

# Self-Initiated Open World Learning for Autonomous AI Agents

Bing Liu<sup>1</sup>, Eric Robertson<sup>2</sup>, Scott Grigsby<sup>2</sup>, Sahisnu Mazumder<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Illinois at Chicago, USA

<sup>2</sup> PAR Government Systems Corporation, USA

liub@uic.edu, {eric\_robertson, scott\_grigsby}@partech.com, sahisnumazumder@gmail.com

## Abstract

As more and more AI agents are used in practice, it is time to think about how to make these agents fully autonomous so that they can learn by themselves in a self-motivated and self-supervised manner rather than being retrained periodically on the initiation of human engineers using expanded training data. As the real-world is an open environment with unknowns or novelties, *detecting* novelties or unknowns, *characterizing* them, *accommodating* or *adapting* to them, *gathering* ground-truth training data, and *incrementally learning* the unknowns/novelties are critical to making the agent more and more knowledgeable and powerful over time. The key challenge is how to automate the process so that it is carried out on the agent's *own initiative* and *through its own interactions* with humans and the environment. Since an AI agent usually has a performance task, *characterizing* each novelty becomes critical and necessary so that the agent can formulate an appropriate response to *adapt* its behavior to *accommodate* the novelty and to *learn* from it to improve the agent's adaptation capability and task performance. The process goes continually without termination. This paper proposes a theoretic framework for this learning paradigm to promote the research of building *Self-initiated Open world Learning* (SOL) agents. An example SOL agent is also described.

## 1 Introduction

Classic machine learning (ML) makes *closed world assumption*, which means that what are seen by the system in testing or deployment must have been seen in training (Fei and Liu 2016; Bendale and Boulton 2015; Liu 2020), i.e., there is nothing new or novel occurring in testing or deployment. This assumption is invalid in practice as the real world is an open environment with unknowns or novel objects. For humans, novelties or unknowns serve as an intrinsic motivation for learning. Human novelty detection results in a cascade of unique neural responses and behavioral changes that enable exploration and flexible memory encoding of the novel information. As learning occurs, this novelty response is soon lost as repeated exposure to novelty results in fast neural adaptation (Tulving and Kroll 1995; Murty et al. 2013). In order to make an AI agent thrive in the real open world, like humans, it has to detect novelties and learn them incrementally to make the system more knowledgeable and adaptable. It must do so *on its own initiative* rather than relying on human engineers to retrain the system periodically. That is, it must

learn in the *open world* in a *self-motivated manner* in the context of its *performance task*.

We use the hotel guest-greeting bot example from (Chen and Liu 2018) to illustrate *Self-initiated Open world Learning* (SOL). The *performance task* of the bot is to greet hotel guests. When it sees a guest it has learned, e.g., John, it greets him by saying

“Hi John, how are you today?”

When it sees a new guest, it should detect this guest as new or novel because it has never seen him/her before. This is a *novelty detection* problem (also known as *out-of-distribution* (OOD) *detection*). It then needs to *accommodate* or *adapt* to the novel situation, e.g., by saying to the new guest

“Hello, welcome to our hotel! What is your name, sir?”

If the guest replies “David,” the bot takes some pictures of the guest to *gather training data* and then *incrementally* or *continually learn* to recognize David. The name “David” serves as the *class label* of the pictures taken. When it sees this guest again, it can say

“Hi David, how are you today?” (he is no longer novel)

Clearly, in an actual hotel, the situation is much more complex than this. First of all, if the bot uses a video camera, when it sees a novel object, it experiences a *data distribution change* as it sees the new object for a period of time rather than just one image instance. It is desirable to detect the new object as quickly as possible. Next, how does the system know that the novel object is actually a person, not a dog? If the system can recognize the object as a person, how does it know that he/she is a hotel guest, not a policeman? In order to adapt to the novel object or situation, the system must first *characterize* the novel object as without it, the agent does not know how to *adapt*. In this case, some classification is needed to decide whether it is a person with luggage. If the object is a person but has no luggage, the bot will not respond or learn to recognize the person as it is *irrelevant* to its performance task. If it is a moving object but not a person, it should notify a hotel employee and learn to recognize the object so that it will no longer be novel when it is seen again in the future. In short, for each characterization, there is a corresponding *response* or *adaptation strategy*, which can be NIL (i.e., do nothing). This discussion shows that in order to characterize, the agent must already have rich world knowledge. Clearly, there is

also a *risk* involved when making an incorrect decision.

