frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 31 March 2023
DOI 10.3389/fams.2023.1000785

OPEN ACCESS

EDITED BY
Mohd Hafiz Mohd,
University of Science Malaysia (USM), Malaysia

REVIEWED BY
Hirohide Haga,
Doshisha University, Japan
Mohd Shareduwan Mohd Kasihmuddin,
University of Science Malaysia (USM), Malaysia

*CORRESPONDENCE
Meritxell Vinyals
✉ meritxell.vinyals@inrae.fr
Patrick Taillandier
✉ patrick.taillandier@inrae.fr

SPECIALTY SECTION
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

RECEIVED 22 July 2022
ACCEPTED 13 March 2023
PUBLISHED 31 March 2023

CITATION
Vinyals M, Sabbadin R, Couture S, Sadou L, Thomopoulos R, Chapuis K, Lesquoy B and Taillandier P (2023) Toward AI-designed innovation diffusion policies using agent-based simulations and reinforcement learning: The case of digital tool adoption in agriculture. *Front. Appl. Math. Stat.* 9:1000785. doi: 10.3389/fams.2023.1000785

Toward AI-designed innovation diffusion policies using agent-based simulations and reinforcement learning: The case of digital tool adoption in agriculture

Meritxell Vinyals[1]*, Regis Sabbadin[1], Stéphane Couture[1], Loïc Sadou[1], Rallou Thomopoulos[2], Kevin Chapuis[3], Baptiste Lesquoy[4,5] and Patrick Taillandier[1,4,5]*

[1]UR 875, MIAT, INRAE, Toulouse University, Castanet Tolosan, France, [2]UMR 1208, IATE, Univ Montpellier, INRAE, Institut Agro, Montpellier, France, [3]UMR 228, ESPACE-DEV, IRD, Montpellier, France, [4]UMI 209, UMMISCO, IRD, Sorbonne Université, Bondy, France, [5]LMI ACROSS, Thuyloi University, Hanoi, Vietnam

In this paper, we tackle innovation diffusion from the perspective of an institution which aims to encourage the adoption of a new product (i.e., an innovation) with mostly social rather than individual benefits. Designing such innovation adoption policies is a very challenging task because of the difficulty to quantify and predict its effect on the behaviors of non-adopters and the exponential size of the space of possible policies. To solve these issues, we propose an approach that uses agent-based modeling to simulate in a credible way the behaviors of possible adopters and (deep) reinforcement learning to efficiently explore the policy search space. An application of our approach is presented for the question of the use of digital technologies in agriculture. Empirical results on this case study validate our scheme and show the potential of our approach to learn effective innovation diffusion policies.

KEYWORDS

innovation diffusion, policy design, reinforcement learning, agent-based simulation, deep reinforcement learning, digital agriculture

1. Introduction

Many areas such as agriculture have been transformed by the arrival of new innovations. These transformations are sometimes desired by institutions that wish to promote for instance more environmentally friendly models. In this paper, we propose to tackle this issue from the perspective of an institution which aims to encourage the adoption of some innovation. In particular, we focus on domains in which individuals are reluctant, for a variety of possible reasons, to adopt the innovation even when it can benefit them: misinformation, distrust of the hidden agenda of the institutions promoting it, lack of skills

to use it, etc. In this situation the institution may make use of different policy actions (information, financial aid, training,...) in order to overcome these barriers hindering the widespread adoption of the innovation. However, designing such innovation adoption policies is a very challenging task because of: (i) the difficulty to quantify and predict the effect of policies on the behaviors of non-adopters (e.g., the effect of an advertising campaign can be widespread over time and difficult to differentiate from the effect of other similar actions); and (ii) the exponential size of the policy search space. In more detail, when designing an innovation diffusion policy the institution faces a budget-constrained sequential decision making problem in which, at each time step, it needs to decide if it launches new (parallel) actions on several areas of interest as well as the parameters that accompany each of these actions. Therefore, it is not only which actions but when, with which combination and in which parameterization. Moreover, to perform such decisions the institution has only access to some aggregated indicators at global level (e.g., the number of adopters) and none at individual level, resulting in a very partial view of the state of the environment.

Against this background, we propose to study the problem of designing innovation diffusion policies using agent-based simulations and reinforcement learning.

On the one hand, a computational simulation environment is necessary given the limited opportunities to experiment with policy-making in the real-world. In this context, agent-based simulation has proven to be an effective tool to study the complex social dynamics that emerge in the diffusion of innovation among an heterogeneous population of potential adopters [1]. In particular, agent-based modeling addresses the limitations of aggregate models [2] by explicitly representing individuals, their social interactions, and their decision-making processes. The difficulty that remains is how to define credible behaviors for the agents while keeping the computational cost of the simulations affordable. In particular, in innovation diffusion we need to model the human decision-making key-factors deciding on innovation adoption. We tackle this problem by building an agent-based model of the innovation diffusion process based on the theory of planned behavior [3] to describe the adoption of a new behavior.

On the other hand, even when the institution is expected to be able to adapt the public policy in place to the context, most current works are content to represent it as a fixed variable or a set of scenarios to explore [4]. Instead, in this paper, the institution is modeled as an intelligent agent which learns, by reinforcement learning (RL), how to adapt the public policy over time. In particular, and as it is common in the literature when dealing with highly complex partially-observable environments, we based our approach on *deep* reinforcement learning, where deep stands for an artificial deep neural network (NN) that is used to approximate the policy function. Recent works have already shown the capacity of deep RL to automatically learn public policies, but on markedly different domains, e.g., on the problem of designing taxation [5], pandemic response [6] and market-price intervention [7] policies. However, to the best of our knowledge, no previous work has used (deep) RL to address the problem of designing public policies that maximize the number of adopters of an innovation, i.e., the so-called innovation diffusion policy design problem that we formalize in this paper. As we analyze in this work, it turns out that the action space of this problem is particularly complex due to the fact that: (i) an action is composed of a set of sub-actions, corresponding to different types of initiatives that the institution launches in parallel; (ii) each sub-action is parameterized by a set of (real-valued) parameters that accompany that action; (iii) actions in a given time are constrained by the available budget, shared among all sub-actions. These characteristics make standard deep RL methods not directly applicable to the innovation diffusion policy design problem. The problem can be cast as a Constrained Markov Decision Process (CMDP) and optimized by one of the general-purpose approaches which have been proposed to solve CMDPs [8]. However, such approaches typically negatively affect the performance and scalability (or both) of the learning compared with the non-constrained case. Given this, in this paper we instead opted for a specific architecture that is able to exploit the structure of the particular action space of this problem. More precisely, we propose a NN policy architecture in which the policy learns the budget allocation among the multiple action types along with the values of the parameters of those actions. All in all, our approach allows for AI-designed policies for innovation diffusion.

