EFFICIENT AND STABLE INFORMATION DIRECTED EXPLORATION FOR CONTINUOUS REINFORCEMENT LEARNING

Mingzhe Chen[1], Xi Xiao[1], Wanpeng Zhang[1], Xiaotian Gao[2,]*

[1]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]Microsoft Research Asia, Beijing, China

ABSTRACT

In this paper, we investigate the exploration-exploitation dilemma of reinforcement learning algorithms. We adapt the information directed sampling, an exploration framework that measures the information gain of a policy, to the continuous reinforcement learning. To stabilize the off-policy learning process and further improve the sample efficiency, we propose to use a randomized learning target and to dynamically adjust the update-to-data ratio for different parts of the neural network model. Experiments show that our approach significantly improves over existing methods and successfully completes tasks with highly sparse reward signals.

***Index Terms*—** Reinforcement Learning, Exploration, Policy

1. INTRODUCTION

In recent years, deep reinforcement learning has enjoyed great success in solving sequential decision problems in various domains such as board games[1], video games[2, 3], and continuous control problems[4, 5, 6]. Successes in these areas rely on efficient and accurate simulators, as deep reinforcement learning algorithms are known for their sample *inefficiency*. A key problem that affects the efficiency of reinforcement learning algorithms is the *dilemma of exploration and exploitation*.

In reinforcement learning, the problem is represented as a Markov Decision Process (MDP) that is unknown to the agent. Thus at each time step, the current policy learned from the limited data previously observed by the agent can be suboptimal. The agent needs to balance between following the current policy and sacrificing the instantaneous rewards to investigate the poorly understood states for potentially greater rewards. Common exploration strategies, such as ϵ-greedy for discrete control and noise injection for continuous control, choose exploratory actions based on random perturbations of the agent's current policy. These exploration methods are undirected and can lead to exponentially slower learning[7].

A popular approach for directed exploration is to utilize the uncertainty information in the agent's model. For example, the upper confidence bound (UCB)[8] algorithm sets the confidence interval of the estimate of rewards as an additional bonus to promote exploration. Thompson Sampling (TS)[9] is a related algorithm that chooses the policies according to their probability of being optimal in the Bayesian posterior distribution[10].

To extend these algorithms to reinforcement learning, a major challenge is to obtain the uncertainty measures on the MDPs. One approach is to maintain the Bayesian posterior distribution on the value function rather than on the MDP[11]. Based on this idea, several RL algorithms with UCB[12] and TS[13] exploration have been developed. Bootstrapped DQN [13] maintains an ensemble of Q-functions. At the start of each episode, Bootstrapped DQN randomly samples a Q-function from the ensemble and acts greedily w.r.t. the sample in that episode. However, for problems with more complex information structures, it is possible to explore much more efficiently than UCB or TS. A framework called information-directed sampling (IDS)[14, 15] has been proposed for bandit problems. The IDS framework explicitly measures the reduction of uncertainty caused by taking an action, i.e., the information provided by that action.

IDS is later introduced into reinforcement learning[16], but remains computationally intractable for practical problems with large state space, because it requires to maintain the posterior distribution of the dynamics model and to plan on a sampled set of models for the optimal policy at the start of every episode. A deterministic approximation of IDS has been proposed for discrete control problems with finite action space[17].

In this paper, we extend the core idea of the IDS framework to continuous reinforcement learning. For continuous reinforcement learning with infinite action space, policies are often modeled by neural networks in the actor-critic style. Since the training of the policy relies on the value function and both are modeled by deep neural networks, the distribution shift caused by exploratory behavior may make the learning process divergent. To mitigate this problem, we propose to dynamically adjust the update-to-data (UTD) ratio based on the uncertainty of the agent. Model-free off-policy reinforcement learning algorithms often employ a UTD ratio of 1, which means that the agent is updated roughly the same number of times as it interacts with the environment. Chen

*Corresponding Author: Xiaotian Gao(Xiaotian.Gao@microsoft.com)

et al.[18] recently achieve significant empirical results by employing a higher UTD ratio. However, a constant high UTD ratio may cause the agents converge too quickly, and failed to explore in the sparse environments.