As classic learning matures, we should go beyond the existing framework to study how to enable the learner to learn by itself via its own interactions with humans and the environment, involving no human engineers. This paper proposes the SOL framework to promote the research of autonomous learning agents so that they can face the real open world and learn by themselves. An example SOL agent in the context of dialogue systems or chatbots that implements the main ideas in this paper will also be discussed.

Although *open world learning* has been studied by several researchers (Bendale and Boulton 2015; Fei, Wang, and Liu 2016; Xu et al. 2019), it mainly focuses on novelty detection (Parmar, Chouhan, and Rathore 2021), which is also called *open set detection* or *out-of-distribution detection* in the literature. Recently, zero-shot out-of-distribution detection is investigated in (Esmailpour et al. 2022) based on the pre-trained model CLIP (Radford et al. 2021). Some researchers have further studied how to automatically identify the classes or categories of the detected novel object instances (Shu, Xu, and Liu 2018). Yet, some others have studied learning the novel objects or classes after they have been detected (Bendale and Boulton 2015; Fei, Wang, and Liu 2016; Xu et al. 2019). An recent survey of the topic can be found in (Yang et al. 2021). There is also a general discussion paper about open world learning in (Langley 2020), which presents many interesting blue sky ideas. However, SOL differs from these prior studies in many aspects.

(1) SOL stresses self-initiation in learning, which means that all the learning activities from start to end are self-motivated and self-initiated by the agent itself. The process involves no human engineers.

(2) Due to self-initiation, SOL enables learning after the model deployment like human learning on the job or while working, which has not been attempted before. In existing learning paradigms, after a model has been deployed, there is no more learning until the model is updated or re-trained on the initiation of human engineers.

(3) SOL is also a lifelong and continual learning paradigm again because learning is self-initiated and can be done continually or lifelong. It is thus connected with lifelong and continual learning, which is another active research area in machine learning (Chen and Liu 2018).

(4) SOL involves extensive interactions of the learning agent and human users, other AI agents, and the environment. The main purpose is to acquire ground-truth labels and training data by itself.

(5) SOL makes learning autonomous and a learning agent equipped with SOL can face the real open world. We also believe that SOL is necessary for the next generation machine learning techniques. Finally, note that although SOL focuses on self-initiated learning, it does not mean that the learning system cannot learn a task given by humans or other AI agents incrementally or continually

## 2 Self-Initiated Open World Learning (SOL)

Our agent has a *primary performance task* and a pair of key modules ( $T, S$ ), where  $T$  is the *primary task-performer* (e.g.,

the dialogue system of the greeting bot) and  $S$  is a *set of supporting or peripheral functions* (e.g., the vision system and the speech system of the bot) that supports the primary task-performer. The primary task-performer as well as each support function may consist of four main sub-components ( $L, E, N, I$ ), where  $L$  is a SOL learner,  $E$  is the executor that performs the task or function,  $N$  is a novelty characterizer for characterizing each novelty so that  $E$  can formulate an appropriate response to the novelty, and  $I$  is an interactive module for the agent or  $L$  to communicate with humans or other agents (e.g., to gain ground-truth training data). Since each function and the primary task performer has the same components, we will discuss them in general rather than distinguishing them, but will distinguish them when necessary. We will not discuss the relationships and interactions of the components. Below, we first discuss the SOL learner  $L$ , which starts with novelty and/or data shift detection.

### 2.1 Data Shift

The classic ML depends on the *independent and identically distributed* (IID) assumption. SOL deals with non-IID data, which is commonly known as *data shift*. We use supervised learning as the task to develop the ideas, which can be easily adapted to other learning settings.

Let the training data be  $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ , where  $\mathbf{x}_i \in X$  is a training example following the training distribution  $P_{tr}(\mathbf{x})$  and  $y_i \in Y_{tr}$  is the corresponding class label of  $\mathbf{x}_i$  and  $Y_{tr}$  is the set of all class labels that appear in  $\mathcal{D}_{tr}$ .