We illustrate the use of our approach on a particular application in the framework of digital technologies in agriculture and more particularly, on the adoption of communicating water meters by farmers in the Louts region (South-West of France). Nowadays, farmers in this area mainly use mechanical meters which poorly estimate water consumption due to a low accuracy. This is an advantage for farmers, as they are less likely to be overcharged if they exceed the allocated quota. Nevertheless, this over-consumption is an important issue in this region where the water level of the rivers tends to decrease every year due to climate change, which has major consequences for the local ecosystem. For this reason, the Ministry of the Environment has required a periodic refurbishment of the metering system every 9 years. The institution in charge of managing water distribution in this area is counting on this regulation to install its new communicating meters. These new meters are more accurate and allow for real-time monitoring of each farmer's consumption and thus better manage water use. However, the institution is having difficulty convincing farmers to install this device because they perceive it negatively. This obstacle is closely linked to the farmers' distrust of the institution. A large part of the farmers think that the new meter does not bring them anything and that it is only useful to the institution. Thus, the institution is now questioning the policy to be implemented in order to encourage the adoption of new meters.

Our main contributions in this article are the following:

- We propose an AI framework for the problem of designing effective innovation diffusion policies that combines agent-based simulations with (deep) reinforcement learning techniques;
- Our agent-based model for innovation diffusion combines for the first time three existing models ([4] for the theory of planned behavior to model the adoption of

the innovation, Deffuant et al. [9] and Deffuant et al. [10] for the bounded confidence model to represent the social influence among individuals and Sadou et al. [11] for the integration of several topics to build an opinion) with the objective to simulate the behavior of adopters in a credible way, while keeping the computational cost of simulations affordable;
- We model the institution as an intelligent agent that learns by (deep) reinforcement learning how to adapt the public policy in place to the particular observed context. The reinforcement learning mechanism of this institution agent extends previous approaches to deal with non mutually-exclusive parameterizable actions and budget constraints;
- We illustrate this generic model on the particular application of the adoption of digital tools in agriculture. We motivate this particular application with a real use case of the adoption of communicating meters by farmers in the Louts region (South-West of France);
- We provide a first validation of our approach showing, using simulations, how our institution agent can learn effective policies that promote the adoption of an innovation for this particular application.

The rest of this paper is structured as follows. Section 2 reviews the related work. Section 3 introduces our agent-based model of innovation diffusion, while Section 4 instantiates this model on the adoption of digital tools in agriculture. Section 5 details the reinforcement learning method used to search for an optimal policy and Section 6 presents our experimental validation. Finally, Section 7 draws conclusions and sets paths for future research.

2. Related work

Concerning the modeling of the innovation adoption process, many studies have tackled this topic with most of the models building on the work of Rogers [12] on the diffusion of innovations. Agent-based modeling is becoming increasingly popular for the study of this type of process [1], each agent representing an individual that can influence the others on their adoption of the innovation. As discussed in Kiesling et al. [1], aggregate models such as the classical Bass model [2] do not explicitly account for consumer heterogeneity and the complex dynamics of social processes involved in innovation diffusion. Agent-based modeling addresses this limitation by explicitly representing individuals, their social interactions, and their decision-making processes. Zhang and Vorobeychik [13] proposed a critical review of these agent-based models. In particular, they proposed to categorize these models based on how the models represent the decision to adopt. Among these categories, we can distinguish cognitive agent models that are closest to our concerns: they aim to explicitly represent how individuals influence each other in cognitive and psychological terms. A particularly popular model in this category is the *relative agreement* model of Deffuant et al. [9], which focuses on the notion of opinion about an innovation. The individual's opinion and uncertainties are represented by numerical values that evolve during interpersonal interactions. Other models are more interested in the adoption process as such: how an individual will decide whether or not to adopt an innovation. In this context, a classical approach is to base the decision of agents on the theory of planned behavior (TPB) [3]. This theory states that the intention to perform a behavior is a reliable predictor of the implementation of that behavior. The intention is derived from 3 factors: attitude, subjective social norm, and perceived behavioral control (PBC). The attitude represents the opinion that an individual has about the behavior. The subjective norm is the individual's perception of the adoption opinion of her/his social network. Finally, the PBC is the capacity felt by the individual to adopt the behavior (in terms of cost, time, skills...). A representative model of this use is that proposed by Bourceret et al. [14] concerning the adoption of more environmentally friendly agricultural practices. They propose a simple model based on the work of Beedell and Rehman [15] for the calculation of the intention from the attitude, the social norm and the PBC. In this model, the interaction between the agents is indirect through the social norm, there is no direct exchange between them. On the contrary, Sadou et al. [11] propose a more complex model, also based on the PBC, in which the agents try to convince each other explicitly through the exchange of arguments. In addition, this model introduces heterogeneity between agents by integrating the notion of point of view on different subjects (economy, environment...) which is used to compute the attitude of agents. While this model offers a powerful way of representing the change of opinion of agents, it requires a lot of data (arguments) and can be computationally heavy.

The model proposed is based on the three previous models: the model of Bourceret et al. [4] for its use of TPB for the decision making regarding the adoption of the innovation, the model of Deffuant et al. [9] and more particularly on the bounded confidence model [10] for the representation of the social influence between people, and finally the model of Sadou et al. [11] for the integration of several topics for the building of opinion.