Our algorithm is based on the combination of IDS framework and dynamical UTD ratio. Inspired by recent advancements on Bayesian reinforcement learning, our method is able to bypass the computational intractability of planning on dynamics models. To stabilize the off-policy learning process of neural networks and further improve the sample efficiency, we also utilize the ensemble to generate a randomized learning target and employ a dynamical UTD ratio based on the uncertainty estimate.

We evaluate our algorithm on the continuous control benchmarks of Mujoco simulated robots[19, 20]. The evaluation demonstrates that our method substantially outperforms SAC, and a recently proposed ensemble-based algorithm called SUNRISE[21], which uses UCB as exploration strategy [22]. We further test our algorithm on a modified version of the tasks, which have sparsified reward functions. The results showed that our algorithm is the only one that successfully completes the tasks with highly sparse reward signals.

2. PRELIMINARIES

2.1. Reinforcement Learning

We model the agent-environment interaction with an MDP $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $R(s,a)$ the unknown reward distribution, $P(s'|s,a)$ the unknown transition distribution, and $\gamma \in [0,1)$ the discount factor. A policy $\pi(\cdot|s) \in P(\mathcal{A})$ maps a state $s \in \mathcal{S}$ to a distribution over action space. Formally, the agent receives a state s_t from the environment and chooses an action a_t at each time step t according to its policy π, then the environment returns the next state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and the reward $r_t \sim R(s_t, a_t)$ for this transition. For a fixed policy π, the discounted cumulative rewards of action a in state s is a random variable $Z^\pi(s,a) := \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$, where $a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$. The primary objective of RL algorithms is to find an optimal policy π that maximizes the expected discounted rewards $Q(s,a) = \mathbb{E}[Z^\pi(s,a)]$.

2.2. Information Directed Exploration

We will briefly review the core ideas of IDS principles under the bandit problem for brevity, as classical bandit problem can be considered as a single-state MDP. Denoting the MDP as M, the agent's action at each episode as a_t and the realization of the stochastic reward function as r_t, we have the filtration as $\mathcal{F}_t$, which is intuitively the "history" of action-return samples $\{(a_t, r_t)\}$ observed by the agent up to episode $(t-1)$. The expected instantaneous return of an action $a \in \mathcal{A}$ at episode t is defined as: $\mathbb{E}_M[R_a|\mathcal{F}_t]$. We denote the optimal action by $a^* \in \arg\max_{a \in \mathcal{A}} \mathbb{E}[R_a|\mathcal{F}_t]$. Note that it is also a random variable and its posterior distribution is updated according to the Bayesian rule. With some abuse of notation, we define the instantaneous regret of the action a as the difference between the maximum expected instantaneous return and the expected instantaneous return of the action a, which is $\Delta_t(a) = \mathbb{E}_{a^*}[\mathbb{E}_M[R_{a^*} - R_a|\mathcal{F}_t, a^*]]$. The regret of a policy is defined as the instantaneous regrets summing up to time T: $\sum_{t=1}^{T} \mathbb{E}_{\mathcal{F}_t}[\Delta_t(a_t)]$.

In the IDS framework, in order to quantify the information gained by the agent by performing certain actions, we also need to define the information gain function $I_t(a)$. The information ratio of a policy is defined as the ratio of the regret and information gain:

$$\Psi_t(\pi) : \mathcal{P}(\mathcal{A}) \to \mathbb{R}_{\geq 0},\ \pi \mapsto \frac{\mathbb{E}_\pi[\Delta_t(a)^2|\mathcal{F}_t]}{\mathbb{E}_\pi[I_t(a)|\mathcal{F}_t]}. \quad (1)$$

At each time step t, the IDS exploration policy is determined by:

$$\pi_t^{IDS} \in \arg\min_{\pi \in \mathcal{P}(\mathcal{A})} \Psi_t(\pi). \quad (2)$$

We note that the existing Bayesian framework of IDS was established for the discrete action space. Fortunately, following a similar treatment as in the frequentist IDS framework[15], it can be naturally extended to the case of continuous action space.

3. PROPOSED METHOD

The intractability of IDS in continuous RL can be attributed to the following aspects:

- Maintain the posterior distribution of MDPs
- Compute regret and information gain over the posterior
- Minimize information ratio over the infinite set of policies

Since maintaining the posterior distribution for large MDPs is difficult, we choose the alternative to track the distribution of the optimal value functions. Specifically, we use a bootstrapped ensemble as an approximation for the posterior. With the optimal policy determined by the value function, dynamic programming over the MDP is bypassed. To tackle the continuous action space, we choose to use SAC[23] as the base algorithm to form an ensemble.

