Active Gamma-Ray Log Pattern Localization With Distributionally Robust Reinforcement Learning

Yuan Zi, *Student Member, IEEE*, Lei Fan, *Senior Member, IEEE*, Xuqing Wu, *Member, IEEE*, Jiefu Chen, *Senior Member, IEEE*, Shirui Wang, *Student Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*— Accurately localizing 1-D signal patterns, such as Gamma-ray well-log depth matching, is crucial in the oilfield service industry as it directly affects the quality of oil and gas exploration. However, traditional methods such as well-log curve analysis and pattern hand-picking matching are labor-intensive and heavily rely on human expertise, leading to inconsistent results. Although attempts have been made to automate this process, challenges such as low computational performance, non-robustness, and nongeneralization remain unsolved. To address these challenges, we have developed a data-driven AI system that learns an active signal pattern localization strategy inspired by human attention. Our artificial intelligence system uses an offline reinforcement learning (RL) framework as its central component, which solves a highly abstracted Markov decision process (MDP) problem via offline training on human-labeled historical data. The RL agent uses top-down reasoning to determine the location of target signal fragments by deforming a bounding window using simple transformation actions. To overcome distribution shifts between logged data and real and ensure generalization, we propose a discrete distributionally robust soft actor-critic (SAC) RL framework (DRSAC-Discrete) to solve the MDP problem under uncertainty. By exploring unfamiliar environments in a restrictive manner, the DRSAC-Discrete algorithm provides a safe solution that can be used when data is limited during the early stage of this industrial application. We evaluated the RL-based localization system on augmented field Gamma-ray well-log datasets, and the results showed promising localization capability. Furthermore, the DRSAC-Discrete algorithm demonstrated relatively robust performance guarantees when facing data shortage.**

***Index Terms*— Distributionally robustness, gamma-ray log, Markov decision processes (MDPs), signal localization.**

NOMENCLATURE

$s \in \mathcal{S}$	States.
s'	Next state after taking action.
$a \in \mathcal{A}$	Actions.
γ	Discount factor.
$P(s', r\|s, a)$	Transition probability from current state to next state with action a and reward r.
S_t, A_t, R_t	State, action and reward at time step t of one trajectory.
$\pi(a\|s)$	Stochastic policy (agent behavior strategy); $\pi_\theta(.)$ is a policy parameterized by θ.
$x_{e,i}$	Indicator for location i that first detects event e
$V(s)$	State-value function measuring the expected return of state s.
$V_\theta(.)$	Value function with parameter θ.
$V^\pi(s)$	Value function $V^\pi(s) = \mathbb{E}_{a\ \pi}[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\|S_t = s]$.
$Q(s, a)$	Action-value function.
$Q_\theta(.)$	Q value function with parameter θ.
$Q^\pi(s, a)$	Value of (state, action) pair under a policy π; $Q^\pi(s,a) = \mathbb{E}_{a\ \pi}[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\|S_t = s, A_t = a]$.
θ	Weights of neural network.
$\mathcal{D}$	Data buffer.
$\mathbb{N}$	Normal distribution.
$\mathcal{H}$	Entropy measure.
α	Temperature parameter to control how important the entropy term is.
κ	Action step ratio.
x_1	Left end of the sequence.
x_2	Right end of the sequence.
D	Rescale operator.
w	Represents the coordinates of the observation window.
η	Stop reward.
τ	Localization threshold.
$\mathcal{T}$	Bellman operator.
M	Number of actions
$\mathcal{U}_\epsilon(\pi)$	Uncertainty set.
$\epsilon(s)$	Uncertainty ball's radius.
C and ξ	Hyperparameters control the size of the uncertainty set.
$n_t(s)$	Visiting times of state s
Ω^*	Approximated Fenchel conjugate of the KL-divergence-based regularized Bellman operator.
λ	Regularization parameter.
μ	Action distribution.

Manuscript received 16 December 2022; revised 5 April 2023; accepted 1 May 2023. Date of publication 30 May 2023; date of current version 7 June 2023. This work was supported in part by NSF under Grant CNS-2107216, Grant CNS-2128368, Grant CMMI-2222810, and Grant NSF-EPCN-2045978; in part by the U.S. Department of Transportation; in part by Toyota; and in part by Amazon. *(Corresponding author: Jiefu Chen.)*

Yuan Zi, Jiefu Chen, and Shirui Wang are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA (e-mail: jchen82@central.uh.edu).

Lei Fan is with the Department of Engineering Technology, University of Houston, Houston, TX 77004 USA.

Xuqing Wu is with the Department of Information and Logistics Technology, University of Houston, Houston, TX 77004 USA.

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446701, South Korea.

Digital Object Identifier 10.1109/TGRS.2023.3278491

I. Introduction

GEOPHYSICAL well-logging interpretation is an essential technique to identify subsurface properties [1]. This information guides reservoir identification, drilling, and development monitoring. The Gamma-ray log is one of many well-log measurements that shows distinct disruptions in physical parameters of subsurface strata. Aligning Gamma-ray data to well depths is crucial by recognizing signal fragments with similar patterns from similar geological formations.

Traditionally, depth alignment is done manually by picking visual patterns within log curves based on human expertise. However, this method is time-consuming. Automatic signal pattern localization has been discussed for a decade, but certain hurdles need to be overcome to implement it, such as follows.

1) *Complex Patterns:* Abstract patterns associated with unknown subsurface conditions are too complex for simple threshold-based methods to classify and locate. The rocks being inspected vary in curvature, size, shape, and material, with different physical characteristics. Typically, experts rely on geological and physical priors to analyze the entire signal curve and the relationship between peaky and concave shapes to determine which signal indicates which rock. The absolute value of the measurement is not meaningful for analysis; instead, the relative value and pattern of value change provide information for human analysis.
2) *Blur, Shift, and Noise:* Measuring signals while inspecting the characteristics of rocks at different depths in a well is challenging due to the motion and extreme subsurface environments, such as high temperatures, which can cause vibrations and noise. These factors make it difficult to accurately profile the subsurface. Traditional automatic localization techniques, like cross-correlations, are not suitable for practical projects due to these challenges.
3) *Generalization:* A generalized localization process is essential to fit any oil field environment. Cross correlation methods delicately designed for one oil field may fail in another location.

Several methods have been proposed to assist in matching signal patterns. While cross correlation methods, such as those described in [2], [3] have provided a feasible way to address simple tasks, they are sensitive to perturbations such as distortion and noise. This sensitivity prevents their application to geophysical signals, which commonly exhibit distortion factors such as shifting, deformation, and missing data. In contrast, dynamic time warping (DTW) [3] and [4] provide a distortion-tolerant approach for measuring similarity between two temporal sequences of different lengths. However, DTW has restrictions and high computational costs, making it unsuitable for matching field geophysical signal patterns, such as those observed in Gamma-ray logs.