Concerning the issue of governance representation in agent-based models, Bourceret et al. [4] have recently conducted a systematic review of the literature for socio-ecological issues. This review shows that in most works, governance is represented in the form of variables and not in the form of an agent: thus, if governance impacts the other agents, very few models take into account the fact that governance can be impacted by the other agents and in particular adapt the policy implemented according to the context. Generally, governance is just studied as a set of scenarios or parameters to be explored, which is not representative of most real contexts where the governance is able to adapt its policies according to the setting. Contrary to works like [4], we propose to represent it as an agent able to learn, by reinforcement learning (RL), how to adapt the policy to the context. Given the large body of work on reinforcement learning, on the remaining of this section we focus on the most relevant areas for our work, namely: deep RL for policy design and RL advances to deal with complex (e.g., parametrized, combinatorial, constrained, ..) action spaces.

Deep RL for policy design. Recently, some works [5–7] have also applied deep reinforcement learning to the problem of designing public policies. In Danassis et al. [7] a Deep RL policymaker agent adjusts the prices in a production market (e.g., the common-fishery market) to consider multiple objectives, including sustainability and resource wastefulness. Nevertheless, the output of this model is just a vector of continuous action values (i.e., corresponding to the price of each good in the next market round) and they do not consider any constraint on the budget of the policymaker (i.e., the cost of the policymaker intervention, defined as the difference between RL computed prices and the traditional competitive market prices, is not bounded). Zheng et al. [5] and Trott et al. [6] in the context of the so-called AI Economist frameworks applied Deep RL to the problem of designing taxation and pandemic response policies, respectively. However, in both works the policy outputs a single action discretized in intervals (so their setting ends up being a basic discrete action space) and do not consider any constraint on the cost of policy actions.

Parametrized action spaces. Several frameworks and algorithms [16, 17] have been proposed to deal with parametrized action spaces, in which the policy requires specified parameters associated to (discrete) action values. However, those works consider mutually-exclusive actions and the solution is typically built through a two-level decision making in which first the action to be applied is selected and, second, the parameters are defined for this action. Here we can not apply this type of approach since in our problem the institution can launch any combination of actions in parallel, i.e., at the same time step.

Combinatorial action spaces. Other works [18, 19] focused on action spaces where each action is a set of multiple interdependent sub-actions (i.e., non-mutually exclusive). However, in such works the complexity of the action space emerge from considering a large number of actions leading to an exponential combinatorial action space. Instead, in innovation diffusion policies the number of types of actions is typically small and the complexity rather emerges from the fact that actions are parametrized by a set of continuous parameters along with a budget constraint that creates interdependencies among them.

Bounded action spaces. Bounded action scenarios in discrete domains are typically addressed by action masking [20] whereas in continuous domains they are typically addressed by bounding the corresponding distributions [21]. However, given that the budget constraint is defined over every types of actions (creating interdependencies among them) we can not use individual bounding techniques here[1].

Constrained action spaces. Given the budget constraint, our work is also related to constrained policy learning. In Liu et al. [8], the problem of learning with constraints is modeled as a Constrained Markov Decision Process (CMDP). Existing approaches to solve CMDPs include enhancing: (i) the NN with an extra layer that projects actions onto a feasible space [22, 23] or (ii) the policy gradient algorithm itself [24–26]. However, given the general-purpose characteristic of these approaches (i.e., they can deal with any kind of constraint), they typically complexify the learning process, affecting its performance/scalability. Thus, even though the innovation diffusion policy design problem can be cast as a CMDP, in this work we take advantage of the fact that the budget constraint is explicit and known by the learner to exploit its structure and represent it in the policy network in a more expressive form (i.e., in our approach the policy network directly outputs the budget distribution among the multiple action types and once this is known we can act as in a bounded action space).

All in all, to the best of our knowledge, innovation diffusion policy design has not been tackled in the machine learning literature. Moreover, the complex action space of this problem, composed of multiple continuous parametrizable actions constrained by a common budget, argues for novel architectures that can deal with this action space efficiently.

TABLE 1 State variables of an *Individual* agent *i*.

State variable	Data type	Description
$social\_network^i$	List of *individual* agents-static	List of *individual* agents with whom *i* is in contact
$P^i_{interact}$	Float [0,1]-static	Probability of interaction with an *individual* agent on 1 day
$W^i_{attitude}$	Float [0,1]-static	Importance of the attitude in the intention computation
W^i_{social}	Float [0,1]-static	Importance of the social norm in the intention computation
W^i_{pbc}	Float [0,1]-static	Importance of the PBC in the intention computation
Ω^i	Float [0,1]-static	Adoption threshold
W^i_k	Float [0,1]-static	Importance of topic k
W^i_c	Float [0,1]-static	Importance of constraint c
$Adoption^i(t)$	Boolean-dynamic	Has the agent adopted the innovation at year $t \in \{1, .., H\}$?
μ^i	Float [0,1]-static	Speed of opinion convergence
d^i	Float [0,1]-static	Maximal opinion difference accepted for convergence
$I^i(t)$	Float [0,1]-dynamic	Current intention value
$A^i(t)$	Float [0,1]-dynamic	Current attitude value
$SN^i(t)$	Float [0,1]-dynamic	Current social norm value
$PBC^i(t)$	Float [0,1]-dynamic	Current PBC value
$Op^i_k(t)$	Float [0,1]-dynamic	Opinion on the topic k
$Skill^i_c(t)$	Float [0,1]-dynamic	Agent's capacity regarding a constraint c
$Support^i_k(t)$	Float [0,1]-dynamic	Value of the temporary support currently in place in relation to topic k
$Support^i_c(t)$	Float [0,1]-dynamic	Value of the temporary support currently in place in relation to constraint c

1 In our approach we use bounding to restrict action parameters to their feasible interval (which is independent of the state).

3. Agent-based model of innovation diffusion

This section presents the model following the Overview, Design concepts and Details (ODD) protocol [27].

3.1. Overview

3.1.1. Purpose and patterns

The model aims to assess the impact of the governance's policy regarding the adoption of an innovation. We evaluate our model by its ability to reproduce two patterns. The first one concerns the impact of interpersonal relation in the innovation diffusion process. Indeed, as stated in many works such as Rogers [12], the process of diffusion of innovation is partly due to interactions between people. The second concerns the capacity of institutions to promote the adoption of an innovation, as public policies can play an important role in the diffusion process [28].

3.1.2. Entities, state variables, and scales

Two types of entities are represented in the model: the possible adopters (*Individual* agents) and the *Institution* (unique agent). Tables 1, 2, respectively, present the state variables of the *Individual* and *Institution* entities.