We denote the ensemble as $\{(\pi_n, Q_n)\}_{n=1}^N$. For each policy network π_n the input is a state s and output a stochastic policy, and for value network Q_n the input is a pair of state-action (s,a) and output is an estimate of the action value. The neural network parameters are not shared between each pair of actor-critic in our implementation. The set of all critics constitutes an approximation of the posterior distribution.

With the approximated posterior of the value function, we can compute the estimates of instantaneous regret and information gain. The notion of instantaneous regret can be naturally extended to reinforcement learning as the difference between the maximum action-value and the action-value for a: $\Delta_t(s,a) = \mathbb{E}_{a^*}[\mathbb{E}_M[Q(s,a^*) - Q(s,a)|\mathcal{F}_t, a^*]$.

Specifically, given an ensemble of actor-critics $\{(\pi_n, Q_n)\}_{n=1}^N$, we can compute the empirical mean and variance of action-value as:

$$\begin{aligned}\hat{\mu}(s,a) &= \frac{1}{N}\sum_{n=1}^{N} Q_n(s,a), \\ \hat{\sigma}^2(s,a) &= \frac{1}{N}\sum_{n=1}^{N}(Q_n(s,a) - \mu(s,a))^2.\end{aligned} \tag{3}$$

The computation of regret relies on the posterior distribution of the optimal action a^*, which can also be estimated with the approximated posterior. In this paper, we use a conservative surrogate function of the instantaneous regret based on the empirical mean and variance:

$$\begin{aligned}\hat{\Delta}_t(s,a) = \max_{a' \in \mathcal{A}} &(\hat{\mu}_t(s,a') + \lambda_t \hat{\sigma}_t(s,a')) \\ &-(\hat{\mu}_t(s,a) - \lambda_t \hat{\sigma}_t(s,a))\end{aligned} \tag{4}$$

where λ_t is a scaling hyperparameter. Intuitively, the first term corresponds to the stochastically plausible maximum value at the given state, while the second term provides a heuristic lower bound for the chosen action.

The notion of information gain can be extended to RL in a similar manner. In this paper, we define the information gain function as

$$I_t(s,a) := \log(1 + \sigma_t^2(s,a)) \tag{5}$$

which prioritizes actions that carry more uncertainty.

Finally, we reformulate the empirical information ratio for reinforcement learning as

$$\hat{\Psi}_t(\pi, s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[\hat{\Delta}_t(s,a)^2 - \eta I_t(s,a)] \tag{6}$$

where we add a hyperparameter η to adjust the behavior policy's sensitivity to the information gain. As $\eta \to 0$, it will reduce to a conservative exploration policy and vice versa.

In theory, the IDS framework guarantees that the optimal exploration policy obtained from Eq (2) has a support set of size no more than two[15, 14] . However, in order to obtain such an optimal policy, we need to minimize over the set of all distributions.

In our algorithm, we utilize the ensemble of policy networks to generate a candidate set of deterministic exploration policies. Since the training of the policy ensemble holds the same elements as the training of the Q ensemble, i.e., bootstrapped data, distinct objectives, and a stochastic training process, we argue that the policy ensemble can be considered as an approximated distribution of the optimal policy function.

Algorithm 1: IDS for continuous RL

Input: The ensemble $\{(\pi_n, Q_n)\}_{n=1}^N$, candidates size K

for *each time step* t **do**
  for *each* π *in* $\{\pi_n\}_{n=1}^N$ **do**
    Sampling candidate deterministic policies $\{a_i\}_{i=1}^k$ according to $\pi(\cdot|s)$;
  end
  for *each* a_i *in* $\{a_i\}_{i=1}^K$ **do**
    Compute $\hat{\mu}(s,a_i)$ and $\hat{\sigma}^2(s,a_i)$ according to (3);
    Compute $\hat{\Delta}(s,a_i)$ according to (4);
    Compute $\hat{I}(s,a_i)$ according to (5);
    Compute $\hat{\Psi}(s,a_i)$ according to (6);
  end
  Execute the action that minimizes $\hat{\Psi}(s,a)$.
end

As the policies of SAC are inherently stochastic, we can scale the number of candidate policies by generating $k \geq 1$ actions for each π_n. We denote the set of candidate policies proposed by the policy ensemble as $\{a_n\}_{n=1}^K$, where $K := k \times N$.