Note that  $P(\mathbf{x}, y) = P(y|\mathbf{x})P(\mathbf{x})$ . Given  $\mathbf{x} \in X$  and  $y \in Y_{tr}$  in both training and testing, existing research has proposed the following three main types of data shift happening in testing (Moreno-Torres et al. 2012).

**Definition (covariate shift).** *Covariate shift* refers to the distribution change of the input variable  $\mathbf{x}$  between training and test phases, i.e.,  $P_{tr}(y|\mathbf{x}) = P_{te}(y|\mathbf{x})$  and  $P_{tr}(\mathbf{x}) \neq P_{te}(\mathbf{x})$ , where  $P_{te}$  is the test distribution.

**Definition (prior probability shift).** *Prior probability shift* refers to the distribution change of the class variable  $y$  between training and test phases, i.e.,  $P_{tr}(\mathbf{x}|y) = P_{te}(\mathbf{x}|y)$  and  $P_{tr}(y) \neq P_{te}(y)$ .

**Definition (concept drift).** *Concept drift* refers to the change in the posterior probability distribution between training and test phases, i.e.,  $P_{tr}(y|\mathbf{x}) \neq P_{te}(y|\mathbf{x})$  and  $P_{tr}(\mathbf{x}) = P_{te}(\mathbf{x})$ .

Clearly, the most general data shift is:  $P_{tr}(y|\mathbf{x}) \neq P_{te}(y|\mathbf{x})$  and  $P_{tr}(\mathbf{x}) \neq P_{te}(\mathbf{x})$ .

However, there is another change not explicitly included in the above data shift, i.e., *novelty* or *novel instances* that appear in testing or deployment and do not belong to any known classes, which are the focus of SOL. Covariate shift may have novel instances belonging to known classes.

### 2.2 Novel Instances and Classes

Novelty is an agent-specific concept. An object may be novel to one agent based on its partial knowledge of the world but not novel to another agent. In the context of supervised learning, the world knowledge is the training data  $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  with  $\mathbf{x}_i \in X$  and  $y_i \in Y_{tr}$ . Let  $h(\mathbf{x})$  be the latent or internal representation of  $\mathbf{x}$  in the agent's mind,

$h(D_{tr}^i)$  be the latent representation of the training data of class  $y_i$ , and  $k$  ( $= |Y_{tr}|$ ) be the total number of training classes. We use  $\mu(h(\mathbf{x}), h(D_{tr}^i))$  to denote the novelty score of a test instance  $\mathbf{x}$  with respect to  $h(D_{tr}^i)$ . The degree of novelty of  $\mathbf{x}$  with respect to  $D_{tr}$ ,  $\mu(h(\mathbf{x}), h(D_{tr}))$ , is defined as the minimum novelty score with regard to every class,

$$\mu(h(\mathbf{x}), h(D_{tr})) = \min(\mu(h(\mathbf{x}), h(D_{tr}^1)), \dots, \mu(h(\mathbf{x}), h(D_{tr}^k))) \quad (1)$$

The novelty function  $\mu$  can be defined based on specific applications. For example, if the training data of each class follows the Gaussian distribution, one may use the distance from the mean to compute the novelty score.

**Definition (novel instance):** A test instance  $\mathbf{x}$  is novel if its novelty score  $\mu(\mathbf{x}, \cdot)$  is greater than or equal to a threshold value  $\gamma$  such that  $\mathbf{x}$  can be assigned a new class not in  $Y_{tr}$ .

In *covariate shift*, a novel instance may still be assigned to an existing class as the class assignment is application or agent specific. For example, we have a training class called *animal* and the learner has seen *dog* and *chicken* in the *animal* class, but during testing, a *tiger* shows up. In the covariate shift case, *tiger* can be added to the existing *animal* class, although *tiger* is novel as it has not been seen before.

**Definition (novel class):** A newly created class  $y_{new}$  ( $y_{new} \notin Y_{tr}$ ) assigned to some novel instances is called a *novel class* (*unknown* or *unseen class*). The classes in  $Y_{tr}$  are called *known* or *seen classes*.