Let H be the simulation length (total number of steps), then three time frames are taken into consideration (see Figure 1):

- Time frame of the interactions between the *Individual* agents (T^{ind})
- Time frame of the institution's action implementation (T^{inst})
- Time frame of the updating of the institution's budget (T^{budget})

There is no explicit representation of space.

3.1.3. Process overview and scheduling

Algorithm 1 details the pseudocode of the model. During the *Individual* agent interaction, each *Individual* agent i first has the probability $P^i_{interact}$ of interacting with another individual agent in $social\_network^i$ (randomly chosen) (lines 3–5), which will result in a potential convergence of their opinion on a topic (lines 6–7). In case of convergence, the *Individual* agents update their intention to adopt (line 8) and decide whether or not they wish to adopt the innovation (line 9). The details of the computation of the intention of the *Individual* agents and the interactions between them are given in Section 3.3.2. The agent adopts the innovation if the intention of the agent at time t, $I^i(t)$, is higher than the adoption threshold Ω^i:

$$\text{if } I^i(t) > \Omega^i\text{: } Adoption^i(t) = true \tag{1}$$

It is possible to consider two cases depending on the innovation: either an *Individual* agent cannot go back (once adopted, it keeps the innovation), or it can decide not to keep the innovation.

During the institution's action implementation time frame (line 11–13), first the supports of all agents will be set to *null*, then the *Institution* agent will choose which actions to implement. The actions can act directly on the *Individual* agents by permanently changing their opinion on a topic or their skills to handle a constraint, or they can be temporary (only during the period when the action is active). In the first case, the action will directly modify the opinion $Op^i_k(t)$ or skill $Skill^i_c(t)$ values of the agents at time t. In the second case, it will modify the support on a topic, $Support^i_k(t)$, or on a constraint, $Support^i_c(t)$ of the agents. The actions that the institution can implement depend on the field of application. As such, we present in Section 4.2 the actions of the institution for the case of the adoption of communicating water meters in agriculture.

```
while time < End_time do
  for all Individual agents i do
    if random(0.0, 1.0) ≤ P^i_interact then
      i' ← one_of(Individual agent in social_network^i)
      topic ← one_of(topics)
      if |Op^i_topic − Op^i'_topic| < threshold d^i then
        convergence of opinion on topic (Equations 9,
        10)
        updating of intention for i and i' (Equation
        2)
        updating of the adoption status for i and i'
        (Equation 1)
      end if
      if time modulo (6 months) = 0 then
        actions of the Institution agent
      end if
      if time modulo (1 year) = 0 then
        updating of the Institution agent's Budget
      end if
    end if
  end for
  time ← time + 1 day
end while
```

Algorithm 1. Pseudocode of the agent-based model of innovation diffusion.

Finally, during the updating of the institution's budget time frame (line 14–16), the agent *Institution* receives a new budget $Budget_{year}$ which is added to its current budget:

$$Budget(t) \leftarrow Budget(t) + Budget_{year}.$$

TABLE 2 State variables of the *Institution* agent at time step t.

State variable	Data type	Description
$Budget(t)$	Float-dynamic	Budget available to implement public policies
$Adopters(t)$	Float [0,1]-dynamic	Fraction of adopters
$Steps(t)$	Int-dynamic	Number of remaining decisions steps left to implement institutional actions (e.g., $T^{inst} - t$)

The computational complexity of the process described above for our agent-based model is linear in the number of agents.

3.2. Design concepts

3.2.1. Basic principles

The model is based on the use of the theory of planned behavior [3] to describe the adoption process of agents.

3.2.2. Interaction

Individual agents can interact directly with each other to try to influence each other. Also, *Individual* agents will interact indirectly through the social norm: the choice to adopt or not depends on the number of adopters in their social network.

The *Institution* agent will implement actions that will impact the *Individual* agents and in particular the opinion they have on the innovation with regard to different topics (at stake in the attitude computation), and also lift constraints to adoption (at stake in the PBC computation).

Also, there is an indirect interaction between the *Individual* and the *Institution* agents: the number of adopters can have an impact on the institution's decisions in terms of which action to implement.

3.2.3. Stochasticity

Apart from the initialization of the model which may involve stochasticity, an important element of stochasticity concerns the interactions between *Individual* agents: an *Individual* agent has a certain probability to choose to interact with another agent. The choice of the *Individual* agent with whom it interacts and the topic on which it wishes to discuss are chosen randomly.

Also, the choice of actions implemented by the *Institution* agent and the scope of these actions may involve stochasticity.

3.3. Details

3.3.1. Initialization

The initialization of the model includes the generation of the population of *Individual* agents: it is in particular a question of giving them values for all their attributes (linked to TPB, to the topics, to their social network...). Similarly, the *Institution* agent must be created and its attributes initialized. Section 4 gives an example of initialization of the model.

3.3.2. Submodels

3.3.2.1. Computation of the Intention

The intention of an *Individual* agent i is computed as follow:

$$I^i(t) \leftarrow W^i_{attitude} \times A^i(t) + W^i_{social} \times SN^i(t) + W^i_{pbc} \times PBC^i(t) \quad (2)$$

With:

$$W^i_{attitude} + W^i_{social} + W^i_{pbc} = 1 \quad (3)$$

The attitude A^i of an agent i considering the policy of the *Institution* agent is computed as follows:

$$A^i \leftarrow \sum_{k \in K} Min[1.0, Op^i_k(t) + Support^i_k(t)] \times W^i_k \quad (4)$$

With, K the set of adoption topics considered, and:

$$\sum_{k \in K} W^i_k = 1 \quad (5)$$

The social norm SN^i of an agent i is computed as follows:

$$SN^i(t) \leftarrow \frac{m^i(t)}{n^i} \quad (6)$$

where, $m^i(t)$ is the number of *Individual* agents of the social network $social\_network^i$ who have adopted at year t ($|\{i' \in social\_network^i, Adoption^{i'}(t) = true\}|$), and n^i is the number of *Individual* agents in $social\_network^i$.

