Intelligent Computing
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Surfing Information: The Challenge of Intelligent Decision-Making

Chenyang Wu, and Zongzhang Zhang*

National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China.

*Address correspondence to: zzzhang@nju.edu.cn

Reinforcement learning (RL) is indispensable for building intelligent decision-making agents. However, current RL algorithms suffer from statistical and computational inefficiencies that render them useless in most real-world applications. We argue that high-value information in the real world is essential for intelligent decision-making; however, it is not addressed by most RL formalisms. Through a closer investigation of high-value information, it becomes evident that, to exploit high-value information, there is a need to formalize intelligent decision-making as bounded-optimal lifelong RL. Thus, the challenge of achieving intelligent decision-making is summarized as effectively surfing information, specifically regarding handling the non-IID (independent and identically distributed) information stream while operating with limited resources. This study discusses the design of an intelligent decision-making agent and examines its primary challenges, which are (a) online learning for non-IID data streams, (b) efficient reasoning with limited resources, and (c) the exploration–exploitation dilemma. We review relevant problems and research in the field of RL literature and conclude that current RL methods are insufficient to address these challenges. We propose that an agent capable of overcoming these challenges could effectively surf the information overload in the real world and achieve sample- and compute-efficient intelligent decision-making.

Citation: Wu C, Zhang Z. Surfing Information: The Challenge of Intelligent Decision-Making. *Intell. Comput.* 2023;2:Article 0041. https://doi.org/10.34133/icomputing.0041

Submitted 27 March 2023
Accepted 29 May 2023
Published 6 July 2023

Introduction

Reinforcement learning (RL) is a fundamental learning paradigm for studying the interactive decision-making of an agent in an environment to maximize its interest, as measured by cumulative rewards [1]. It has been hypothesized that rewards alone are sufficient to enable the development of a diverse range of abilities exhibited by natural and artificial intelligence, including knowledge representation, learning, perception, social interaction, language, generalization, and imitation [2]. However, in its current form, RL cannot make this grand vision a reality.

Current RL approaches primarily rely on large-scale trial-and-error learning, as opposed to reasoning in light of limited experience as humans do. For example, DeepMind's general RL algorithm MuZero [3] trains on 1 million mini-batches of size 2,048 in board games, where each sample is collected at a computational cost of 800 simulations. One might argue that this wastefulness is caused by the inefficiency of the current algorithms. However, the learning theory suggests that complex situations exist for nonlinear problems for which the sample size required by RL increases exponentially [4,5]. Sample inefficiency occurs because in classic domains such as Go and Atari games [6], the agent must collect information through trial and error, which generally reveals little information about the optimal strategy. Moreover, even if we have the information required to identify the optimal strategy, searching for the optimal policy (specifically, planning) remains a computationally daunting task. For instance, we have all the information needed to derive an optimal policy for Go once the rules of the game are revealed to us. However, understanding the rules is by no means equivalent to playing Go flawlessly. Mastering Go is intellectually onerous and computationally intensive. The computational inefficiency of planning is an insurmountable hindrance [7]. Both statistical and computational inefficiencies render general RL from scratch impractical.

For an agent to learn as efficiently as humans, it is crucial to give the agent access to high-value information such as manuals, tutorials, and demonstrations. Such information can inform the agent about how the world works, how to act, how to reason, how to evaluate, how to explore, etc. For example, a Go-playing agent may observe instructions stating, "A solidly connected group of stones is removed from the board when all directly adjacent intersections are occupied by the enemy," and "A fundamental Go strategy involves keeping the stones connected." These instructions provide informational and computational benefits, respectively, to intelligent agents. The former can dramatically reduce the need for trial and error in determining the rule. The latter structures the Go-playing problem in terms of the connectedness of stones and enables efficient reasoning at this more abstract level. In addition, it provides a good starting point and heuristic for policy search.

As demonstrated above, the existence of high-value information in observations helps in overcoming both statistical and computational inefficiencies, and makes it possible to improve policy purely through observation rather than interaction. In human society, individuals acquire knowledge of the world and learn to act primarily by reading, hearing, watching, and contemplating. Each benefits from human civilization, and everyone stands on the shoulders of giants. Learning from information-rich observations is the fundamental ability needed to solve intricate real-world tasks that are otherwise intractable. This is the central problem in building an autonomous intelligent agent [8].

Notably, the high-value information we demonstrate exhibits 2 distinct features. First, high-value information is inherently not independent and identically distributed (IID). The IID assumption assumes a fixed unknown probability distribution, and a new sample from the distribution reveals historically independent information. However, high-value information does not follow the same distribution as past observations. Moreover, the underlying process for generating these pieces of information involves complex interactions and dependencies that cannot be ignored. The present observation should not be treated with disregard for history; in fact, the meaning carried by high-value information is comprehensible only when due consideration is given to past information. In the example of Go, knowledge of the language and common sense is essential for understanding the declared rules and high-level strategies. Methods that ignore dependencies fail. Second, certain high-value information is beneficial only for computationally aware agents. An agent with unlimited compute can safely ignore the second Go instruction concerning high-level strategy and derive the optimal strategy purely from the rules of Go at the level of primitive states and actions. All high-level abstractions are subject to a certain degree of inaccuracy and should be abandoned by such an agent. Only a computationally aware agent has the potential to appreciate the value of computationally beneficial information and to make trade-offs between accuracy and computational cost.