Novelty is not restricted to the perceivable physical world but also includes the agent’s internal world, e.g., novel interpretations of world states or internal cognitive states that have no correspondence to any physical world state. Interested readers may also read (Boult et al. 2021) for a more nuanced and perception-based study of novelty.

There are several related concepts to novelty. We clarify their difference from novelty here.

**Outlier or anomaly:** An outlier is a data point that is far away from the main data clusters, but it may not be unknown. For example, the salary of a company CEO is an outlier with regard to the salary distribution of the company employees, but that is known and thus not novel. Unknown outliers are novel. Anomalies can be considered outliers or instances that are one off and never repeated. Though technically “novel” they may not need to result in a new class.

**Surprise or unexpectedness:** Based on the prior knowledge of the agent, the probability  $P(\mathbf{x}|Q)$  of  $\mathbf{x}$  occurring in a particular context  $Q$  is very low, but  $\mathbf{x}$  has occurred in  $Q$ , which is *surprising* or *unexpected*. If  $\mathbf{x}$  has been seen before, it is not novel. In human cognition, surprise is an emotional response to an instance which greatly exceeds the expected uncertainty within the context of a task. Outliers, anomalies, and novelty can all lead to surprise.

### 2.3 Definitions of Learning in SOL

The classic ML makes the *i.i.d* assumption, which is often violated in practice. Here we define several other assumptions and learning paradigms that are progressively more and more aligned with the real-world learning needs of SOL.

**Definition (closed-world assumption):** No new or novel classes appear in testing. Other types of data shift may occur.

**Definition (closed-world learning):** It refers to the learning paradigm that makes the closed-world assumption.

**Definition (learning with data shift):** It refers to the learning paradigm that deals with certain types of data shift in testing or deployment, but not new classes.

**Definition (open world learning):** It refers to the learning paradigm that can detect data shifts and novel instances in testing or deployment. The learner can learn the novel classes labeled by humans from the identified novel instances and update the model using the new data. The re-training or model updating is initiated by human engineers. That is, there is no self-initiated learning after model deployment.

Note that for covariate shift, the assignment of shifted data (or novel) instances to existing training classes is normally done by humans. For concept shift, the shifted classes are caused by humans or by the environment.

**Definition (self-initiated open world learning (SOL)):** It refers to the learning paradigm that has the capability of open-world learning and the agent is able to initiate a learning process by itself during application (after deployment) with no involvement of human engineers. The learning of the new classes is incremental, i.e., no re-training of previous classes. The process is thus lifelong or continuous, which makes the agent smarter and smarter over time.

Note that apart from learning, SOL’s performance task also demands characterization and adaptation, which we discuss in Section 2.5.

### 2.4 Steps in Learning in SOL

Learning in SOL involves the following three main steps. We ignore updating the existing model to deal with data shift because if we can deal with novel classes, it is relatively easy to deal with the other types of data shift (see Section 2.1).

**Step 1 - Novelty detection.** This step involves detecting data instances (1) whose classes do not belong to  $Y_{tr}$  or (2) have covariate shift, and it is done automatically. A fair amount of research has been done on this under open-set classification or out-of-distribution detection (Pang et al. 2021).

**Step 2 - Acquiring class labels and creating a new learning task on the fly:** This step first clusters the detected novel instances. Each cluster represents a new class. It may be done automatically or through interactions with humans using the interaction module  $I$ . Interacting with humans should produce more accurate clusters and also obtain meaningful class labels. If the detected data is insufficient for building an accurate model to recognize the new classes, additional ground-truth data may be collected via interaction with humans. A new learning task is then created.

In the case of our hotel greeting bot, since the bot detects a single new guest (automatically), no clustering is needed. It then asks the guest for his/her name as the class label. It also takes more pictures as the training data. With the labeled ground-truth data, a new learning task is created to incrementally learn to recognize the new guest on the fly.

The learning agent may also interact with the environment to obtain training data. In this case, the agent must have an *internal evaluation system* that can assign rewards to different states of the world, e.g., for reinforcement learning.

**Step 3 - Incrementally learn the new task.** After ground-truth training data has been obtained, the learner  $L$  incrementally learns the new task. This is *continual learning* (Chen and Liu 2018), an active and challenging ML research area, which is defined as learning a sequence of tasks incrementally.