The attitude $PBC^i(t)$ of an agent i at year t is computed as follows:

$$PBC^i(t) \leftarrow \sum_{c \in C} Min[1.0, Skill^i_c(t) + Support^i_c(t)] \times W^i_c \quad (7)$$

With, C the set of adoption constraints considered, and:

$$\sum_{c \in C} W^i_c = 1 \quad (8)$$

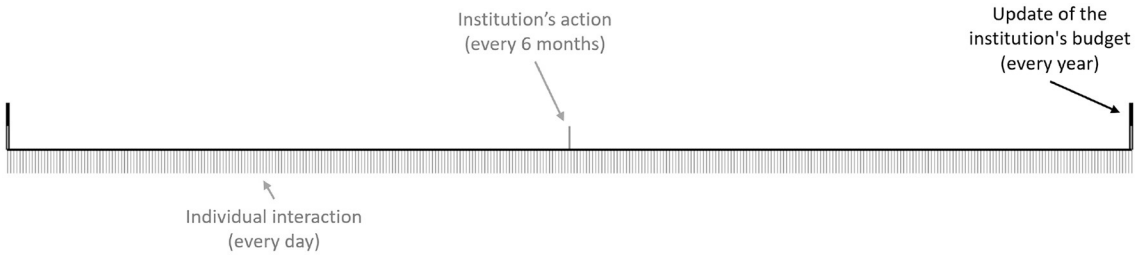

FIGURE 1
Time frames considered: every day, interaction of individuals; every 6 month's, implementation of the actions of the institution; every year, updating of the institution's budget.

3.3.2.2. Interaction between Individual agents

To manage interactions between *Individual* agents, we use the classic bounded confidence model [10]: when two *Individual* agents i and i' meet and begin a discussion on a topic k, they adjust their opinions on k as long as their difference of opinion on k is below a given threshold d^i. More formally, the opinions on k of i, i' are modified at time $t+1$ if $|Op_k^i(t) - Op_k^{i'}(t)| < d^i$ as follows:

$$Op_k^i(t+1) \leftarrow Op_k^i(t) + \mu \times \left(Op_k^{i'}(t) - Op_k^i(t)\right) \quad (9)$$

$$Op_k^{i'}(t+1) \leftarrow Op_k^{i'}(t) + \mu \times \left(Op_k^i(t) - Op_k^{i'}(t)\right) \quad (10)$$

4. Application for adoption of communicating water meters by farmers

The model presented in the previous section is intended to be completely generic and adaptable to any type of innovation[2]. In this section we instantiate this generic model for the particular application of the adoption of communicating water meters by farmers in South-West France.

4.1. Context

During the summer period, problems of water availability are common in some regions of France, particularly in the southern part of the country. These periods of drought have a direct impact on the daily life of irrigating farmers who must closely monitor their water consumption.

In a context of better resource management, various studies have highlighted the advantages that communicating meters could represent. Thus, on the Louts river (South-West of France), the Compagnie d'Aménagement des Coteaux de Gascogne (CACG), which is in charge of water distribution in this area, is proposing to irrigating farmers new communicating water meters to replace the aging mechanical meters. This is encouraged by the Ministry of the Environment, which requires the metering system to be replaced every 9 years.

Communicating water meters offer advantages over mechanical meters: they are more accurate and, above all, they allow for remote reading of consumption in real time. However, despite these advantages, CACG is having difficulty convincing farmers to install this device because, in general, they have a negative perception of it [29].

The questions then arise as to whether communicating meters will be adopted by farmers, what the impacts of the different information circulating on these devices are, and whether it is possible to implement public policies to promote a virtuous impact of these technologies. Answering these questions requires studying the social dynamics that lead to the adoption (or not) of an innovation within a population.

2 The model in Bourceret et al. [14] can be seen as a particular application of our model with low-input agricultural practices as innovation and with the following parameters: no interaction between individuals ($P^i_{interact}$ =0), 2 topics (Economy and Environment) and one constraint related to farmers' skills.

4.2. Specification of the model and description of the institution's actions

We have initialized the generic model presented in Section 3 for the case of the adoption of communicating water meters. In this model, *Individual* agents represent farmers. Based on the work of Sadou et al. [30] who have analyzed the arguments used by the stakeholders regarding communicating water meters, we identified 3 major topics: economy, environment and farm management (ease of management).

Concerning the adoption constraint, we chose to use the same type as Bourceret et al. [14], i.e., the technical skill.

We have integrated 3 types of actions for the institution: training, financial aid and environmental awareness. The actions are defined as follows:

- *Training*: For $N_{train}(t)$ *Individual* agents chosen randomly at time t, increasing permanently their opinion on the "Farm management" topic ($Op^i_{management}$) and their skill regarding the constraint "Technical" ($Skill^i_{technical}$) of a value $\theta_{train}(t)$:

$$Op^i_{management}(t+1) \leftarrow \min(1.0, Op^i_{management}(t) + \theta_{train}(t)) \quad (11)$$

$$Skill^i_{technical}(t+1) \leftarrow Skill^i_{technical}(t) + \theta_{train}(t) \quad (12)$$

  The cost of this action is: $N_{train}(t) \times \theta_{train}(t)$
- *Financial support*: For all agents, increases temporary the opinion on the "Economy" topic ($Support^i_{economy}$) of a value $\theta_{aid}(t)$:

$$Support^i_{economy}(t+1) \leftarrow \min(1.0, Support^i_{economy}(t) + \theta_{aid}(t)) \quad (13)$$

  The cost of this action is: $N_{new\_adopters}(t) \times \theta_{aid}(t)$, with $N_{new\_adopters}(t)$ the number of *Individual* agents who have adopted during the policy period t.
- *Environmental awareness*: For $N_{env}(t)$ *Individual* agents chosen randomly, increases permanently the opinion on the "Environment" topic ($Op^i_{environment}$) of a value $\theta_{env}(t)$:

$$Op^i_{environment}(t+1) \leftarrow \min(1.0, Op^i_{environment}(t) + \theta_{env}(t)) \quad (14)$$

  The cost of this action is: $\frac{1}{2} \times N_{env}(t) \times \theta_{env}(t)$. We defined a lower cost for this action because unlike the *training* action which concerns both *Attitude* and *Perceived behavior control*, this one concerns only *Attitude*. Moreover, unlike the *financial support* action which only incurs a cost on the budget if an agent adopts the innovation, here the cost of the action is spent as soon as the action is triggered even if it has no effect on the adoptions.