Researchers are using data-driven machine learning to overcome the challenges mentioned above. Promising applications [5], [6] have been developed, and some attempts [7], [8] have been made to address the Gamma-ray signal matching problem using supervised learning. These methods use neural networks to predict how to move a fixed-size sliding window to mimic the way humans perform manual depth matching of two Gamma-ray logs. However, all of these methods have limitations, such as assuming that the signal is partially located inside the initial window. The problem of pattern localization is also a popular and challenging topic in the computer vision community. They have used reinforcement learning (RL) [9] to address image pattern localization, inspired by a human attention mimic object [10].

We propose a new method for Gamma-ray pattern localization inspired by previous works mentioned above. Our approach uses state-of-the-art safe RL to create a pattern-specific active detection model that learns to localize the target signal object using RL [11], [12] and distributionally robust RL [13]. Our model follows a top-down search strategy, applying a sequence of transformations to progressively reduce the window size while keeping the target signal pattern inside and minimizing background noise. This dynamic attention-action strategy is based on the observation that humans use a global-to-local attention pattern for localization. Our method formulates the 1-D signal pattern localization problem as a Markov decision process (MDP) that maximizes information value at each search step. Unlike traditional correlation and supervised learning methods with fixed windows, our approach searches regions using a high-level reasoning strategy that processes global features first and adapts to local features at the end of searching, reducing redundant searching actions. Each search action guides the next step and provides significant guidance for the following search actions.

Modern RL algorithms are capable of solving abstract decision-making problems, but their performance heavily relies on the quality of training data. However, the agent's solutions can be vulnerable if it encounters a new or shifted environment or dataset. This is due to two types of uncertainty: epistemic and aleatoric uncertainty. Epistemic uncertainty arises from the neural network's nature of finding local optima, leading to estimated errors. Aleatoric uncertainty is prevalent in real-world industry problems, where noise and unexpected events can occur. Building a model that can tolerate all types of noise and disturbance is impossible, but having a risk-free model may lead to low returns. Therefore, balancing robustness and acceptable performance is essential. The distributionally robust optimization (DRO) method has gained popularity in research as it ensures the system's robustness without sacrificing too much performance.

After considering uncertainty and analyzing robust methods, we explored an algorithm that integrates DRO and RL to maximize performance and tolerance to uncertainty. DRO [14] is a data-driven method that can efficiently utilize limited data and provide safe solutions for problems under uncertainty. The actor-critic RL algorithm, originally developed from the policy iteration method, includes policy evaluation and improvement steps. In the absence of uncertainty, the algorithm can converge to the optimal solution, but when uncertainty exists, the final policy is at high risk due to the algorithm's greed for the highest reward without considering uncertainty. To address

this, we propose building an uncertainty set centered around the current policy according to DRO's instructions. The size of the uncertainty set is determined by the level of uncertainty measured from real-time observed data. The default updated policy is then replaced with the most conservative policy from this uncertainty set to pivot from a greedy policy to a safe one. This is similar to the safe RL method based on robust optimization (RO) [15], which replaces the default policy with the most conservative policy from the entire policy space to avoid the worst outcome. The DRO method's major advantage over robust-optimization-based RL is its ability to adaptively balance performance and robustness based on data understanding and avoid overly conservative policies. We refer to this as the risk-aware policy update step, which represents a significant improvement in our proposed method compared to previous safe RL research.

The article is structured as follows: Section II covers the problem formulation, RL, and related work on uncertainty. Section III provides a detailed explanation of our approach. Section IV presents the localization results of our method using an augmented field dataset. Section V concludes the article and highlights current concerns in class methods and RL. Nomenclature summarizes the notation used in the problem.

II. Preliminaries

A. Gamma-Ray Pattern Matching

The oil and gas industry employs underground sensor logging to continuously obtain 1-D records of rock properties in formations. One type of well-logging commonly used is the Gamma-ray well-log [16], which records variations in Gamma radiation with depth. Rocks at different depths emit varying amounts of Gamma Ray, which can indicate changes in lithology with specific patterns in the logging signal. Human experts typically interpret well-logging data using their domain knowledge. However, identifying repetitive patterns in new loggings with the same physical properties is a labor-intensive task, and manual interpretation can lead to errors. Automating the pattern-matching workflow can significantly reduce human errors and labor costs, thereby improving efficiency.

B. Soft Actor-Critic

RL is a machine learning subfield that aims to teach machines decision-making techniques by letting them interact with their environment using a trial-and-error learning approach. The decision-maker is modeled as an agent that interacts with the environment, receiving feedback in the form of rewards. The soft actor-critic (SAC) algorithm is a state-of-the-art RL method that aims to solve an MDP and includes several fundamental components, as shown in Nomenclature, from the current state s to the action-value function $Q^{\pi}(s, a)$.

The problem of localizing 1-D signal patterns can be viewed as a RL task, where an agent observes a reference pattern and an initial localization window within a longer signal record and takes actions to move the window closer to the target pattern. Each action leads to a better or worse localization state, which is rewarded or punished accordingly to encourage the agent to achieve better localization performance. In this way, the problem can be framed in the classic RL framework, with state s, action a, and reward r.

The SAC algorithm is an instance of actor-critic type algorithms and includes two models: the actor and the critic. The actor model is a function that maps states to actions, while the critic is a Q value function that maps state-action pairs to Q values. Both models utilize deep neural networks for representation.

During training, the agent interacts with the environment and collects episode samples (s, a, s', r), where s is the current state, a is the action taken, s' is the next state reached after the action, and r is the reward obtained. The SAC algorithm updates the critic and actor models alternatively to produce two prediction models, where the critic can accurately evaluate the state value, and the actor can take actions based on the current state observation, resulting in a substantial accumulated reward.

The critic's objective function can be expressed as follows:

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}, a_{t+1}) \sim \mathcal{D}} \left\{ \frac{1}{2} \left[Q_\theta(s_t, a_t) - \left[r(s_t, a_t) + \gamma Q_\theta(s_{t+1}, a_{t+1}) \right] \right]^2 \right\}. \tag{1}$$

The critic estimates the value of a state by predicting the Q value of the current state and the subsequent state following an action. The neural network incorporates the reward signal to enhance the precision of its predictions. Both the current Q network and the target Q network estimate the Q function of the current state and the Q value function of the next state. During the training process, the weights of the target Q network remain unchanged while the current Q network is updated. After a specific number of iterations, the weights of the current Q network are migrated to the target Q network.

Entropy regularization is a crucial aspect of SAC, which allows the policy to balance exploration and exploitation. Specifically, the policy is trained to optimize the tradeoff between maximizing the return and maximizing the entropy, which measures the diversity of the policy. A diverse policy can improve final performance by accelerating learning and preventing premature convergence to a bad local optimum. The objective function of SAC includes an entropy term that encourages exploration and is defined as follows:

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D, \epsilon_t \sim \mathbb{N}} \left[\alpha \mathcal{H}\left(\pi_\phi(a_t|s_t)\right) - Q_\theta(s_t, a_t) \right]. \tag{2}$$

The objective function for the policy network is denoted by $J_\pi(\phi)$ and incorporates the entropy measure $\mathcal{H}(\cdot)$. The temperature parameter α controls the importance of the entropy term. When the action space is continuous and the action distribution is assumed to be Gaussian, the mean errors of the actions cannot be backpropagated. To address this issue, a reparameterization trick was introduced by Kingma and Welling [17] that utilizes ϵ as a latent variation of the action $a_t = f_\phi(\epsilon_t, s_t)$, where $\epsilon_t \sim \mathbb{N}(0, 1)$. The maximization of entropy encourages policies to explore more and capture

multiple modes of near-optimal strategies, which can prevent premature convergence to a suboptimal solution.