These novel features demand a formalization of intelligent decision-making that captures the non-IID nature of the agent's observation stream and is computationally aware. Inspired by [2,9,10], we formalize intelligent decision-making as bounded-optimal lifelong RL in the "The formalization of intelligent decision-making" section.

After introducing a formalism, we discuss the design of an intelligent agent in terms of 3 indispensable parts: knowledge, reasoning, and a goal. These components are shown in Figure. We identified 3 fundamental problems in agent design.

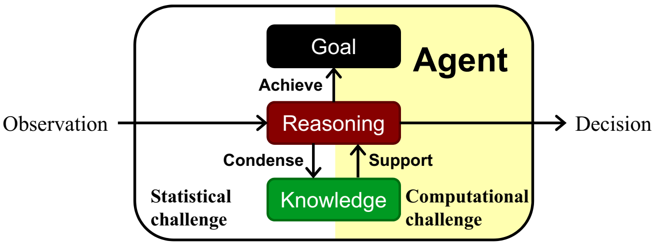

Fig. 1. Generic diagram for agent design. An agent can be understood in terms of 3 parts working together: the knowledge part, the reasoning part, and the goal. The knowledge part retains everything the agent has learned from its entire history. The reasoning part is a computational process that processes newly arrived information based on the learned knowledge. It helps condense information into various kinds of knowledge and concludes the actions the agent should take. The goal is something that orients the reasoning process and is aligned with the agent's lifelong interests.

- The first problem is overcoming the non-IID nature of the information stream and obtaining knowledge on the fly.
- The second problem is to support efficient reasoning given bounded resources.
- The third problem concerns determining the goal of reasoning such that the agent seeks a long-term return and avoids being hooked by short-term interests. This is known as the exploration–exploitation dilemma.

In the "Design of the agent" section, these issues are discussed in detail. Related problems and related research are discussed in the "Related problems and research" section. Finally, the deficiencies of contemporary RL are discussed in the "The deficiency of the contemporary RL" section. All symbols we used are listed in Table.

Materials and Methods

The formalization of intelligent decision-making

In the "Formalization" section, we formalize intelligent decision-making by considering the non-IID nature of the observation stream and limited computation. The relationship between this formalism and the conventional RL formalisms is discussed in the "Relation to conventional RL" section.

Formalization

Intelligent decision-making involves the interaction of an environment $\mathcal{E}$ and an agent $\mathfrak{A}$. In particular, we consider the case in which an interaction occurs in discrete time steps. An environment $\mathcal{E}$ is a tuple $(\mathcal{A}, \mathcal{O}, \rho)$, where $\mathcal{A}$ represents the action space, $\mathcal{O}$ denotes the observation space, and ρ represents the observation model. The observation model specifies that $\rho(o_{t+1}|h_t)$ represents the probability of an event in which, given the interaction history h_t, the agent receives the observation $o_{t+1} \in \mathcal{O}$. Herein, h_t denotes the interaction history until time t, $h_t = (a_1, o_1, \ldots, a_t, o_t)$, and $a_t \in \mathcal{A}$ and $o_t \in \mathcal{O}$ represent action and observation at time t, respectively. An agent $\mathfrak{A}$ is a function that maps the interaction history h_t to a distribution of actions. The probability of executing an action a_{t+1} at time $t+1$ is $\mathfrak{A}\left(a_{t+1}|h_t\right)$.

To establish the computing constraint, the interaction between the agent and the environment is set to occur in real-time with a fixed interval, and the agent function is constrained to the set C of functions implementable in a real machine.

It is the job of the agent designer to design an agent $\mathfrak{A} \in C$ such that the performance measured by the agent's expected to return $V_{\mathfrak{A}} = \mathbb{E}_{h_T \sim \mathfrak{A},\rho}\left[\sum_{t=1}^{T} R\left(h_t\right)\right]$ is maximized, where the expectation is taken with respect to the interaction between the agent $\mathfrak{A}$ and the observation model ρ of the environment $\mathcal{E}$, the reward function R encodes the agent's preference for histories, and T denotes the potentially random lifespan of the agent. This criterion is called bounded optimality [9,11].