3.1. Uncertainty based UTD ratio

In our algorithm, exploratory policies are determined by the IDS framework, and the significant distribution shift between the behavior policy and target policy can lead the training to diverge. Moreover, since candidate policies are generated by the policy networks, inconsistency between the policy and value networks can affect the quality of exploratory policies.

Inspired by REDQ[18], we use a subset of the ensemble to generate a conservative target for each Q_i to stabilize training:

$$y_i(r, s') := r + \gamma \min_{j\in\mathcal{M}}(Q^j(s', a') - \log \pi^j(a'|s'))$$

where $\mathcal{M}$ is a uniformly random subset from the ensemble and $a' \sim \pi_j(a|s')$ is a sampled action.

With conservative targets to stabilize training, we can use a high UTD during the training of the algorithm. However, a conservative target and a constant high UTD ratio may cause the agent to converge too quickly to sufficiently explore the environment. We propose to dynamically adjust the UTD ratio based on the variance of the targets . Specifically, at every training step of the value networks, the agent will stop according to the Bernoulli distribution Bern(p), where $p =$ Sigmoid$(-\hat{\sigma}^2(s,a)) + \frac{1}{2}$. The update of the policy network occurs when the value networks stop. Intuitively, the UTD ratio will be higher when the empirical variance is large to

speed up training and improve sample efficiency, and lower when the empirical variance is small to avoid overfitting and help exploration.

4. EXPERIMENTS

We evaluate our algorithm on Mujoco environments [19] through the OpenAI Gym[20] interface. Three challenging environments, Hopper, Ant and Walker2d are selected.

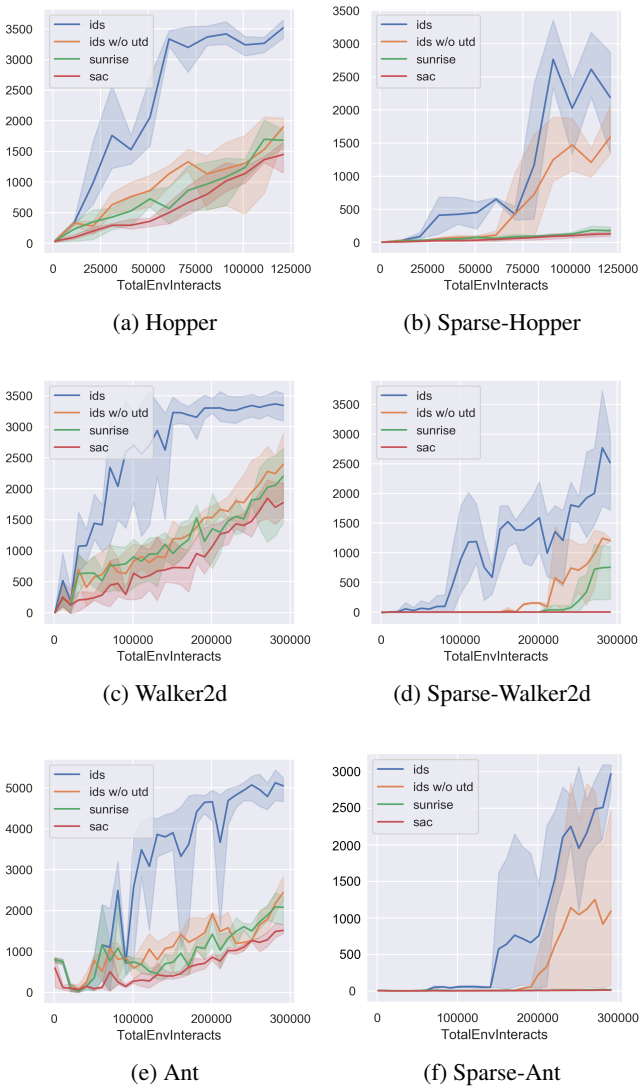
We use the same hyperparameters in vanilla SAC[23] in the ensemble for all experiments. In evaluation mode, the action is chosen greedily with regard to the mean action value, i.e. $a = \arg\max \sum_n Q_n(s, a')/N$. We report the performance every 10K steps. Three additional hyperparameters are added for empirical regret estimation, including the ensemble size N, the scaling λ and the bootstrapped sampling proportion β. We find that by setting $\beta = 1$ the agent consistently achieves better performance, which means that the models in the ensemble are trained on all samples collected by the agent. We argue that the stochasticity in initialization and training of neural network can induce enough diversity in the ensemble, while setting $\beta = 1$ allows each network in the ensemble to train on more data. We also evaluate the impact of different ensemble sizes N on the dense environment. The performance improves by increasing the ensemble size and saturates when the ensemble size reaches 8.