Table 3 summarizes all the argument values linked to the actions.

At the time of applying an action, the cost of the action is subtracted from the available budget and the action only applied if there is enough budget left. In more detail, for training and

TABLE 3 Parameters that determine the action of the *Institution* agent at time step t.

Action variable	Data type	Description
$N_{train}(t)$	Int [0, 100]	Number of *Individual* agents who will be trained
$\theta_{train}(t)$	Float [0.0, 1.0]	Level of training
$\theta_{aid}(t)$	Float [0.0, 1.0]	Level of financial support
$N_{env}(t)$	Int [0, 100]	Number of *Individual* agents who will be environmentally educated
$\theta_{env}(t)$	Float [0.0, 1.0]	Level of environmental sensibilization

environmental awareness actions (in this order), the simulation: (i) verifies whether there is enough budget left to apply the action (otherwise the action is not applied at all); and (ii) if it is the case, it applies the action, subtracts its cost from the available budget and continues to process the next type of action if any. Finally, for financial support actions, the aid is offered to new adopters (at the financial level decided by the institution for that period) and its cost subtracted as soon as new individuals adopt until no remaining budget is left or until the period ends.

The model is initialized with the creation of N_{ind} *Individual* agents.

The initialization of an *Individual* agent works as follows: its social network ($social\_network^i$) is filled by N_{social} *Individual* agents chosen at random. Then for each topic k (among Economy, Environment and farm management), the initial value of opinion $Op_k^i(1)$ and weight W_k^i for this topic are chosen randomly between 0.0 and 1.0 (uniform distribution). The weights of the different topics are then normalized so that the sum of the weights is equal to 1.0. The value of TPB weights ($W^i_{attitude}$, W^i_{social} and W^i_{pbc}) are also chosen randomly between 0.0 and 1.0 (uniform distribution) and then normalized. The adoption threshold (Ω^i) is initialized between 0.0 and 1.0 using a truncated Gaussian distribution with a mean of Ω_{mean} and a standard deviation of Ω_{std}. The speed of opinion convergence (μ), the maximal opinion difference accepted for convergence (d) and the probability of interaction ($P_{interact}$) are considered homogeneous for all agents.

Once all these parameters are initialized, the initial value of the Intention is computed using Equation 2.

All (non random) parameters' values as instantiated in experiments are given in Table 4.

TABLE 4 Parameter values used for the simulations.

State variable	Initial value
N_{ind}	100
N_{social}	5
$P_{interact}$	0.1
Ω_{mean}	0.7
Ω_{std}	0.1
μ	0.1
d	0.5
$Budget_{year}$	10.0
End_{time}	5 years

5. Policy design using reinforcement learning

This section details our approach that builds on machine learning, and in particular on reinforcement learning, to automatically learn effective policies for innovation diffusion. As initially discussed in the introduction, designing an effective policy for innovation is a very challenging problem. At each decision step, the institution needs to decide if it launches new (parallel) actions on several areas of interest (i.e., training, financial aid and environmental awareness in the particular case study) as well as the extent of each of these actions (i.e., level of increment on the opinion/skill and, for training and environmental awareness actions, the number of individuals reached by the action). Therefore, it is not only which actions to launch but when, in which combination and which parameters' values to choose. This task gets even more complex considering that, to do achieve it, the institution cannot observe any characteristic of the internal state of any individual (e.g., intention, preferences, ...) but only some aggregated statistics at the level of the population (e.g., the number of adopters). The institution can neither target the policy actions to reach specific individuals, but only choose the number of individuals reached.

The remainder of this section describes in detail the approach used: we mathematically formalize the policy design problem faced by the institution as a reinforcement learning problem (Section 5.1) and we describe how to solve it by a deep learning approach (Section 5.2).

5.1. The innovation policy design problem

In what follows we formally define the optimization problem faced by the institution when designing a policy that aims to maximize the number of adopters of an innovation over a finite time horizon, T^{inst} (i.e., the policy will be evaluated for the state reached after T^{inst} institution decision steps). In this problem, the policy actions launched in a given time step t are restricted by the available budget on that time step ($Budget(t)$). We cast this problem in the RL framework in which the institution learns efficient innovation diffusion policies by directly applying action policies on the (simulated) environment. Figure 2 depicts the main steps of interaction occurring between the institution and the environment within a RL iteration.

At every institution decision step $t \in T^{inst}$, the institution receives three observations from the environment (① in Figure 2): the fraction of adopters, the available budget and the number of decision steps remaining before time limit. These observations form the state of the environment perceived by the institution on that time step, i.e., $S(t) =< Budget(t), Adopters(t), Steps(t) >$. Notice that the only indicator that the institution can observe

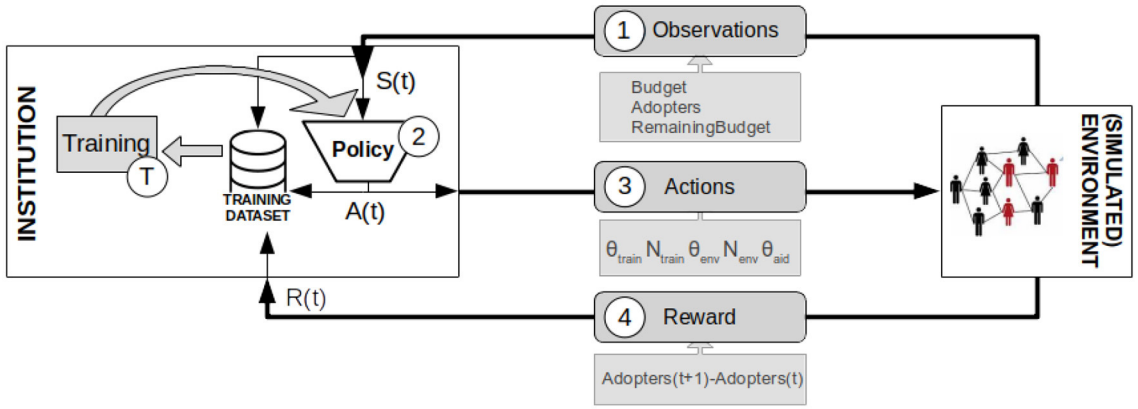

FIGURE 2
Schematic overview of the interactions between the institution (the learner) and the environment during the different RL steps.

from the population is the fraction of adopters. The decision steps remaining before time limit are included to provide time-awareness in the state representation, an approach that has proven good results in time-limited domains [31].