High-value information is indispensable for making intelligent decision-making tractable despite limited compute. This can be characterized by the value of information (VOI):

$$\mathrm{VOI}\left(o_{t+1}|h_t\right) = V_{\mathfrak{A}}\left(h_t, o_{t+1}\right) - V_{\mathfrak{A}}\left(h_t\right), \tag{1}$$

where$V_{\mathfrak{A}}\left(h_t\right) = \mathbb{E}_{h_T \sim \mathfrak{A},\rho}\left[\sum_{i=t+1}^{T} R\left(h_i\right)|h_t\right]$,and$V_{\mathfrak{A}}\left(h_t, o_{t+1}\right) = \mathbb{E}_{a_{t+1} \sim \mathfrak{A}(\cdot|h_t)}\left[V_{\mathfrak{A}}\left(h_t, a_{t+1}, o_{t+1}\right)\right]$. The VOI depends on the agent's history and is affected by the agent's information

Table. Table of symbols.

Symbol	Description
$\mathcal{E}$	Environment
$\mathfrak{A}$	Agent
$\mathcal{A}$	Action space
$\mathcal{O}$	Observation space
ρ	Observation model
$\mathfrak{A}$	Agent
a_t	Action at time t
o_t	Observation at time t
h_t	Interaction history until time t
$\mathcal{C}$	Set of implementable functions
R	Reward function
T	Lifespan
VOI	Value of information
$V_{\mathfrak{A}}$	Agent's expected return
$\mathcal{S}$	State set
μ	Initial state distribution
$o_{\varnothing}$	Empty observation
P	State transition model
Z	State observation model
$\mathcal{T}$	Task space
$\mathfrak{T}$	Task distribution
$\mathbb{N}$	Set of natural number
δ	Dirac delta function

processing capabilities. An agent designed for intelligent decision-making should be able to exploit high-value information if it exists.

Unfortunately, this formalism provides little information on how an agent should be designed. Although the maximization $\max_{\mathfrak{A}\in C} V_{\mathfrak{A}}$ constitutes a proper optimization problem, it does not seem amenable to human effort because the space of agents is vast. Only miraculous nature could ever complete such a magnificent feat and create intelligent agents such as *Homo sapiens*. In the "Design of the agent" section, we discuss agent design at a higher level, focusing on learning and utilization of knowledge.

Relation to conventional RL

The formalism of intelligent decision-making considers all previous RL settings as special cases. The closest formalism is that adopted in [2], with the sole difference being that, in our formalism, the agent makes decisions at time t based solely on the history before time t. The decision is delayed because any computation is time-consuming.

The paradigm of continual (lifelong) RL [12] is closely related to our formalism. It addresses the challenge of non-IID learning and recognizes the limitations of computation. However, a major difference exists. Instead of aiming for bounded optimality, this approach seeks calculative rationality, which approximates the optimal solution given limited computing power [9]. Calculative rationality does not fully acknowledge the sequential nature of reasoning, and limited computational resources are not organized toward the agent's long-term interests. To illustrate this difference, we consider an example in which an agent is requested to answer 2 mathematical questions sequentially. Each question has a time limit of 1 min, and both are revealed to the agent at the beginning. Although the 2 questions are equally challenging, the payoff is low for the first question and high for the second. The optimal strategy, which a lifelong learning agent seeks to approximate, is to answer both questions perfectly, whereas a bounded optimal agent may intentionally sacrifice its performance on the first question to boost its performance on the second. Similar trade-offs appear repeatedly for lifelong learning agents with limited compute.

The reward function in our formalism is deterministic; however, this is not a strong restriction. To model stochastic reward RL, all we need is to treat the stochastic reward as an extra dimension of the observation, that is, to concatenate the reward and observation to form a new observation.

To model episodic RL, we can reset the environment periodically such that $\rho(o_{t+1}|h_t)=\rho(o_{t+1}|h_{r:t})$, where r is the latest resetting time, which is initially 1, and $h_{r:t}$ denotes the partial history $(a_r, o_r, \ldots, a_t, o_t)$. In our formalism, the environment and agent put forward their responses to each other simultaneously. However, this does not preclude alternate responses, as in conventional RL. The agent and environment are free to agree on certain interactive patterns. For example, we may allow the environment to respond to an agent only after it receives an action, and the agent is allowed to take an action only after receiving a new observation. In this manner, the agent and environment respond alternately.

In the following, we reduce common RL settings to intelligent decision-making. For simplicity, we assume that the agent acts on even time steps and the environment responds on odd time steps.