Additionally, to address the issue of the sensitivity of the SAC to the temperature hyperparameter setting, Haarnoja et al. [18] proposed an automatic temperature parameter tuning mechanism that updates the parameter during the training process. The temperature parameter loss can be expressed as follows:

$$J(\alpha) = E_{a_t \sim \pi}\left[-\alpha\left(\log \pi(a_t|s_t) + \hat{\mathcal{H}}\right)\right] \tag{3}$$

where $\hat{\mathcal{H}}$ is a constant vector representing the target entropy.

C. Uncertainty and Overestimation Bias

In industrial RL applications, the performance of the RL algorithms is often hindered by two types of uncertainty. The first is epistemic uncertainty, which stems from limited training data samples. Since the agent only learns from a subset of the real environment, policy state values are computed inexactly. In testing, the agent may overestimate the value of certain wrong actions when encountering a new state, thereby introducing risks to decision-making [19], [20]. The second type of uncertainty is aleatoric uncertainty, which refers to noise from the industrial environment [21], [22]. This type of uncertainty can also cause observations of a new state to differ from those in training. Both types of uncertainty can shift the distribution of the testing environment from the distribution learned by the agent, resulting in a performance-dropping issue known as the generalization challenge. To address these issues, we propose a risk-averse strategy in this article to lower the risk of catastrophic outcome estimation errors.

D. Related Works

RL algorithms have achieved human-level performance in games like AlphaGo [23] and Atari Games [24]. However, there is still a significant gap between game environments and real-world RL applications due to high uncertainty. Collecting a dataset with all situation information is impractical, and there is often a shift in distribution between training and testing datasets. Therefore, building a robust AI model with incomplete observation is a crucial topic in the field. To address uncertainty challenges, previous work such as [15] and [25] introduced perturbations during training to achieve safe RL procedures. For instance, [26] added random force to a robotic agent while training it to walk, simulating the unpredicted forces applied by the environment's uncertainty. These methods show promising results, but designing an optimal perturbation strategy that mimics real-world scenarios is nearly impossible. Moreover, the data shortage is a more severe challenge in the early stage of RL real-world applications, leading to estimation errors in the policy state values computation. Recently, modern offline RL researchers have developed new strategies to tackle uncertainty challenges. The first category involves introducing policy constraints [27] that control the policy distribution based on safe constraints and policy learned from training, presenting promising results. The second category, called value function regularization methods [28], [29], modifies Q-function training objectives by introducing a safe regularizer, such as subtracting different formed uncertainty terms from Q value or reward to obtain a safe and robust performance. Uncertainty has been discussed in the mathematical community for decades, and there are three optimization approaches: RO, stochastic programming (SP), and DRO. The RO method considers worst case scenarios as constraints, which have extremely low probabilities of happening in practice. Although it provides a safe guarantee, it leads to conservative decisions with moderate performance. The SP method assumes the decision-maker has complete information on the uncertainty distribution, which is too extreme since it usually does not hold. The DRO method [14] bridges the gap between RO and SP by building an uncertainty set of the distribution for uncertainty parameters based on the data. This data-driven method efficiently utilizes the limited dataset and provides a robust solution with a safer guarantee than traditional methods that do not consider uncertainty. In short, the DRO method adaptively considers the uncertainty constraints level of the project's needs based on historical data [30] and has the potential to provide a safe and robust guarantee [31].

III. Method

Section III-A introduces a flexible approach that applies offline RL to solve 1-D signal pattern matching problems. The signal pattern localization problem is formulated as a MDP, with actions A, states S, and reward function R. An agent is trained in the environment (i.e., a signal sequence) to locate a relatively narrow window that includes the target signal fragment. Section III-A presents details of these three components. To solve this problem, Section III-B presents a unique and effective solution that has not been previously explored in the literature. We demonstrate how the uncertainty estimation technique from [13] can be combined with the discrete SAC algorithm [32] to tackle problems in real-world applications like data shortage. This proposed method represents one of the earliest attempts to use offline RL in this domain, and it offers a new and effective approach to solving the 1-D Gamma Ray signal pattern matching problem. In the testing set, the optimal policy of the agent is evaluated using the proposed method. By combining the uncertainty estimation technique and the discrete SAC algorithm, our approach offers a significant improvement in solving robust prediction under the data shortage problem. Section III-B presents technical details of the algorithm.

A. Signal Pattern Localization as a Dynamic Decision Process

1) Action: Following the seven-actions design [10] of the 2-D image pattern matching method, we proposed five-actions designs for this 1-D problem. $a \in$ {left, right, expand, shrink, stop} as illustrated in Fig. 1 top five icons. The action is represented by a single hot code. The left movement, for instance, is a vector with a length of five: [1; 0; 0; 0; 0]. The distribution of the probability of action is likewise represented by the vector. The collection of action A consists of four changes done to the window and

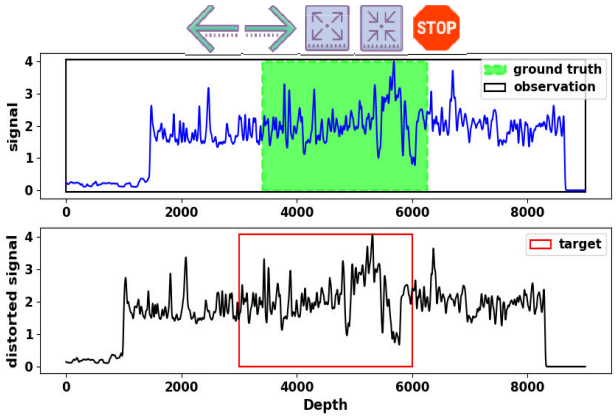


Fig. 1. Actions: left, right, expand, shrink, stop. Signals: the upper figure is signal_1 (blue line), and the bottom figure is signal_2 (black line). The target signal object is the signal section (red box), where the associated section exists in signal_1 (green section). The black bounding box is the signal section of the agent's observation after taking action in each time step.

one action to end the search procedure. These transformation actions can be divided into two sub-sets: actions to move the window in the horizontal axes: left and right, and actions to change the window scale: expand, shrink. In this way, the agent has two degrees of freedom to transform the box during any interaction with the environment. $w = [x_1, x_2]$ denotes a window in the order of its two edges. Where w represents the coordinates of the observation window, x_1 is the left boundary, and x_2 is the right boundary. Any of the transformation actions produces a discrete change to the size of the window by a factor proportional to its current size

$$\kappa_w = \kappa(x_2 - x_1) \tag{4}$$

where $\kappa \in [0, 1]$ is a action scale ratio. The transformations are then achieved by adding or subtracting κ_w from the x coordinates, depending on the effect required. For example, the horizontal move left/right action operator subtracts/adds κ_w to x_1 and x_2

$$x_1 := x_1 \pm \kappa_w; \quad x_2 := x_2 \pm \kappa_w. \tag{5}$$

While scale expand/shrink action adds/subtracts $(\kappa_w/2)$ from x_2, and adds it to x_1

$$x_2 := x_2 \pm \frac{\kappa_w}{2}; \quad x_1 := x_1 \pm \left(-\frac{\kappa_w}{2}\right). \tag{6}$$

Note that the initial observation window is located at the start to the end of the whole signal sequence. The stop-action does not transform the window as a trigger to indicate that the current window correctly localizes a signal object. This action terminates the current search sequence and restarts the window in an initial position to begin the search for a new object.