Given the observed state of the environment, the institution applies its policy (② in Figure 2) to decide which actions, $A(t)$, to launch in that state. For this, in this work we consider that the institution needs to define the value of a set of (continuous) parameters for each possible type of action and that multiple types of actions can be launched in parallel[3]. As detailed in Section 4.2, in the particular application studied in this paper, i.e., the adoption of communicating water meters by farmers, there are three types of policy actions: training, financial aid and environmental awareness. In more detail, the institution acts by specifying: (i) the level of each type of action (i.e., the level of training, θ_{train}, the level of the financial aid, θ_{aid} and the level of environmental sensibilisation, θ_{env}); and (ii) the number of individuals to be reached by the action for training (N_{train}) and environmental awareness (N_{env}) actions[4]. Therefore, for this application, the *policy* of the institution will be a stochastic mapping of the following form:

$$\begin{pmatrix} Budget \\ Adopters \\ Steps \end{pmatrix} \rightarrow \begin{pmatrix} \theta_{train} \\ N_{train} \\ \theta_{env} \\ N_{env} \\ \theta_{aid} \end{pmatrix}$$

This stochastic mapping builds on a policy function $\pi(S(t); \Theta)$ which in turn is a parametrized mapping (we use the notation Θ to denote the set of policy parameters) from environmental states to probability distributions over actions. As standard-practice in continuous action spaces, we based our approach on policy-gradient methods in which the institution will learn directly a parameterized policy that can select actions without consulting a value function[5].

Given a policy function, the actions to apply at a given time step t are obtained by directly sampling π on the current state:

$$A(t) \sim \pi(S(t); \Theta) \tag{15}$$

These actions, sent by the institution (③ in Figure 2), are applied to the environment at time step t constrained to the available budget (i.e., an action is only applied if there is enough budget left). In this model we consider that each type of action incurs a cost on the budget which depends on its particular parametrisation and that the institution is aware of the cost of its actions. In Section 4.2, we detailed the costs of each type of action for the particular application studied in this paper.

At time step $t+1$ the environment sends the reward signal[6] observed to the institution (④ in Figure 2). In RL, the learner's sole objective is to maximize the total reward received in the long run. Thus, given that here we are trying to maximize the number of adopters, the reward at a time step t is naturally defined as the increment on the fraction of adopters with respect to the previous step:

$$R(t) = Adopters(t) - Adopters(t-1) \tag{16}$$

Then, for learning purposes, the *Institution* agent stores in its training dataset the information related to this experience, i.e., $< S(t), A(t), R(t+1) >$.

The objective is to find a parametrization of the policy function that maximizes the discounted sum of rewards over time:

$$\max_{\Theta} E\left[\sum_{t=0}^{T^{inst}} \gamma^t \cdot R(t+1) \mid \Theta\right] \tag{17}$$

3 This differs from similar problems studied in the literature where at each time step the agent can only select one action whose parameters are typically decided in a second decision phase.

4 As discussed in Section 4.2, for financial aid the number of individuals reached is not decided ex-ante but determined ex-post since the aid is given iff an individual adopts.

5 A value function may still be used to learn the policy parameters, but is not required for action selection.

6 Reward at time t, $R(t)$ is typically a (stochastic) function of S_{t-1} and A_{t-1}.

where $\gamma \in [0, 1]$ is the discount factor.

At each training step[7] (Ⓣ in Figure 2), the *Institution* agent will input its training dataset to the training module which will output a new set of parameters (i.e., which will be used from this moment to parameterize the current policy in place).

In the next section we describe in detail the design of this training module as well as the particular parametric distributions we propose for the innovation diffusion policy design.

5.2. A deep learning approach to optimize public innovation diffusion policies

This section details the learning approach proposed to optimize the public innovation diffusion policies following the objective detailed in Equation (17). As common in the recent literature when dealing with highly complex partially-observable environments, we based our approach on deep reinforcement learning, where deep stands for an artificial deep neural network (NN) that is used to approximate the policy function. In deep RL, the set of parameters Θ that are adjusted by the learning are the weights of the NN. Also, as typically done in policy parametrization for continuous actions, the policy function is defined as a parametric probability distribution over actions[8] and the outputs of the NN are used to update the parameters of this distribution.

There are several challenges that we need to overcome when designing a neural network policy for this problem. These challenges emerge from the budget constraint that bounds the space of feasible actions depending on the state (i.e., the budget available). The typical approach of dealing with continuous action spaces in deep RL consists in using a normal (Gaussian) policy parametrization in which the NN outputs the mean and the standard deviation of the corresponding normal distribution. If the action is composed of multiple sub-actions, one normal distribution is used to specify each sub-action and the global action distribution is defined as the aggregation of individual distributions. However, notice that if we apply this approach to our problem there is no guarantee that the sampled solution will respect the budget. Still, this does not prevent us from sending these (possibly unfeasible) actions to the environment. As explained in Section 4.2, in this case the actions whose cost exceeds the available budget will not be applied, having the same effect as not launching these actions. However, as highlighted in the literature [20], acting as if there was no constraint in environments with large action spaces typically leads to inefficient learning and poor convergence rate if any. Given this, in this work we opted for an approach which explicitly considers the constraint in the policy, guaranteeing that the actions selected by the policy respect the budget.

Creating a policy that respects the budget constraint is particularly challenging in the innovation policy design problem. In more detail, many constrained problems deal with constraints that apply to individual actions (i.e., define independent bounds on the domain of each sub-action) and, in the continuous action domain, they can be addressed by independently bounding the corresponding distribution of each sub-action [21]. Instead, here the budget is shared among all sub-actions, which in turn can be launched in parallel, so the bounds on an action not only depend on the state but also on the values of other sub-actions. Thus, we can not guarantee that we will respect the constraint by bounding each action individually depending on the state.

Next section describes the particular NN architecture that we propose to overcome the above-mentioned open challenges of using deep RL for innovation diffusion policy design.