- RL in Markov decision processes (MDPs): An MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mu, P, R)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, μ is the initial state distribution, and $\mu(s_1)$ is the probability of $s_1 \in \mathcal{S}$ in the first time step. For $n \in \mathbb{N}$, P is the state transition model, $P(s_{2n+3}|s_{2n+1}, a_{2n+2})$ is the transition probability for states $s_{2n+3}, s_{2n+1} \in \mathcal{S}$ and action $a_{2n+2} \in \mathcal{A}$, and $R(s_{2n+1}, a_{2n+2})$ is the reward of taking action a_{2n+2} at state s_{2n+1}. Let $\delta(\cdot)$ be the Dirac delta function and $o_{\varnothing}$ be a special observation standing for an empty observation. Setting $\mathcal{O} := \mathcal{S}$, $\rho(o_1|h_0) := \mu(o_1)$, $\rho(o_{2n+2}|h_{2n+1}) := \delta(o_{2n+2} - o_{\varnothing})$, and $\rho(o_{2n+3}|h_{2n+2}) = \rho(o_{2n+3}|a_{2n+2}, o_{2n+1}, h_{2n}) := P(o_{2n+3}|o_{2n+1}, a_{2n+2})$ for $n \in \mathbb{N}$, we transform the MDP into an environment $(\mathcal{A}, \mathcal{O}, \rho)$. The history-dependent reward function is similarly constructed by that of the MDP, $R(h_{2n+2}) := R(s_{2n+1}, a_{2n+2})$. Therefore, RL in MDPs is reduced to intelligent decision-making.
- RL in partially observable Markov decision processes (POMDPs): A POMDP is defined as a tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mu, P, Z, R)$, where Z is the state observation model specifying the probability of observing o_{2n+1} at state s_{2n+1} as $Z(o_{2n+1}|s_{2n+1})$, and the reward function R maps interaction histories to a reward. Setting $\rho(o_1|h_0) := \int \mu(s_1)Z(o_1|s_1)\mathrm{d}s_1$, $\rho(o_{2n+2}|h_{2n+1}) := \delta(o_{2n+2} - o_{\varnothing})$, and

$$\rho\left(o_{2n+1}|h_{2n}\right) \propto$$

$$\int \mu\left(s_1\right) Z\left(o_1|s_1\right) \prod_{i=1}^{n} P\left(s_{2i+1}|s_{2i-1}, a_{2i}\right) Z\left(o_{2i+1}|s_{2i+1}\right) \mathrm{d}s_1 \mathrm{d}s_3 \cdots \mathrm{d}s_{2n+1} \tag{2}$$

for $n \in \mathbb{N}$, we get an environment $(\mathcal{A}, \mathcal{O}, \rho)$ out of the POMDP. Hence, we reduce the RL in POMDPs to an intelligent decision-making problem.

- Meta RL: In meta RL, the agent solves a series of MDP tasks defined by $(\mathcal{S}, \mathcal{A}, \mu, P, \mathcal{T}, \mathfrak{T}, R)$, where $\mathcal{T}$ is the task space, $\mathfrak{T}$ is a distribution on $\mathcal{T}$, and $R(s_{2n+1}, a_{2n+2})$ is the reward of the state-action pair (s_{2n+1}, a_{2n+2}). In contrast to normal MDPs, the initial state distribution μ and the transition function P take the task $t \in \mathcal{T}$ as an extra variable that is unobservable for the agent. In meta RL, we sample a new task from the task distribution $\mathfrak{T}$ every $2k$ time steps. After sampling, the environment is reset by the initial state distribution $\mu(\cdot | t)$ of the sampled task t. Let $\mathcal{O} := \mathcal{S}$. For even time step $2n + 2$ $(n \in \mathbb{N})$, we set $\rho(o_{2n+2} | h_{2n+1}) := \delta(o_{2n+2} - o_{\varnothing})$. The odd time steps $2n + 1$ $(n \in \mathbb{N})$ can be reexpressed as $m + 2l$, where $m = (\lceil (2n + 1)/2k \rceil - 1)2k + 1$ is the first time step of the latest task, and $l = ((2n + 1) - m)/2$ is the times of transitions in the latest task. If $l = 0$, we set $\rho(o_{2n+1} | h_{2n}) = \rho(o_m | h_{2n}) := \int \mu(o_m | t) \mathfrak{T}(t) \mathrm{d}t$. If $l > 0$,

$$\rho(o_{2n+1} | h_{2n}) = \rho(o_{m+2l} | h_{2n}) \propto \int \mu(o_m | t) \prod_{i=1}^{l} P(o_{m+2i} | o_{m+2i-2}, a_{m+2i-1}, t) \mathfrak{T}(t) \mathrm{d}t. \tag{3}$$

Thus, meta RL is reduced to an intelligent decision-making problem.

- Transfer RL: Transfer RL generalizes meta RL by allowing arbitrary dependencies among tasks. The sequence of tasks t_1, t_2, … is defined by $(\mathcal{S}, \mathcal{A}, \mu, P, \mathcal{T}, \mathfrak{T}, R)$. The d-th task t_d is sampled from $\mathfrak{T}(t_d | t_{1:d-1})$, which depends on all previous tasks $t_{1:d-1} = (t_1, \ldots, t_{d-1})$. This formalization is general enough to include the case where we have a series of prespecified tasks. Let $\mathcal{O} := \mathcal{S}$, $\rho(o_{2n+2} | h_{2n+1}) := \delta(o_{2n+2} - o_{\varnothing})$, and

$$\rho(o_{2n+1} | h_{2n}) \propto \int \prod_{i=1}^{d} \mu(o_{m_i} | t_i) \mathfrak{T}(t_i | t_{1:i-1}) \prod_{j=1}^{l_i} P(o_{m_i+2j} | o_{m_i+2j-2}, a_{m_i+2j-1}, t_i) \mathrm{d}t_1 \cdots \mathrm{d}t_d, \tag{4}$$

where $n \in \mathbb{N}$, $d = \lceil (2n + 1)/2k \rceil$ is the number of tasks at time $2n + 1$, $m_i = (i - 1)2k + 1$ is the first time step of the i-th task, and

$$l_i = \begin{cases} k - 1 & \text{if } i < d \\ (2n + 1 - m_d)/2 & \text{if } i = d \end{cases} \tag{5}$$

is the times of transitions in the i-th task. This reduces the transfer RL problem to an intelligent decision-making problem.