2) State: Fig. 1 shows two signal sequences signal_1, signal_2 (blue line, black line). The target signal object is the signal section $\text{signal}_{\text{target}} = \text{signal}_2(x_l{:}x_r)$ (red box), where the associated section exists in signal_1 (green section). Initial the whole signal_1 as the observation in time step 0: $x_1(0) = 0, x_2(0) = \text{length}(\text{signal}_1)$. And define the attend section (black box) as the signal section of observation after taking action in each time step t: $x_1(t+1), x_2(t+1) = T(x_1(t), x_2(t), a)$; $\text{signal}_{\text{attend}} = \text{signal}_1(x_1(t){:}x_2(t))$. T is the transition operator. For instance, the action of shrink is taken at time 0, $\kappa(0) = \kappa[x_2(0) - x_1(0)]$; $x_1(1), x_2(1) = x_1(0) + (\kappa(0)/2), x_2(0) - (\kappa(0)/2)$, $\text{signal}_{\text{attend}}(1) = \text{signal}_1(x_1(1){:}x_2(1))$. Finally, define the state representation as rescaled observed region and target signal fragment

$$s = \left[D\left(\text{signal}_{\text{attend}}\right), D\left(\text{signal}_{\text{target}}\right)\right]. \tag{7}$$

D is the rescale operator, which downsamples/upsamples signal to a size of 512 by using bi-linear interpolation. This operator matches the arbitrary size observation and target of signals with the fixed network's input (2×512). This design is an effective state representation for enormous attend or target signal scenarios from a large set of signals.

3) Reward Function: The reward function R is proportional to the agent's improvement to localize the signal object after selecting a particular action. Improvement is measured using the intersection-over-union (IoU) between the target signal object and the predicted window at any given time. Like the computer vision task of object detection, we propose to use human-labeled annotations as ground truth for evaluation localization methods. IoU differences between the present and next states are used to calculate the reward function. w_g is the ground truth window for the target signal object, as stated in Section III-A1. Then, IoU between w and w_g are defined as $\text{IoU}(w, w_g) = ((\text{area}(w \cap w_g))/(\text{area}(w \cup w_g)))$. The reward function $R_a(s, s')$ is awarded to the agent when it chooses the action to move from state s to s'. Each state s has an associated window w that contains the attended section. Then, the reward is as follows:

$$R_a(s, s') = \text{sign}\left(\text{IoU}\left(w', w_g\right) - \text{IoU}\left(w, w_g\right)\right). \tag{8}$$

If IoU progresses from state s to state s', then the reward will be positive; otherwise, it will be negative. This reward system is binary $R \in \{-1, +1\}$, and is employed for any action that transforms the box. Without quantization, the difference in IoU might be too small to guide the agent in accessing the actions. Binary rewards clearly communicate which transformations keep the object inside the window and take the window away from the target. In this way, the agent is penalized for taking the window away from the target and is rewarded for keeping the target object in the visible region until no other transformation improves localization. In that case, the best action to choose should be the stop trigger. The stop action has a different reward scheme because it leads to a terminal state that does not change the window, and thus, the differential of IoU will always be zero for this action. The reward for the trigger is a thresholding function of IoU as follows:

$$R_{\text{stop}}(s, s') = \begin{cases} +\eta, & \text{if IoU}\left(w, w_g\right) \geq \tau \\ -\eta, & \text{otherwise} \end{cases} \tag{9}$$

where R_{stop} is the reward for stop action, η is the stop reward, which is much higher than the improvement reward, and τ is a threshold that indicates the minimum IoU, which can be considered as a positive detection. The standard threshold

for object detection evaluation is 0.5, but $\tau = 0.6$ is used in our training process to obtain a better localization. A larger value for τ has a negative effect on the performance because it leads the agent to learn only to detect clearly visible objects, which will cause the neglect of truncated or naturally occluded objects. Finally, the proposed reward scheme implicitly considers the number of steps as a cost because all RL algorithms, like SAC, already consider the discount of future rewards (positive and negative).

In short, about the reward function, once the agent moves closer to the target, we reward it 1. If they move away from the target, we punish it with a reward of -1. After the search is complete, the agent will make a determination regarding the ultimate target. If the agent is successful in locating the target, it will be rewarded with ten points, the value of which will be determined empirically. In addition to this, the discount ratio takes into account the passage of time. It is imperative that it locate the target in the shortest amount of time feasible so that it may maximize its reward.

B. Distributionally Robust SAC for Signal Object Localization

1) Distributionally Robust RL Theory: The DRSAC [13] risk-averse strategy first uses the experience data to model the errors of underestimation. Then provide a modified policy to avoid catastrophic results. The Bellman equation in the RL algorithm can be written as follows:

$$[\mathcal{T}V](s) = \mathbb{E}_{a\sim\pi(.|s)}\left\{r(s,a) + \gamma \mathbb{E}_{s'\sim p(s'|s,a)}\left[V\left(s'\right)\right]\right\} \quad (10)$$

where $\mathcal{T}$ is the bellman operator.

Under the policy iteration algorithm theory, the policy evaluation step is subject to an estimation error δ_t due to a finite sample of transitions used to perform evaluation

$$\pi_{t+1} \in \mathcal{G}\left(\tilde{V}_{t+1}\right) \quad (11)$$

$$\tilde{V}_{t+1} = \mathcal{T}^{\pi_{t+1}}\tilde{V}_t + \delta_t \quad (12)$$

where $\mathcal{G}$ is the greedy strategy based on the value observations.

Given a policy π and an error function $\epsilon \in \mathbb{R}^S$, define the uncertainty set $\mathcal{U}_\epsilon(\pi)$ by

$$\mathcal{U}_\epsilon(\pi) := \left\{\tilde{\pi} \in \Delta_A^S \middle| D_{KL}\left(\tilde{\pi}(.|s)\middle|\middle|\pi(.,s) \le \epsilon(s)\right)\ \forall s \in S\right\}. \quad (13)$$

Δ_A^S is the function space mapping from state space to action space, and $\epsilon(s)$ is the uncertainty ball's radius. Then we can calculate the most conservative policy associated with this uncertainty set through the most conservative value estimation regards the uncertainty set mentioned before

$$\mathcal{T}^{\pi^\epsilon}V := \min_{\tilde{\pi}\in\mathcal{U}_\epsilon(\pi)} \mathcal{T}^{\tilde{\pi}}V. \quad (14)$$

The uncertainty size can be determined by the following equation:

$$\epsilon_t(s) = \begin{cases} Cn_t(s)^{-\xi}, & \text{if } \dfrac{t}{S} \le n_t(s) \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where constants C and ξ control the size of the uncertainty set. $n_t(s)$ denotes the visiting times of state s visited by the agent. This equation determined the size of uncertainty based on the agent's familiarity level with the current state. This familiarity level is quantified by using visiting times.

