, we decompose the strategy policy network $\pi_F$ into several sub-policies networks $\pi_i$ and organize them as the corresponding BSN. Furthermore, according to Eq. (4), each sub-policies uses the same value network to update its parameters in every iteration.

$$\pi(\mathcal{A}_t, s) \approx \pi(\mathcal{A}_t|s; \theta_1) = \prod_{i=1}^{m} \pi_i(a_i|s; \theta_i)$$ (2)

$$V(s; \theta, w) = \sum_{t \in \mathcal{T}} \pi(\mathcal{A}_t|s; \theta) \cdot q(s, \mathcal{A}_t; w)$$ (3)

$$\frac{\partial V(s; \theta, w)}{\partial \theta} = \sum_{i=1}^{m} \mathbb{E}_{\mathcal{A}_t} \left[ \frac{\partial \log \pi_i(a_i|s; \theta_i)}{\partial \theta} \right] \cdot q(s, \mathcal{A}_t; w)$$ (4)

### 2.3 Bayesian Soft Actor-Critic (BSAC)

Our method incorporates the maximum entropy concept into the actor-critic deep RL algorithm. According to the definition of conditional entropy [9], joint entropy is equal to the sum of the conditional entropies of the variables in the set, which also follows the chain rule in probability theory. For each step of soft policy iteration, the joint policy $\pi$ will calculate the value to maximize the sum of sub-systems’ $\pi_i$ entropy in the BSN using the below objective function Eq. (5). In order to simplify the problem, we assume that the weight and the corresponding temperature parameters $a_t$ for each action is the same in each sub-system.

$$J_V(\pi) = \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}_{s_t, \mathcal{A}_t} - \rho_{\pi} \left( r(s_t, \mathcal{A}_t) + \frac{a_t}{m} \sum_{i=1}^{m} \mathcal{H}(\pi_i|s_t) \right)$$ (5)

The soft Q-value can be computed iteratively, starting from any action $Q: S \times A \rightarrow \mathbb{R}$ and repeatedly applying a modified Bellman backup operator $T^\pi \footnote{Here, $Z^\text{old}(s_t)$ is the partition function to normalize the distribution.}$. In the Bayesian Soft Actor-Critic, the $T^\pi$ is given by Eq. (6). Considering that the evaluation of each sub-policy applies the same Q-value and weight, the soft state value function can be represented in Eq. (7) \footnote{We open the source of the BSAC algorithm on GitHub: https://github.com/RickYang2016/Bayesian-Soft-Actor-Critic--BSAC.}.

$$T^\pi Q(s_t, \mathcal{A}_t) \approx r(s_t, \mathcal{A}_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho} [V(s_{t+1})]$$ (6)

$$V(s_t) = \mathbb{E}_{\mathcal{A}_t \sim \pi} \left[ Q(s_t, \mathcal{A}_t) - \frac{1}{m} \sum_{i=1}^{m} \log \pi_i(a_i|s_t) \right]$$ (7)

Specifically, in each sub-policy $\pi_i$ improvement step, for each state, we update the corresponding policy according to Eq. (8).

$$\pi_{new} = \arg \min_{\pi' \in \Pi} \mathcal{D}_{KL} \left( \frac{1}{m} \sum_{i=1}^{m} \pi'_i(s_t|\cdot) \exp(Q^\text{old}(s_t, \cdot)) \right) Z^\text{old}(s_t)$$ (8)

### 3 EXPERIMENTS AND RESULTS

In our experiments, we use three of the standard continuous control benchmark domains – Hopper-v2, Walker2d-v2, and Humanoid-v2 in the OpenAI’s Gym environment. We study the performance of the proposed BSAC against the state-of-the-art continuous control algorithm, the SAC [3] and other benchmark DRL algorithms, PPO [8], DDPG [5], and TD3 [2].

In the corresponding experiments, we decompose hopper, walker2d, and humanoid behaviors into different BSNs. Such as three policies (3P) BSN (hip action $t_1$, knee action $t_2$, and ankle action $t_3$) in Hopper-v2 (Fig. 3(a)), five policies (5P) BSN (hip action $t_1$, left knee action $t_2$, right knee action $t_3$, left ankle action $t_4$, and right ankle action $t_5$) in Walker2d-v2 (Fig. 3(b)), and five policies (5P) BSN (abdomen action $t_1$, the actions $t_2$ of right hip and right knee, the actions $t_3$ of left hip and knee, the actions $t_4$ of right shoulder and right elbow, and actions $t_5$ of left shoulder and left elbow) in Humanoid-v2 (Fig. 3(c)). Furthermore, for corresponding BSAC models, we can formalize their joint policy (action) as Eq. (9), Eq. (10), and Eq. (11).