Design of the agent

This section discusses the challenges faced by agent designers.

We begin with a universal diagram of an agent in 3 parts (Figure). The tripartite view of an agent is based on the field of knowledge representation and reasoning, which studies how knowledge can be represented and manipulated in an automated manner using reasoning programs [13]. In contrast to having a static knowledge base, as most studies in this field do, the knowledge of an agent develops gradually from an information stream. In addition, the reasoning of an agent has aftereffects on both internal knowledge and the external world.

From this perspective, we can summarize agent design into 3 problems: how reasoning helps in condensing the information stream into knowledge, how knowledge facilitates efficient reasoning given limited compute, and how the goal of reasoning maximizes lifelong return. These problems are discussed below.

Online learning of non-IID data stream

The first problem is condensing on the fly the information stream into knowledge that can be reused in the future. Notably, the interaction history is a data stream with serial dependency and nonstationarity generated by the history-dependent functions $\mathfrak{A}$ and ρ. If dependency and nonstationarity are arbitrarily complex, it becomes impossible to make reliable predictions based on historical data alone, and the future could simply be anything. Hence, we require an inductive bias that connects the past to the future, similar to the IID assumption in conventional machine learning [14]. This bias and others introduced by model classes, loss functions, and optimization procedures [15] are essential for generalization, as indicated by the no-free-lunch theorem [16].

In addition to the statistical challenge of handling non-IID data, computational constraints constitute another severe challenge. An agent with unlimited compute could simply encode the entire history into a knowledge base and reason using the raw data. However, given limited resources, continually remembering the entire history is not feasible because the ever-increasing demand for memory, as well as the implied time demand, will eventually exceed computing constraints. Hence, we require a knowledge representation and a corresponding online learning algorithm that organizes information in a structured manner, thereby facilitating the incremental incorporation of new information.

Efficient reasoning given bounded resources

Given limited compute, efficient reasoning is crucial to effective learning and decision-making, without which an agent necessarily fails both to learn the statistical regularity of observations in time and to respond promptly to the environment.

Reasoning depends on the knowledge part of an agent to provide sensation understanding, action recommendation, environmental transition prediction, and utility evaluation. This information is sufficient for conventional RL, but insufficient for reasoning under computing constraints. In addition, efficient reasoning demands a structured knowledge representation [14] that represents the problem at hand and suggests efficient reasoning approaches that exploit the problem structure.

Reasoning under computing constraints involves sequential decision-making that determines not only the course of action but also which information to keep, forget, or attend to, and how to process information and learn new things. This indicates the need for internal actions that regulate the reasoning process. The reasoning process for determining these internal actions is referred to as meta-level reasoning [17]. The agent designer determines the mechanisms of meta-level reasoning.

Moreover, sequentiality implies that the designed agent should be able to learn to reason through delayed feedback. This is possible in our formalism because the influence of computation is made explicit by modeling the interaction between the environment and the agent in physical time. An agent that learns to reason efficiently can conduct reasoning using its internal computing devices and external environment. For instance, calculators are used for mathematical calculations and the computing results are retrieved through observation.

Exploration–exploitation dilemma

As the expected lifetime return measures performance, the agent should be able to cautiously weigh foreseeable rewards against uncertain potential payoffs. This is referred to in RL as the exploration–exploitation dilemma, where exploration refers to moving around and acquiring knowledge of the world [18], and exploitation means executing the best available strategy given the existing information.

The information-theoretic perspective is appropriate only when the agent retains all the historical information and executes arbitrarily complex computations. When computing resources are constrained, the agent inevitably forgets some information and can only perform limited operations per time step. In this case, both exploration and exploitation refer to broader behaviors. The agent needs to not only explore the uncertain part of the environment but also learn new things with no foreseeable usage and process information in unanticipated ways. Correspondingly, exploitation refers to the execution of the currently best acting, learning, and information-processing strategies. This highlights the computational perspective on the exploration–exploitation dilemma. The computational perspective is related to but is more complicated than a similar dilemma presented in searching, planning, and optimization, where one trades off local improvement for global optimality.

Exploring an immensely complex environment thoroughly is impractical because the amount of information needed to fully identify the real environment is enormously large and infeasible to collect, and most exploratory efforts will be unprofitable. Thus, an intelligent agent should rely on its inductive bias for generalizing to the uncharted world. In this case, the information-theoretic perspective is less important and the computational perspective dominates.