After a series of derivations using the Fenchal duality and Taylor approximation, the core perspective of distributionally robust RL can be described with the following equation that achieves the robustness by encouraging or penalizing the variance of Q value under action distribution μ induced by the prior policy, i.e.,

$$\Omega^*[Q_V(s,.)] = \mathbb{E}_{a\sim\mu}\left\{Q_V(s,a) + \frac{1}{2\lambda}\mathrm{Var}_{a\sim\mu}[Q_V(s,a)]\right\} + O\left(\frac{1}{\lambda^2}\right) \quad (16)$$

where Ω^* is the approximated Fenchel conjugate of the KL-divergence-based regularized Bellman operator, and λ is the regularization parameter. For derivation detail or convergence and lower bound proof, please check the [13].

Finally, the reward shaping method is implemented for a combination of DRO and RL algorithms SAC

$$\Phi(s) := \frac{1}{2\lambda}\mathrm{Var}_{a\sim\mu}[Q_V(s,a)] \quad (17)$$

$$r^\Omega\left(s,a,s'\right) := r(s,a) + \gamma\Phi\left(s'\right) - \Phi(s) \quad (18)$$

where the r^Ω is the regularized reward. There are further derivatives for the continued robotic control problem for r^Ω calculation because the variance of the Q value is not tractable. Accordingly, we designed the discrete action space for our task to make the regularized reward explicitly calculated. More detail will be discussed in the sequel.

2) Discrete Distributionally Robust SAC for Signal Object Localization: As with all recent actor-critic algorithms, we proposed to use neural networks to serve as the actor and critic. Inspired by discrete SAC in [32] and DRSAC in [13], this article proposes a discrete DRSAC by discretizing the action space. This discrete setting brings five important changes as mentioned in SAC-Discrete work [32] as follows.

1) Q function moves from $Q : S \times A \to \mathbb{R}$ to $Q : S \to \mathbb{R}^{|A|}$. When there are infinitely possible actions, it is impossible to give the exact Q table for each action. However, the discrete setting can achieve it.
2) The discrete algorithm does not require that the action distribution is a Gaussian distribution. The neural network can provide the action distribution as output.
3) The soft state-value function can be directly calculated instead of using the Monte Carlo estimation. This change can reduce the variance involved in the estimate of the objective. The soft state-value function can be written as follows:

$$V(s_t) := \pi(s_t)^T\left[Q(s_t) - \alpha\log(\pi(s_t))\right]. \quad (19)$$

4) Similarly, the temperature parameter changes from (3)–(20) and reduces the variance of estimation

$$J(\alpha) = \pi(s_t)^T\left[-\alpha(\log\pi(s_t) + \mathcal{H})\right]. \quad (20)$$

Algorithm 1 Distributionally Robust Soft Actor-Critic With Discrete Actions (DRSAC-Discrete)

Initialise actor-network: $\pi_\phi : S \to [0,1]^{|A|}$;
Initialise local critic networks: $Q_{\theta_1}, Q_{\theta_2} : S \to \mathcal{R}^{|A|}$;
Initialise target critic networks: $Q'_{\theta_1}, Q'_{\theta_2} : S \to \mathcal{R}^{|A|}$;
Initialize an empty replay buffer: $\mathcal{D}$;
Update target network weights: $\theta \to \theta'$;
Set uncertainty level: $C, \eta > 0$;
for *each iteration* **do**
  for *each time step* **do**
    Sample action from the policy: $a_t \sim \pi_\phi(a_t|s_t)$;
    Sample transition from the environment: $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$;
    Store the transition in the replay buffer: $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r(s_t, a_t), s_{t+1})$
  end
  for *each gradient step* **do**
    Sample $(s, a, r, s') \sim \mathcal{D}$;
    Update regularized reward: $r^\Omega \leftarrow r(s,a) + \frac{C}{N_t(s)^\xi}[\gamma Var(Q(s')) - Var(Q(s))]$;
    Update the critic: $\theta_i \leftarrow \theta_i - \nu \nabla_\theta J(\theta_i) \forall i \in \{1,2\}$;
    Update actor network: $\phi \leftarrow \phi - \nu\pi \nabla_\phi J_\pi(\phi)$;
    Update temperature: $\alpha \leftarrow \alpha - \nu \nabla_\alpha J(\alpha)$;
    Update target critic networks: $Q'_i \leftarrow \beta Q'_i + (1-\beta) Q'_i$ for $i \in \{1,2\}$
  end
end
Result: θ_1, θ_2, ϕ

5) Instead of using reparameterization, the discrete setting allows the loss function of policy can be calculated directly by using the exact action distribution. The policy objective function changes from (2) to

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D}\{\pi_t(s_t)^T [\alpha \mathcal{H}(\pi_\phi(a_t|s_t)) - Q_\theta(s_t, a_t)]\}. \tag{21}$$

Except for those innovative improvements and modifications, the core idea of the introduction of the discrete action space is that the equation of Φ in (18) can be calculated by the following equation directly:

$$\mathrm{Var}_{a \sim \pi_\phi}[Q(s,a)] = \frac{\sum_{i=1}^{M} (Q(s,a_i) - \bar{Q})}{M} \tag{22}$$

where $\bar{Q}$ is the mean of Q distribution and M is the number of actions in our setting: $M = 5$. This change makes the calculation of a distributionally robust regularizer becomes feasible and directly reduces the approximating error.

The algorithm for DRSAC with discrete actions (DRSAC-Discrete) is given by Algorithm 1.

IV. Experiments

1) Neutral Network Architecture: Many previous actor-critic methods with a convolutional network show their effectiveness on different tasks, such as human activity classification [33], fault recognition [34], and elevator group control [35]. The architecture of the neural network used in this article to represent the actor and critic is shown in Fig. 2. The network is built utilizing exactly three convolutional layers and two fully connected layers of network structure as proposed in the original SAC-discrete [32] article. The only difference is that the proposed network in our article has modified input and hidden layer sizes. The fixed-size 2×512 state matrix mentioned in Section III-A is the input of both actor and critic. Both actor and critic networks extract the features of the state by a stack of convolutional layers followed by two fully-connected layers. Firstly, the input is passed through two 1-D convolutional layers, the first one with kernel size 8 and stride 4, and the connected second 1-D convolutional layer has kernel size 4 and stride 2. Then, two same convolutional layers with kernel size 3 and stride 1 are concatenated to the first two layers. Finally, the flattened feature map output of convolutional layers is passed through two fully connected layers with 3712 and 517 neurons, respectively. All convolutional layers and fully connected layers have the ReLU active function except for the last fully connected layers. Finally, the actor network using the softmax active layer at the output end outputs a vector with length 5 to predict the distribution of the probability of action in this state. Critics straightforwardly output a vector of the last fully connected layer with length five to provide the Q value of these five actions in this state.