## 2.5 Novelty Characterization and Adaptation

In a real-life application, classification may not be the primary task. For example, in a self-driving car, object classification supports its primary performance task of driving. To drive safely, the car has to take some actions to adapt or respond to the new object, e.g., slowing down and avoiding the object. In order to know what actions to take, it must characterize the new object. Based on the characterization, appropriate actions are formulated to accommodate or respond to the novel object. The response process may also involve learning.

**Definition (novelty and response):** It is a pair  $(c, r)$ , where  $c$  is the *characterization of the novelty* and  $r$  is the *response* to the novelty, which is a plan of dynamically formulated actions based on the characterization of the novelty and the agent's interactions with the novel item. If the system cannot characterize a novelty, it will take the *default response*. In some situations, the agent does not know what to do. The response is *Learn*, i.e., to learn the actions to take.

**Definition (characterization of novelty):** It is a description of the novelty based on the agent's existing knowledge of the world. According to the description, the agent chooses a specific course of actions to respond to the novelty.

Characterization of novelty can be done at different levels of detail, which may result in more or less precise responses. Based on the ontology and attributes related to the performance task, the description can be defined based on the *type of the object* and the *attribute of the object*. For example, for self-driving cars, when sensing a novel object, the car can identify both the movement of the object and the location of the object relative to the direction of travel—on the road or off the road. Thus, some classification of movement and location in this case is needed to characterize the novelty which, in turn, facilitates determination of the agent's responding action(s). For instance, if the novel object is a mobile object, the car may wait for the object to leave the road before driving.

Another common characterization strategy is to *compare the similarity* between the novel object and the existing known objects. For example, if it is believed that the novel object looks like a pig, then the agent may react like when it sees a pig on the road.

In our greeting bot example, when it can characterize a novelty as a new guest, its response is to say "Hello, welcome to our hotel! What is your name, sir?" If the bot has difficulty with object characterization, it can take a *default action*, either 'do nothing' or 'report to a hotel staff.' The set of responses are specific to the application. For a self-driving car, the default response to a novel object is to slow down the car as soon as possible so that it will not hit the novel object.

This discussion implies that in order to effectively characterize a novelty, the agent must already have a great of world knowledge that it can use to describe the novelty. Without such knowledge, it will not be able to characterize the novelty in meaningful ways.

The characterization and response process is often interactive in the sense that based on the initial characterization, the agent may choose a course of actions, but after some actions are taken, it will get some feedback from the environment. Based on the feedback and the agent's additional observations, the course of actions may be revised.

**Focus of attention.** Due to the performance task, the agent should focus on detecting and characterizing novelties that are critical to the performance task. For example, in the self-driving car application, the agent should focus on novel objects or events that are or may potentially appear on the road in front of the car. It should not pay attention to novel objects in the shops along the street as they do not affect driving. In characterizing a novel object on the road, it should focus on those aspects that are important to driving, i.e., whether it is a still or a moving object. If it is a moving object, the agent must determine its direction of moving.

**Learning to respond.** As indicated above, in some situations, the system may not know how to respond to a new object or situation. In this case, it needs to learn by itself (see Section 2.4). There are many ways to learn, e.g.,

(1) *Asking a human.* In the case of the self-driving car, when it does not know what to do, it may ask the passenger using the interactive module  $I$  (e.g., in natural language) and then follow the instruction and also remember or learn it for future use. For example, if the car sees a black patch on the road that it has never seen before, it can ask "what is that black thing in front?" The passenger may answer "that is tar." In the case of an unrecognized response, such as no prior information on tar, the system may progress with further inquiry, asking the passenger "what should I do?"

(2) *Imitation learning.* On seeing a novel object, if the car in front drives through it with no issue, the car may choose the same course of action as well and learn it for future use.

(3) *Performing reinforcement learning.* By interacting with the environment through trial and error exploration, the agent learns a good response policy for future use. As mentioned earlier, in this case the agent must have an internal evaluation system that can assign rewards to environment states.

If multiple novelties are detected at the same time, it is more difficult to respond as the agent must reason over the characteristics of all novel objects to dynamically formulate an overall plan of actions that prioritize the responses.