Exploration and exploitation are conflicting forces in the reasoning process. Resolving the exploration–exploitation dilemma amounts to setting the reasoning goal properly such that it is aligned with the agent's long-term interests. However, our understanding of this problem remains scarce, particularly when considering the computational perspective.

Related problems and research

All 3 identified problems have been proposed and discussed in various studies using different terms. In this section, we discuss connections with previous research.

Learning from data streams

Learning from data streams with limited computational resources presents a important challenge [19,20]. Some forms of dependency and nonstationarity have been investigated, including delayed labeling and concept drift [20]; however, overall, the non-IID characteristic has not been treated in its full generality. Concerning limited compute, a recent study developed a theoretical framework for characterizing the learnability of data streams under resource constraints [21], which is still at a very primitive stage.

Temporal credit assignment

The non-IID problem has been concomitant with RL since its inception and is manifested by the temporal credit assignment problem. Temporal credit assignment requires the agent to understand the temporal dependency of an outcome on the sequence of decisions and credit decisions that contribute to the outcome [22]. In stochastic or partially observable environments, the agent should also properly identify the correlation between observations and the reward outcome because, sometimes, the outcome should be ascribed to randomness or uncontrollable factors.

Inductive bias

Inductive bias is a set of assumptions used by an agent to interpret data. It may be present in various forms, such as parameterization [23], a loss function [24], an optimizer [25,26], or training procedures [27]. Essentially, anything that affects the hypothesis choice apart from the data introduces an inductive bias. Any learning algorithm possesses some bias, and a proper bias is essential for good generalization, according to the notable no-free-lunch theorem [16]. The inductive bias of deep learning is considered to have contributed to its success [15].

Meta, transfer, and lifelong learning

Meta-learning involves learning algorithms that generalize across a distribution of tasks [28]. Transfer learning focuses on transferring knowledge implied in the source domain(s) and task(s) to improve decisions in the target domain(s) and tasks(s) [29]. Continual or lifelong learning involves the sequential learning of different tasks with or without task boundaries, in which previously learned knowledge is continually transferred to subsequent tasks [30,31]. Among these learning paradigms, meta-learning is slightly easier because the IID structure of the tasks can be exploited for model learning. Both transfer and lifelong learning encounter the challenge of learning from non-IID datasets. To transfer knowledge, an algorithm must shift its inductive bias based on historical data. The manner of shifting bias is an inductive bias that connects history to the future.

Another line of research is open-environment machine learning [32], which considers machine learning in an evolving open world. It addresses the non-IID evolution of a data stream by focusing on new emerging classes, decremental/incremental features, changing data distributions, and varied learning objectives.

Stability–plasticity dilemma

The stability–plasticity dilemma asks how to preserve learned knowledge while continuously learning new things [33]. It is related to the well-known catastrophic forgetting problem of deep learning caused by the catastrophic interference of information in the model [34]. Because of interference, when new data arrive, we cannot localize the update to a small fraction of the parameters, because most parameters require an update. A global update will inevitably erase some information. Therefore, we are left with no choice but to continue retraining on the entire dataset in case of forgetting.

Systematic generalization

Humans are excellent at manipulating combinations of primitive concepts, an ability that is referred to as systematic generalization [35]. It has been argued that deep learning requires the prefrontal cortex [36], an argument consistent with our view on the need for meta-level reasoning. However, we argue that a structured knowledge representation is also necessary because it provides primitives for the meta-reasoner to manipulate compositionally.

The deficiency of the contemporary RL

This section discusses contemporary deep RL methods from the viewpoint of intelligent decision-making and points out where the current methods are inadequate.

As mentioned, the information stream in RL is non-IID. However, instead of overcoming this challenge, deep RL finds a workaround by retaining the latest data, shuffling them, and fitting the value function irrespective of the non-IID characteristic [37]. The nonstationarity caused by the change in policy is ameliorated by keeping the latest data, and the shuffling alleviates the dependency. This technique, referred to as experience replay, works well and underlies the success of deep RL. However, it merely masks the inherent non-IID issue instead of directly solving it. Although existing methods extend this approach to handle continual RL and mitigate forgetting [38,39], the long-term serial dependency of the information stream is hardly addressed, which is vital for truly understanding non-IID data. In addition, these methods are unlikely to scale up because they do not organize information in a structured manner.

Some deep RL methods incorporate self-supervised learning to extract knowledge from observations and, to some extent, realize learning from observations. However, self-supervised learning relies on human-designed objectives for information extraction. Different self-supervised learning methods extract different types of information from observations, and there is no generic method that is consistently helpful in all tasks [40]. It is important for an intelligent agent to decide by itself what to learn, rather than relying on a fixed learning objective.

Most deep RL methods reason about optimal decisions using experience samples at the primitive level of observations, actions, and rewards. Model-based planning calculates a policy based on model-generated samples, whereas model-free methods are based on newly collected samples or samples in a replay buffer. This starkly contrasts with the abstract reasoning of humans and substantially limits the scalability and applicability of these deep RL methods. This failure reflects the inability to learn the problem structure from history through inductive learning or to reason by exploiting the structure.