A. Dataset

After applying elastic distortion, [36] and random shifting, 177 synthetic Gamma-ray log pairings were produced from 59 genuine logs in order to enlarge the training set. For each data sample, there are two series. We define one segment of this series as the pattern's reference. Because the other series is generated by distorting the first series, it contains the same pattern, but the location of this pattern is different from the one in the first series. We define it as the search object. Because known elastic transformations generate this dataset, the ground truth location of the target signal series of each sample is known and can be used for training and evaluation. In the first experiment, the logs are divided into two categories: training and testing, with 83% and 17%, respectively. Experiment 2 involves removing 13% of the data from the training dataset in order to simulate a data shortage condition typical of early-stage RL applications.

B. Training, Evaluation

For evaluation of algorithms' performance, we define accuracy as the same as object detection: accuracy = (TP/(TP + FP)), where TP stands for True Positive and FP stands for False Positive. Then we test both implemented SAC-discrete and DRSAC-discrete to compare with the baseline FastDTW.

In experiment 1, we train agents in a no-data shortage setting by using both implemented SAC-discrete and DRSAC-discrete algorithms on the whole training dataset. Then we test both algorithms on the testing dataset. We also conduct the baseline algorithm FastDTW on the same dataset. But we find mismatching case like Fig. 1 is shown. For quantity

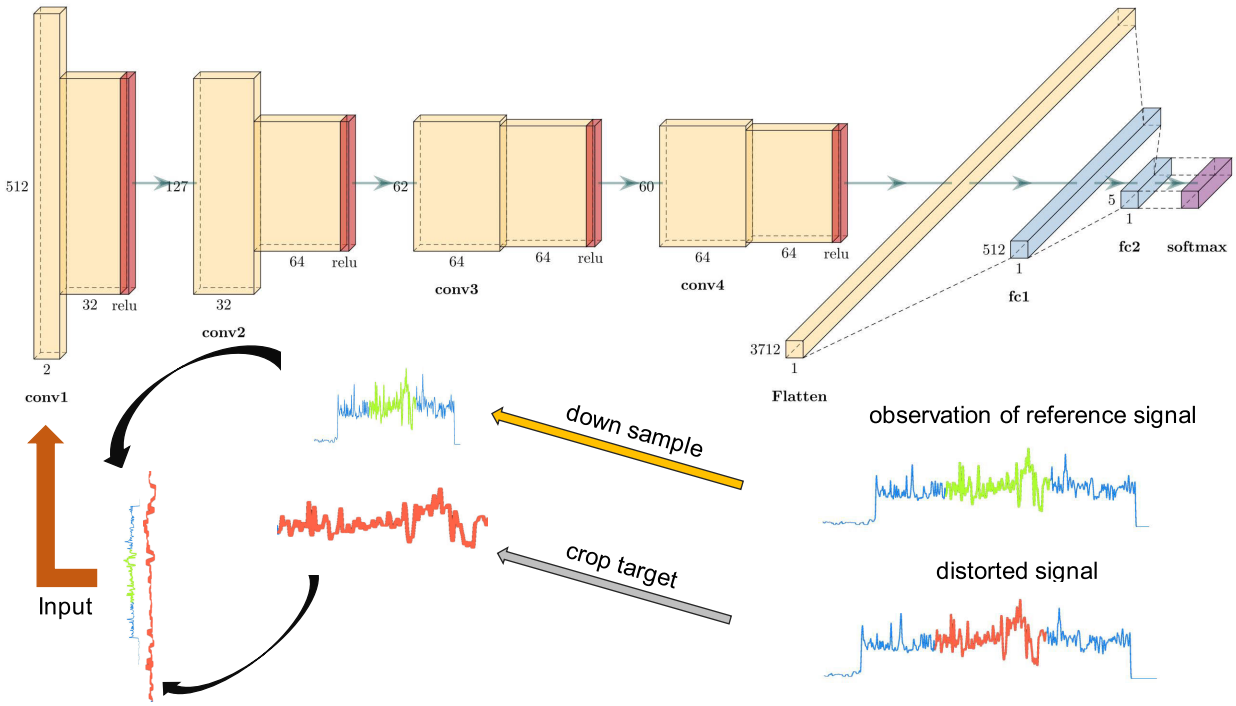


Fig. 2. Neural network architecture as actor and critic: the fixed-size 2 × 512 state matrix is the input of both actor and critic. Both actor and critic networks extract the features of the state by a stack of convolutional layers followed by two fully-connected layers. Critics straightforwardly output a vector of the last fully connected layer with length five to provide the Q value of these five actions in this state.

TABLE I
ABLATION STUDY

Method	EXP1:Accuracy (no shift)	EXP2:Accuracy (shift)	Steps (no shift)	Steps (shift)	Pattern length needed	Downsampled pattern
baseline FastDTW + window search	83%	83%	$\frac{\textbf{length of log}}{\textbf{length of pattern}} = 3000$	3000	✓	✗
SAC-discrete	94%	64%	6.503	9.03	✗	✓
DRSAC-discrete	85%	71%	8.157	9.95	✗	✓

evaluation, as Table I shows, the agent learned to localize the target signal object with high accuracy of 94% in the only average of 6.5 steps. Compared to the SAC-Discrete, the DRSAC-Discrete has a more conservative performance. More specifically, its accuracy was lower than SAC but still achieved 85%. Comparing those RL methods with the baseline, Fast-DTW [37] (83%), both algorithms show superior.

In experiment 2, 15% data are removed from the training dataset to mimic the data shortage situation, repeat training and testing of SAC-discrete and DRSAC-discrete with the same hyperparameters' setting as experiment 1. The SAC-Discrete performance dropped rapidly to 64% because the training dataset no longer provided enough information for learning, the enlarged uncertainty level, and the distributional shift between the training and testing dataset confused the agent's decision. However, since the conservative policy of DRSAC-Discrete, it can avoid some risks caused by the difference in training and testing datasets and still maintain a fine accuracy of 71% compared to SAC-Discrete.

In summary, both RL algorithms outperform the baseline algorithm when training dataset distribution can provide enough information for testing the dataset. Moreover, unlike the traditional handcraft window searching problem, the RL algorithms adaptively change the window size to suit the target and do not need any human involved in searching grid and window design. But both RL failed when the training dataset's qualify degenerated. Therefore, practitioners should build both model and dataset quality assessment and evaluation monitoring modules in their deployment pipeline and keep the traditional method and original model as their backup and baseline solutions for A/B testing and rollback. The Q value, the variance of Q, and the parameter of reward regularizer

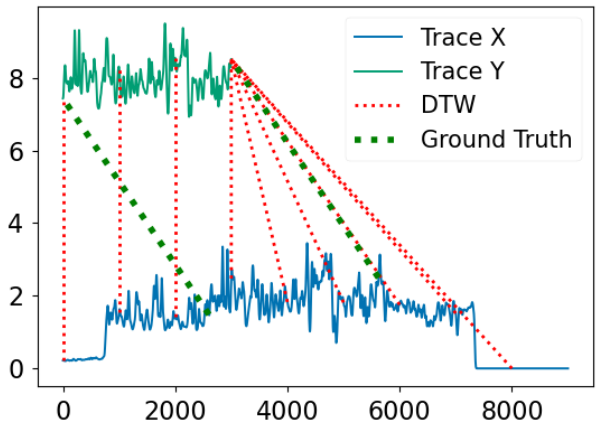


Fig. 3. Baseline method (DTW) evaluation.