We compare our algorithm to the a recently proposed ensemble algorithm SUNRISE that uses UCB to explore and further utilizes the parametric uncertainty to stabilize the training process[21]. We also evaluate our algorithm on a sparsified version of the environments, where the rewards are set to zero when they are lower than a certain threshold τ, which requires the algorithms to actively explore to learn non-trivial behavior. To demonstrate the effectiveness of the IDS method, we designed the threshold value for each environment separately, so that the agent would face a $\sim 90\%$ sparse environment at initialization. We report the mean and standard deviation of averaged returns over four runs in Fig1. For baselines, we use the implementation in SUNRISE[21].

Experiments show that our algorithm improves significantly over SAC and outperforms the existing efficient RL methods in all three environments. In the absence of dynamic UTD ratio, the sampling efficiency is still a little better than SUNRISE in the dense environment. However, in the sparse environment, only the IDS-based algorithm successfully explores all three environments, as shown in Fig1, while the absence of dynamic UTD slows down the learning speed. The results suggest that combining the dynamic UTD ratio can significantly boost the proposed IDS method.

5. CONCLUSIONS

In this paper, we extend the idea of information-directed sampling to the setting of continuous reinforcement learning. In combination with uncertainty-based dynamic UTD ratios, the resulting approach significantly outperforms existing methods and enables exploration in highly sparse environments.



(a) Hopper (b) Sparse-Hopper

(c) Walker2d (d) Sparse-Walker2d

(e) Ant (f) Sparse-Ant

Fig. 1: Learning curves on the original and sparsified environments. The solid line and shaded regions represent the mean and standard deviation across four runs.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (61972219), the Research and Development Program of Shenzhen (JCYJ20190813174403598, SGDX20190918101201696), the National Key Research and Development Program of China (2018YFB1800601), and the Overseas Research Cooperation Fund of Tsinghua Shenzhen International Graduate School (HW2021013).

6. REFERENCES

[1] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al., "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[2] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al., "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[3] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al., "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

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

[5] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[6] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[7] Tze Leung Lai and Herbert Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[8] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[9] William R Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[10] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen, "A tutorial on Thompson sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018.

[11] Ian Osband, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen, *Deep exploration via randomized value functions*, Ph.D. thesis, Stanford, 2017.

[12] Richard Y. Chen, Szymon Sidor, Pieter Abbeel, and John Schulman, "UCB exploration via q-ensembles," *arXiv*, 2017.

[13] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy, "Deep exploration via bootstrapped DQN," *Advances in Neural Information Processing Systems*, pp. 4033–4041, 2016.

[14] Daniel Russo and Benjamin Van Roy, "Learning to optimize via information-directed sampling," *Operations Research*, vol. 66, no. 1, pp. 230–252, 2018.

[15] Johannes Kirschner and Andreas Krause, "Information directed sampling and bandits with heteroscedastic noise," in *Conference On Learning Theory*. PMLR, 2018, pp. 358–384.

[16] Andrea Zanette and Rahul Sarkar, "Information directed reinforcement learning," Tech. Rep., Technical report, Technical report, 2017.

[17] Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause, "Information-directed exploration for deep reinforcement learning," *arXiv preprint arXiv:1812.07544*, 2018.

[18] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross, "Randomized ensembled double q-learning: Learning fast without a model," *arXiv preprint arXiv:2101.05982*, 2021.

[19] Emanuel Todorov, Tom Erez, and Yuval Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[20] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[21] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel, "Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning," *arXiv preprint arXiv:2007.04938*, 2020.

[22] Tingwu Wang and Jimmy Ba, "Exploring model-based planning with policy networks," *arXiv preprint arXiv:1906.08649*, 2019.

[23] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.