A recent trend in offline RL exploits transformer architecture for decision-making [41]. Offline RL enables training with a larger dataset, and the transformer provides the agent with better generalization. This is certainly a promising research direction for circumventing the difficulties of online learning and exploration. However, without non-IID learning, the agent cannot fully understand the shared structure across domains and can only generalize to a limited extent. Without efficient reasoning, the agent cannot effectively organize its limited cognitive capability for problem-solving. The latter problem has recently attracted considerable attention [42].

Although a diverse set of exploration strategies has been designed for deep RL [43], the existing strategies are inherently information-seeking and can be considered variants or approximations of information-theoretic exploration methods that are provably efficient [44,45,46]. The theoretical guarantee of these information-theoretic exploration methods is relevant only when the agent's lifespan is sufficiently long to explore the hypothesis space thoroughly. However, the hypothesis space of deep learning models appears too large to be fully explored in an agent's finite life, which makes the theoretical guarantee vacuous. In addition, information-theoretic methods often do not consider limited computing resources, which further diminishes their validity.

Conclusion

This study clarifies the significance of high-value information in enabling computation- and sample-efficient intelligent decision-making. To effectively handle high-value information, we propose to formalize intelligent decision-making as bounded-optimal lifelong RL. An intelligent agent designed for this formalism is computationally aware, learns from a non-IID observation stream, and is capable of handling high-value information.

We discuss the design of intelligent agents and identify 3 main problems. We examine these problems from both informational and computational perspectives and compare them with related problems in the literature. We also highlight the inadequacy of contemporary RL in addressing these issues. While we present these problems from a top-down perspective, we do not provide solutions. A complementary bottom-up investigation of brain function could deepen our understanding and guide future research.

Acknowledgments

Funding: This work was supported by the National Key R&D Program of China (2022ZD0114804), the National Science Foundation of China (62276126), the Natural Science Foundation of Jiangsu (BK20221442), and the Fundamental Research Funds for the Central Universities (114380010). **Author contributions:** C.W. conceived the idea and laid the foundation for this research. His innovative thinking and insights were instrumental in shaping the direction of the project. Z.Z. provided valuable guidance and supervision throughout the research process. His insightful feedback greatly enhanced the quality of the manuscript. **Competing interests:** The authors declare that they have no competing interests.

Data Availability

The authors declare that there are no data involved in this article.

References

1. Sutton RS, Barto AG. *Reinforcement learning: An introduction*. Cambridge (MA): MIT Press; 2018.
2. Silver D, Singh S, Precup D, Sutton RS. Reward is enough. *Artif Intell*. 2021;299:Article 103535.
3. Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, Guez A, Lockhart E, Hassabis D, Graepel T, et al. Mastering Atari, go, chess and shogi by planning with a learned model. *Nature*. 2020;588 (7839):604–609.
4. Magureanu S, Combes R, Proutiere A. Lipschitz bandits: Regret lower bound and optimal algorithms. Paper presented at: Proceedings of the 27th Conference on Learning Theory; 2014 May 19; Barcelona, Spain.