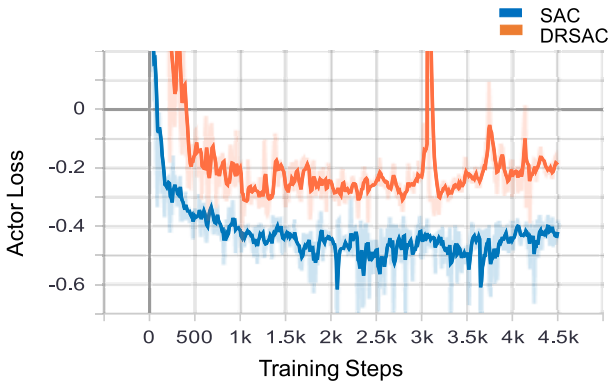


Fig. 4. Actor loss.

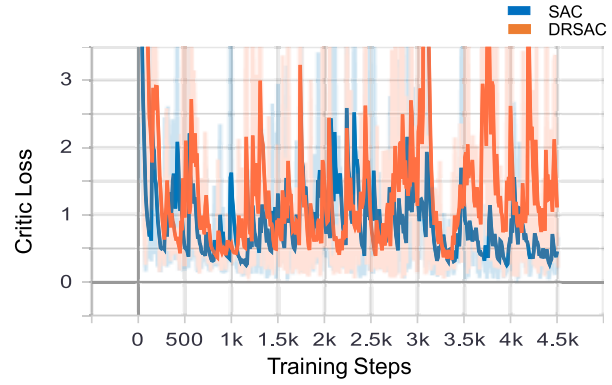


Fig. 5. Critic loss.

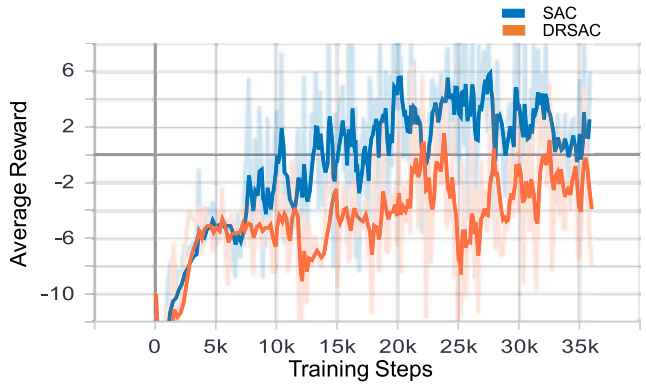


Fig. 6. Average reward.

also are good indexes for monitoring the model drift after deployment, except for the accuracy, prediction mean, or other statistic indexes. Because the Q value shows us the estimation value of all actions in the current state, the variance of Q shows the certainty of the model's prediction, and the parameter of the reward regularizer shows the model's familiar level of the current state. By combining that information, engineers can easily troubleshoot and improvement of the system.

The training loss and average reward are shown in Figs. 4 and 5. The critic loss defines as (1) represents how good the critic's assessment regarding each state visited by the agent is. The X-axis is the training steps; the y-axis represents the loss value. Moreover, actor loss defines as the

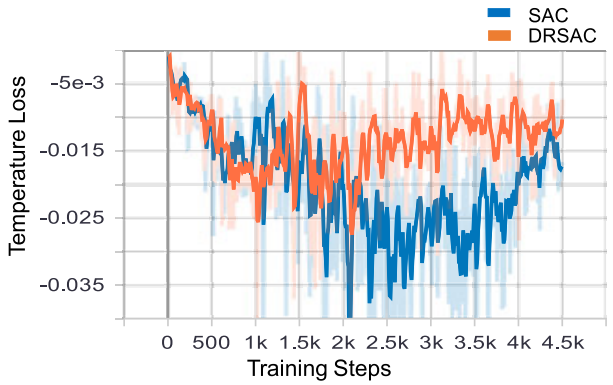


Fig. 7. Temperature loss.

(2) representing the capability to pick a good action with a higher Q value. Both losses converge to a stable level during training. The accumulated reward of a trajectory is the total gained rewards from initial to stop for one signal object searching. The average reward is the last 100 trajectories, averaged accumulated rewards. Fig. 6 shows that both SAC-Discrete and DRSAC-Discrete agents were learning the environment successfully during training. In the early steps, the training is low because most steps in the trajectory will not lead to a searching window closer to the target. In the later stage of training, almost all steps can improve the location of the search window and generate a high average reward value.

V. Conclusion

This article presents an MDP formulation of the Gamma Ray log pattern localization problem that draws inspiration from human attention. Our contributions and conclusions are summarized as follows: This article represents one of the earliest attempts to apply offline RL to solve 1-D signal pattern-matching problems in the oil and gas industry. Our agent learns pattern-matching decision processes from data, and the experiments suggest that this method could potentially aid in localizing complex signal patterns. However, the performance of RL algorithms can be hampered by data shortages and insufficient sample sizes in the environment. The proposed DRSAC-Discrete approach's accuracy result is lower than that of the traditional RL method because safe solutions require a tradeoff between accuracy performance and robustness. Nevertheless, compared to the traditional RL method, the proposed DRSAC-Discrete approach performs better with less training data because it takes uncertainty into account during training. Experimental evaluations with augmented field logging data [37] demonstrate our method's superior performance and generalization ability.

References

[1] P. Kearey, M. Brooks, and I. Hill, *An Introduction to Geophysical Exploration*, vol. 4. Hoboken, NJ, USA: Wiley, Apr. 2002.

[2] J. Zangwill, "Depth matching—A computerized approach," in *Proc. SPWLA 23rd Annu. Logging Symp.* Corpus Christi, TX, USA: Society of Petrophysicists and Well-Log Analysts, Jul. 1982, pp. 1–11.

[3] D. Lineman, J. Mendelson, and M. N. Toksoz, "Well to well log correlation using knowledge-based systems and dynamic depth warping," in *Proc. SPWLA 28th Annu. Logging Symp.* London, U.K.: Society of Petrophysicists and Well-Log Analysts, Jun. 1987, pp. 1–34.

[4] J. Mei, M. Liu, Y. Wang, and H. Gao, "Learning a Mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1363–1374, Jun. 2016.

[5] A. D. Dileep and C. C. Sekhar, "GMM-based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1421–1432, Dec. 2013.

[6] K. J. Bergen, P. A. Johnson, M. V. de Hoop, and G. C. Beroza, "Machine learning for data-driven discovery in solid earth geoscience," *Science*, vol. 363, no. 6433, Mar. 2019, Art. no. eaau0323.