## 2.6 Risk

There is risk in achieving performance goals of an agent when making an incorrect decision. For example, classifying a known guest as unknown or an unknown guest as known may negatively affect guest impressions resulting in negative reviews. For a self-driving car, misidentifications can result in wrong responses, which could be a matter of life and death. Thus, risk assessment must be made in making each decision. Risk assessment can also be learned from experiences or mistakes. In the example of a car passing over tar, after the experience of passing over shiny black surfaces safely many times, if the car slips in one instance, the car agent must assess the risk of continuing the prior procedure. Given the danger, a car may weight the risk excessively, slowing down on new encounters of shiny black surfaces.

### 3 An Example SOL System

Although novelty detection (Yang et al. 2021; Pang et al. 2021) and incremental or continual learning (Chen and Liu 2018; Parisi et al. 2019) have been studied widely, little work has been done to build a SOL system. Here we describe a dialogue system (called CML) that is based on the SOL framework and performs each function in SOL continually by itself after the system has been deployed (Mazumder et al. 2020; Liu and Mazumder 2021).

CML is a natural language interface like Amazon Alexa and Siri. Its *performance task* is to take a user command in natural language (NL) and perform the user requested API action in the underlying application. There is no support function in this application. The key issue is how to understand paraphrased NL commands from the user in order to map a user command to a system’s API call. *Novelty* equates to the system’s failure in understanding a user command. After the system automatically detects a novelty (a hard-to-understand user command), it will try to understand the command and also learn the command so that it will be able to understand it and related commands in the future. The novelty *characterization* step of SOL here is to identify the part of the command that the system does not understand. Based on the characterization, the system *adapts* by asking the user via an interactive dialogue to obtain the ground truth API action requested by the user, which also serves as a piece of training data for *continual learning*. In the adaptation or accommodation process, *risk* is also considered.

Consider the following example. The user issues the command “*turn off the light in the kitchen*” that the system does not understand (i.e., a **novelty**). Based on the current system state, it decides which part of the command it can understand and which part it has difficulty (i.e., **characterization**). Based on the characterization result, it provides the user a list of *top-k* predicted actions (see below) described in NL and asks the user to select the most appropriate action from the given list (i.e., **adaptation**).

---

**Bot:** Sorry, I didn’t get you. Do you mean to:

- option-1.** switch off the light in the kitchen, or
  - option-2.** switch on the light in the kitchen,
  - option-3.** change the color of the light?
- 

The user answers the desired action (option-1). The action API [say, `SwitchOffLight(arg:place)`] corresponding to the selected action (option-1) is retained as the ground truth action for the issued NL command. In subsequent turns of the dialogue, the agent can also ask the user questions to acquire ground truth values associated with arguments of the selected action, as defined in the API. CML then **incrementally learns** to map the original command “*turn off the light in the kitchen*” to the API action, `SwitchOffLight(arg:place)`. This learning ensures that in the future the system will not have problem understanding the related commands.

**Risk** is considered in the system in two ways. First, it does not ask the user too many questions in order not to annoy the user. Second, when the characterization is not confident, the system simply asks the user to say his/her command again rather than providing a list of random options to choose from.

### 4 Key Challenges

Although novelty detection and continual learning have been researched extensively (Yang et al. 2021; Pang et al. 2021), they remain to be challenging. Limited work has been done to address the following (this list is by no means exhaustive):

**Obtaining training data on the fly.** One key feature of SOL is interaction with humans to obtain ground-truth training data, which needs a dialogue system. Building an effective dialogue system is very challenging. We are unaware of any such system for SOL except CML (Mazumder et al. 2020), but CML is only for simple command learning.

**Few-shot continual learning.** It is unlikely for the learning agent to collect a large volume of training data via interaction with the user. Then, an effective and accurate few-shot continual/incremental learning method is necessary.

**Novelty characterization.** This is critical because it defines the characteristics used to recognize world state and determine the best response strategy. For example, if a self-driving car encounters a novel/new mobile object, its response will be different from encountering an immobile object. Even if we know it is a mobile object, the system may also need to know which direction it is moving and the moving speed in order to formulate an appropriate response. The challenge is that a large number of classifiers or other models may need to be built. This means that the system or agent must have a very rich world model and a large amount of knowledge related to its performance task before it can effectively characterize novelties.