5. Dong K, Yang J, Ma T. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. Paper presented at: Advances in Neural Information Processing Systems. 2021;**34**:26168–26182.
6. Bellemare MG, Naddaf Y, Veness J, Bowling M. The arcade learning environment: An evaluation platform for general agents. *J Artif Intell Res*. 2013;47:253–279.
7. Papadimitriou CH, Tsitsiklis JN. The complexity of Markov decision processes. *Math Oper Res*. 1987;12(3):441–450.
8. LeCun Y. A path towards autonomous machine intelligence version 0.9.2. Preprint posted on openreview; 2022 Jun 27. https://openreview.net/pdf?id=BZ5a1r-kVsf.
9. Russell S. *Fundamental issues of artificial intelligence*. Cham: Springer; 2016.
10. Lu X, Roy BV, Dwaracherla V, Ibrahimi M, Osband I, Wen Z. Reinforcement learning, bit by bit. ArXiv 2021. https://doi.org/10.48550/arXiv.2103.04047
11. Russell SJ, Subramanian D. Provably bounded-optimal agents. *J Artif Intell Res*. 1994;2(1):575–609.
12. Khetarpal K, Riemer M, Rish I, Precup D. Towards continual reinforcement learning: A review and perspectives. *J Artif Intell Res*. 2022;75:1401–1476.
13. Brachman R, Levesque H. *Knowledge representation and reasoning*. San Francisco (CA): Morgan Kaufmann; 2004.
14. Russell S, Norvig P. *Artificial intelligence: A modern approach*. Hoboken (NJ): Pearson; 2009.
15. Goyal A, Bengio Y. Inductive biases for deep learning of higher-level cognition. *Proc Society A: Math Phys Eng Sci*. 2022;478(226):Article 20210068.
16. Wolpert DH. The lack of a priori distinctions between learning algorithms. *Neural Comput*. 1996;8(7):1341–1390.
17. Russell S, Wefald E. Principles of metareasoning. *Artif Intell*. 1991;49(1):361–395.
18. Thrun S. *Efficient exploration in reinforcement learning*. Tech. Rep. CMU-CS-92-102Pittsburgh; PA: Carnegie Mellon University; 1992.
19. Gama J. *Knowledge discovery from data streams*. Boca Raton (FL): Chapman & Hall/CRC; 2010.
20. Bahri M, Bifet A, Gama J, Gomes HM, Maniu S. Data stream analysis: Foundations, major tasks and tools. *Data Min Knowl Disc*. 2021;11(3):Article e1405.
21. Zhou Z-H. Stream efficient learning. ArXiv 2023. https://doi.org/10.48550/arXiv.2305.02217
22. Sutton RS. Temporal credit assignment in reinforcement learning [thesis]. [Amherst (MA)]: University of Massachusetts Amherst; 1984.
23. Cohen T, Welling M. Paper presented at: Proceedings of the 33rd International Conference on Machine Learning; 2016 Jun 19–24; New York, USA.
24. Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J. Self-supervised learning: Generative or contrastive. *IEEE Trans Knowl Data Eng*. 2021;35(1):857–876.
25. Bartlett PL, Montanari A, Rakhlin A. Deep learning: A statistical viewpoint. *Acta Numerica*. 2021;30:87–201.
26. Smith SL, Dherin B, Barrett D, De S. On the origin of implicit regularization in stochastic gradient descent. Paper presented at: International Conference on Learning Representations; 2022; Virtual.
27. Xu Z-QJ, Zhang Y, Xiao Y. Training behavior of deep neural network in frequency domain. *Neural Inform Process*. 2019;264–274.
28. Hospedales T, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: A survey. *IEEE Trans Pattern Anal Mach Intell*. 2022;44(9):5149–5169.
29. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q. A comprehensive survey on transfer learning. *Proc IEEE*. 2021;109(1):43–76.
30. Silver DL, Yang Q, Li L. Lifelong machine learning systems: Beyond learning algorithms. In *Lifelong machine learning*; California, USA: AAAI; 2013.
31. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. *Neural Netw*. 2019;113:54–71.
32. Zhou Z-H. Open-environment machine learning. *Natl Sci Rev*. 2022;9(8):Article nwac123.
33. Carpenter G, Grossberg S. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*. 1988;21(3):77–88.
34. McCloskey M, Cohen NJ. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychol Learn Motiv*. 1989;24:109–165.
35. Fodor JA, Pylyshyn ZW. Connectionism and cognitive architecture: A critical analysis. *Cognition*. 1988;28(1):3–71.
36. Russin J, O'Reilly RC, Bengio Y. Deep learning needs a prefrontal cortex. Paper presented at: *ICLR Bridging AI and Cognitive Science (BAICS) Workshop*. 2020;107:603–616.
37. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing Atari with deep reinforcement learning. ArXiv 2013. https://doi.org/10.48550/arXiv.1312.5602
38. Isele D, Cosgun A. Selective experience replay for lifelong learning. *AAAI*. 2018;32(1):3302–3309.
39. Rolnick D, Ahuja A, Schwarz J, Lillicrap T, Wayne G. Experience replay for continual learning. Paper presented at: Advances in Neural Information Processing Systems; 2019;**32**.
40. Li X, Shang J, Das S, Ryoo MS. Does self-supervised learning really improve reinforcement learning from pixels?. Paper presented at: Advances in Neural Information Processing Systems; 2022;**35**:30865–30881.
41. Li W et al. A survey on transformers in reinforcement learning. ArXiv 2023. https://doi.org/10.48550/arXiv.2301.03044
42. Mialon G, Dessi, R, Lomeli M, Nalmpantis C, Pasunuru R, Raileanu R, Roziere B, Schick T, Dwivedi-Yu J, Celikyilmaz A, et al. Augmented language models: A survey. ArXiv 2023. https://doi.org/10.48550/arXiv.2302.07842
43. Ladosz P, Weng L, Kim M, Oh H. Exploration in deep reinforcement learning: A survey. *Inf Fusion*. 2022;85:1–22.
44. Azar MG, Osband I, Munos R. Minimax Regret Bounds for Reinforcement Learning. Paper presented at: Proceedings of the 34th International Conference on Machine Learning; 2017;**70**:263–272.
45. Jin C, Yang Z, Wang Z, Jordan MI. Provably efficient reinforcement learning with linear function approximation. Paper presented at: Proceedings of 33rd Conference on Learning Theory; 2020;**125**:2137–2143.
46. Wu C, Li T, Zhang Z, Yu Y. Bayesian optimistic optimization: Optimistic exploration for model-based reinforcement learning. Paper presented at: Advances in Neural Information Processing Systems; 2022;35:14210–14223.