[7] L. Liang, T. Le, T. Zimmermann, S. Zeroug, and D. Heliot, "A machine learning framework for automating well log depth matching," in *Proc. SPWLA 60th Annu. Logging Symp.* Woodlands, TX, USA: Society of Petrophysicists and Well-Log Analysts, Jun. 2019, pp. 585–595.

[8] S. Wang, Q. Shen, X. Wu, and J. Chen, "Automated gamma-ray log pattern alignment and depth matching by machine learning," *Interpretation*, vol. 8, no. 3, pp. SL25–SL34, Mar. 2020.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, Nov. 2018.

[10] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2488–2496.

[11] Y. Zi, L. Fan, X. Wu, J. Chen, S. Wang, and Z. Han, "Active gamma-ray well logging pattern localization with reinforcement learning," in *Proc. 2nd Int. Meeting Appl. Geosci. Energy*. Houston, TX, USA: OnePetro, Aug. 2022, p. 3694.

[12] K. Tsai, L. Fan, L. Wang, R. Lent, and Z. Han, "Multi-commodity flow routing for large-scale LEO satellite networks using deep reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Austin, TX, USA, Apr. 2022, pp. 626–631.

[13] E. Smirnova, E. Dohmatob, and J. Mary, "Distributionally robust reinforcement learning," in *Proc. ICML Workshop Real-Life Reinforcement Learn.*, Long Beach, CA, USA, Jun. 2019, pp. 1–10.

[14] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, Dec. 2019.

[15] J. Morimoto and K. Doya, "Robust reinforcement learning," *Neural Comput.*, vol. 17, no. 2, pp. 335–359, 2005.

[16] L. G. Howell and A. Frosch, "Gamma-ray well-logging," *Geophysics*, vol. 4, no. 2, pp. 106–114, Apr. 1939.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[18] T. Haarnoja et al., "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[19] H. V. Hasselt, "Double Q-learning," in *Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 23, Dec. 2010, pp. 2613–2621.

[20] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6584–6598, Nov. 2022, doi: 10.1109/TNNLS.2021.3082568.

[21] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, Apr. 2019, doi: 10.1016/j.neucom.2019.01.103.

[22] R. L. Russell and C. Reale, "Multivariate uncertainty in deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7937–7943, Dec. 2022, doi: 10.1109/TNNLS.2021.3086757.

[23] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[24] A. P. Badia et al., "Agent57: Outperforming the Atari human benchmark," in *Proc. Int. Conf. Mach. Learn.*, Vienna, Austria, Jul. 2020, pp. 507–517.

[25] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," 2017, *arXiv:1712.03632*.

[26] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 1–11.

[27] X. Chen, Z. Zhou, Z. Wang, C. Wang, Y. Wu, and K. Ross, "BAIL: Best-action imitation learning for batch deep reinforcement learning," 2019, *arXiv:1910.12179*.

[28] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," 2020, *arXiv:2006.04779*.

[29] R. Agarwal, D. Schuurmans, and M. Norouzi, "An optimistic perspective on offline reinforcement learning," in *Proc. ICML*, Vienna, Austria, Nov. 2020, pp. 104–114.

[30] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Oper. Res.*, vol. 58, no. 3, pp. 595–612, Jun. 2010.

[31] M. Staib and S. Jegelka, "Distributionally robust optimization and generalization in kernel methods," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 9134–9144.

[32] P. Christodoulou, "Soft actor-critic for discrete action settings," 2019, *arXiv:1910.07207*.

[33] Y. Lu, Y. Li, and S. Velipasalar, "Efficient human activity classification from egocentric videos incorporating actor-critic reinforcement learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, China, Sep. 2019, pp. 564–568.

[34] Z. Wang and J. Xuan, "Intelligent fault recognition framework by using deep reinforcement learning with one dimension convolution and improved actor-critic algorithm," *Adv. Eng. Informat.*, vol. 49, Aug. 2021, Art. no. 101315. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034621000689

[35] Q. Wei, L. Wang, Y. Liu, and M. M. Polycarpou, "Optimal elevator group control via deep asynchronous actor–critic learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5245–5256, Dec. 2020.

[36] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, Edinburgh, U.K., 2003, pp. 1–6.

[37] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

Yuan Zi (Student Member, IEEE) received the B.S. degree in exploration technology and engineering (geophysical exploration) from the China University of Petroleum (East China), Dongying, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA, where he is co-advised by Dr. Jiefu Chen and Dr. Zhu Han.

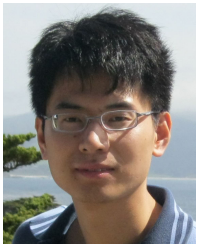
In 2021, he worked as a Machine Learning Research Summer Intern at Siemens Sustainable Automation Solution (SAS), Princeton, NJ, USA, and in 2022, he interned with Shell AI, Houston. His research interests lie in technology for sustainability, machine learning, computer vision, anomaly detection, inverse problems, optimization, and signal processing.

Lei Fan (Senior Member, IEEE) received the Ph.D. degree in operations research from the Industrial and System Engineering Department, University of Florida, Gainesville, FL, USA, in 2015.

He is currently an Assistant Professor with the Engineering Technology Department, University of Houston, Houston, TX, USA. Before this position, he worked in the electricity energy industry for several years. His research interests include quantum computing, optimization methods, complex system operations, and power system operations and planning.

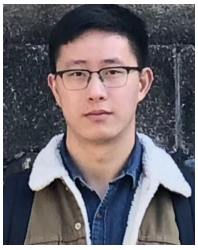
Xuqing Wu (Member, IEEE) received the bachelor's degree in automation from the University of Science and Technology Beijing, Beijing, China, in 1995, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 2011.

He is currently an Associate Professor at the University of Houston. Prior to joining the University of Houston, he was a Data Scientist and a Software Engineer in the energy and IT industry. His research interests include machine learning, probabilistic modeling, computer vision, and their applications in sensing and imaging.



Jiefu Chen (Senior Member, IEEE) received the B.S. degree in engineering mechanics and the M.S. degree in dynamics and control from the Dalian University of Technology, Dalian, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Duke University, Durham, NC, USA, in 2010.

From 2011 to 2015, he was a Staff Scientist with the Advantage Research and Development Center, Weatherford International Ltd., Houston, TX, USA. He joined the University of Houston, Houston, in 2015, and currently he is an Associate Professor of electrical and computer engineering. His research interests include computational electromagnetics, inverse problems, machine learning for scientific computing, oilfield data analytics, seismic data processing, underground and underwater wireless communication, and well logging.



Shirui Wang (Student Member, IEEE) received the B.S. degree in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA.

His research interests centers signal processing, data analysis, machine learning, and computer vision and their implementations for seismic data processing and interpretation.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was an Research and Development Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an Assistant Professor at Boise State University, Boise, ID, USA. Currently, he is a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid.

Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018, has been an AAAS Fellow since 2019, and has been an ACM Distinguished Member since 2019. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the Best Paper Award for the *EURASIP Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of communications systems (Best Paper Award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS) in 2016, and several best paper awards in IEEE conferences. He has been a 1% Highly Cited Researcher since 2017 according to Web of Science. He is also the Winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation "for contributions to game theory and distributed management of autonomous communication networks."