**Learning to respond.** This task is especially challenging in a real-world physical environment (Dulac-Arnold et al. 2021). For example, due to safety concerns, learning during driving (which is required by SOL) by a self-driving car using reinforcement learning (RL) is very dangerous. Furthermore, for RL to work, the agent must have a highly effective internal reward or evaluation system to assign rewards to actions and states and to be aware of *safety constraints*. Little work has been done so far.

**Knowledge revision.** It is inevitable that the system may misinterpret, generalize or otherwise assemble incorrect knowledge. A SOL system must have a mechanism to detect and revise the inaccurate knowledge on its own. Again, little work has been done.

### 5 Conclusion

A truly intelligent system must be able to learn autonomously and continually in the open world on its own initiative after deployment in order to adapt to the ever-changing world and gain more and more knowledge to become more and more powerful over time. This paper proposed a *self-initiated open world learning* (SOL) framework for the purpose, and presented the concepts, steps and key challenges. An example SOL system is also described. We believe that the future research in SOL will bring ML and AI to the next level.

### Acknowledgments

This paper benefited greatly from numerous discussions in the DARPA SAIL-ON Program PI meetings. This work was supported in part by a DARPA Contract HR001120C0023. Bing

Liu and Sahisnu Mazumder are also partially supported by two National Science Foundation (NSF) grants (IIS-1910424 and IIS-1838770), and a Northrop Grumman research gift. The views expressed in this document are those of the authors and are not those of funders.

## References

- Bendale, A.; and Boulton, T. 2015. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1893–1902.
- Boulton, T.; Grabowicz, P.; Prijatelj, D.; Stern, R.; Holder, L.; Alseptor, J.; Jafarzadeh, M.; Ahmad, T.; Dhamija, A.; Li, C.; et al. 2021. Towards a Unifying Framework for Formal Theories of Novelty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15047–15052.
- Chen, Z.; and Liu, B. 2018. *Lifelong machine learning*. Morgan & Claypool Publishers.
- Dulac-Arnold, G.; Levine, N.; Mankowitz, D. J.; Li, J.; Paduraru, C.; Gowal, S.; and Hester, T. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning* 1–50.
- Esmailpour, S.; Liu, B.; Robertson, E.; and Shu, L. 2022. Zero-Shot Out-of-Distribution Detection Based on the Pre-trained Model CLIP. In *Proceedings of the AAAI conference on artificial intelligence*.
- Fei, G.; and Liu, B. 2016. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 506–514.
- Fei, G.; Wang, S.; and Liu, B. 2016. Learning cumulatively to become more knowledgeable. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1565–1574.
- Langley, P. 2020. Open-world learning for radically autonomous agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13539–13543.
- Liu, B. 2020. Learning on the job: Online lifelong and continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2020)*.
- Liu, B.; and Mazumder, S. 2021. Lifelong and continual learning dialogue systems: learning during conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2021)*.
- Mazumder, S.; Liu, B.; Wang, S.; and Esmailpour, S. 2020. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. *NeurIPS-2020 Workshop on Human in the Loop Dialogue Systems*.
- Moreno-Torres, J. G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N. V.; and Herrera, F. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45(1): 521–530.
- Murty, V. P.; Ballard, I. C.; Macduffie, K. E.; Krebs, R. M.; and Adcock, R. A. 2013. Hippocampal networks habituate as novelty accumulates. *Learning & Memory* 20(4): 229–235.
- Pang, G.; Shen, C.; Cao, L.; and Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54(2): 1–38.
- Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*.
- Parmar, J.; Chouhan, S. S.; and Rathore, S. S. 2021. Open-world Machine Learning: Applications, Challenges, and Opportunities. *arXiv preprint arXiv:2105.13448*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.
- Shu, L.; Xu, H.; and Liu, B. 2018. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*.
- Tulving, E.; and Kroll, N. 1995. Novelty assessment in the brain and long-term memory encoding. *Psychonomic Bulletin & Review* 2(3): 387–390.
- Xu, H.; Liu, B.; Shu, L.; and Yu, P. 2019. Open-world learning and application to product classification. In *The World Wide Web Conference*, 3413–3419.
- Yang, J.; Zhou, K.; Li, Y.; and Liu, Z. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.