$$P(t_1, t_2, t_3) = P(t_1)P(t_2|t_1)P(t_3|t_2)$$ (9)

$$P(t_1, t_2, t_3, t_4, t_5) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)P(t_5|t_4)$$ (10)

$$P(t_1, t_2, t_3, t_4, t_5) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)P(t_5|t_4)$$ (11)

Comparing the performance of the BSAC with SAC, TD3, DDPG, and PPO in Hopper-v2 (Fig. 4(a)), Walker2d-v2 (Fig. 4(b)), and Humanoid-v2 (Fig. 4(c)), we prove that BSAC can achieve higher performance than other DRL algorithms. Furthermore, with the increasing complexity of the agent’s behaviors and strategy, decomposing the complex behaviors into simple actions or tactics and organizing them as a suitable BSN, building the corresponding joint policy model in the BSAC can substantially increase training efficiency\footnote{Comparing with Different BSN Models. This section analyzes two other BSN models corresponding to different action decomposition methods in the Humanoid-v2 domain. In Fig. 3(d), for the BSAC three sub-policies model (BSAC-3P), it generates the distribution of the abandon actions $P(t_1)$, the distribution $P(t_2|t_1)$ of the actions of shoulder and elbow, and the distribution $P(t_3|t_1)$ of the actions of hip and knee, respectively. Within a sub-space, the actions are independent. For example, we do not consider the conditional}.
dependence between the actions of the shoulder and elbow, but merging them as one joint action (t2) depends on the abdomen joint (t1). And the same situation is in the joints (t3) of the hip and knee actions. The results shown in Fig. 4(d) demonstrate that all BSAC models can achieve higher performance than the SAC. On the other hand, compared to the BSAC-3P models’ performance, the BSAC-9P presents more advantages than BSAC-5P and BSAC-3P, and the five sub-policies model presents the worst performance among them. It implies that the joint policy distribution designed in the BSAC-9P model is more similar to the Q-value distribution than the other BSAC models and describes the more reasonable relationships among those actions in the current reward mechanism.

4 CONCLUSIONS
We introduce a novel directed acyclic strategy graph termed Bayesian Strategy Network (BSN) to achieve deep reinforcement learning (DRL) sample efficiency and improve training speed. Based on the Soft Actor-Critic (SAC) algorithm, we propose a new DRL model termed the Bayesian Soft Actor-Critic (BSAC), which integrates other BSAC models and describes the more reasonable relationships among those actions in the current reward mechanism.

REFERENCES
[1] Michael Buro. 2003. Real-time strategy games: A new AI research challenge. In IJCAI, Vol. 2003. 1534–1535.
[2] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In International conference on machine learning. PMLR, 1587–1596.
[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning. PMLR, 1864–1870.
[4] Pat Langley, John E. Laird, and Seth Rogers. 2009. Cognitive architectures: Research issues and challenges. Cognitive Systems Research 10, 2 (2009), 141–160.
[5] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015).
[6] Robin R Murphy. 2014. Disaster robotics. MIT press.
[7] Yara Rizk, Mariette Awad, and Edward W Tunstel. 2018. Decision making in multiagent systems: A survey. IEEE Transactions on Cognitive and Developmental Systems 10, 3 (2018), 544–559.
[8] John Schulman, Wolski, et al. 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).
[9] Alfred Wehrl. 1978. General properties of entropy. Reviews of Modern Physics 50, 2 (1978), 221.
[10] Qin Yang, Zhiwei Luo, Wenzhan Song, and Ramviyas Parasuraman. 2019. Self-reactive planning of multi-robots with dynamic task assignments. In 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE, 89–91.
[11] Qin Yang and Ramviyas Parasuraman. 2020. Hierarchical needs based self-adaptive framework for cooperative multi-robot system. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2991–2998.
[12] Qin Yang and Ramviyas Parasuraman. 2020. Needs-driven heterogeneous multi-robot cooperation in rescue missions. In 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 252–259.
[13] Qin Yang and Ramviyas Parasuraman. 2021. How Can Robots Trust Each Other For Better Cooperation? A Relative Needs Entropy Based Robot-Robot Trust Assessment Model. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2656–2663.
[14] Qin Yang and Ramviyas Parasuraman. 2022. Game-theoretic utility tree for multi-robot cooperative pursuit strategy. In ISR Europe 2022; 54th International Symposium on Robotics. VDE, 1–7.
[15] Qin Yang and Ramviyas Parasuraman. 2023. A Hierarchical Game-Theoretic Decision-Making for Cooperative Multiagent Systems Under the Presence of Adversarial Agents. In Proceedings of the 30th ACM SIGAPP Symposium on Applied Computing. 771–782.