5.2.1. A NN architecture to optimize public policies

In the proposed architecture, a classical Gaussian policy parametrisation approach will be used for the probability distributions that define the level with which we apply each type of action at individual level (i.e., θ_{env}, θ_{train} and θ_{aid}). But for the other parameters that characterize the actions, i.e., the number of individuals reached by training and environmental awareness actions, we take a different approach (i.e., those actions will not be directly sampled from probability distributions) that guarantees that the actions selected by the policy respect the budget constraint. In more detail, the NN will output a second set of parameters that define a probability distribution on the allocation of the available budget among the different types of actions. Then, the number of individuals reached by training and environmental awareness actions are unequivocally determined after the realizations of the budget allocation and the level of these actions. Figure 3 depicts this NN architecture along with the different steps of the process that goes from the NN output to the institution policy actions.

5.2.1.1. Normal (Gaussian) distributions for each real-valued θ action

As shown in Figure 3, the NN will output a first set of parameters composed of a mean and a standard deviation for each type of action (e.g., θ_a^{mean}, θ_a^{std} for each $a \in \{train, env, aid\}$). These parameters will be used to define a tanh-squashed normal (Gaussian) distributions over the real-valued action levels. Formally,

$$P(\theta_a|s) = T(\mathcal{N}(\theta_a^{mean}, \theta_a^{std})) \quad \forall a \in \{train, env, aid\} \tag{18}$$

Where T is a tanh-squashed transformation to bound the range of actions ($R \rightarrow [0, 1]$) and $\mathcal{N}$ is the gaussian distribution.

The values of the extent levels of actions are obtained by directly sampling the corresponding distributions on the current state:

$$\theta_{train}(t) \sim P(\theta_{train}|s),\ \theta_{env}(t) \sim P(\theta_{env}|s),\ \theta_{aid}(t) \sim P(\theta_{aid}|s) \tag{19}$$

5.2.1.2. Dirichlet distribution for budget allocation

For modeling the budget allocation among the different types of actions, we propose to use a Dirichlet distribution, a type of distribution typically used in the allocation of (continuous) resources [32] given that the realizations of the distribution satisfy a simplex constraint (i.e., $\sum \mathbf{x} = 1, \mathbf{x} \sim Dir(\cdot)$). Thus, as depicted in Figure 3, the NN will output the concentration parameters of a three-dimensional Dirichlet distribution: one for the training

7 An agent is trained by batches of experiences, the number of experiences per batch being determined by the hyperparameters of the particular learning algorithm used.

8 This parametric distribution over actions can be defined as the aggregation of several parametric distributions on different actions as we do in this work.

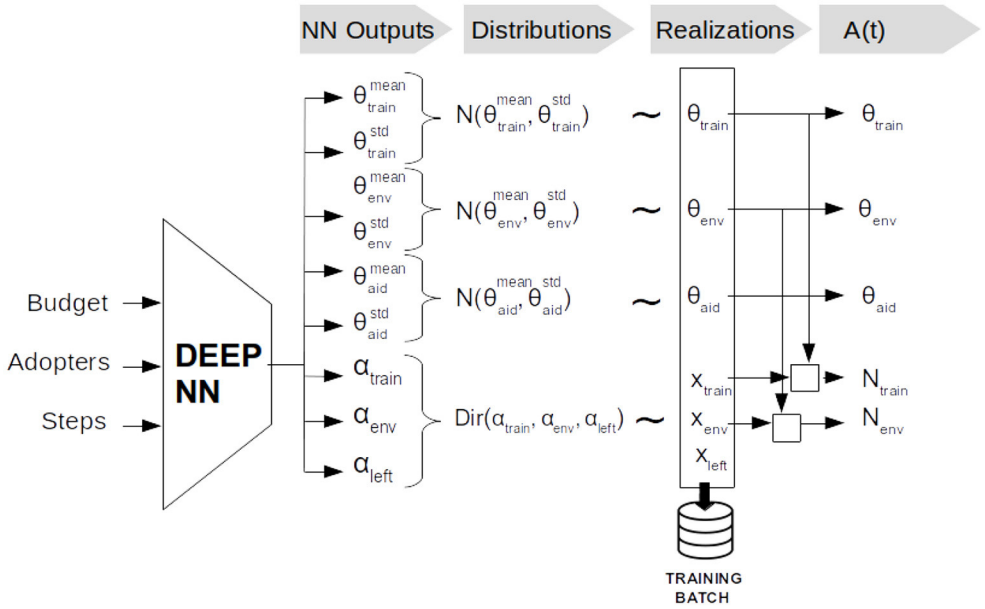

FIGURE 3
Schematic overview of the process from the NN input to the institution policy actions. (1) The NN gets as input the current state of the environment. (2) The outputs of the NN are used to define some parametric distributions over actions in the current state. (3) The distributions are sampled to obtain realizations for the current iteration. (4) Policy actions are defined from the realizations.

dimension (α_{train}), one for the environmental awareness dimension (α_{env}) and one for the budget left for the financial aid dimension and/or to be transferred to the next time step (α_{left}) [9]. Formally,

$$P(x_{train}, x_{env}, x_{left}|s) = Dir(\alpha_{train}, \alpha_{env}, \alpha_{left}) \quad (20)$$

A realization $(x_{train}(t), x_{env}(t), x_{left}(t))$ of the Dirichlet distribution represents a partition of the current budget $Budget(t)$: $(x_{train}(t) \times Budget(t), x_{env}(t) \times Budget(t), x_{left}(t) \times Budget(t))$.

5.2.1.3. Determining N_{env} and N_{train} actions given the budget and θ-actions realizations

Finally, given the current budget partition and the level of extent of each action, the number of individuals reached by training and environmental awareness actions are determined as the maximum number of individuals to which we can apply the selected level for that action while respecting the allocated budget. Formally, $\forall a \in \{train, env\}$

$$N_a(t) = \arg \max_{N \in \{0,\ldots,N_{ind}\}} \left[N \times C_a(\theta_a(t)) \leq x_a(t) \times Budget(t)\right] \quad (21)$$

Where $C_a(\theta_a(t))$ is a function which, given the level of extent to which the action will be applied in the period, returns the cost of applying that action to a single individual. The costs of the actions are specified in Section 4.2, $C_{train}(\theta_{train}(t)) = \theta_{train}(t)$ and $C_{env}(\theta_{env}(t)) = \frac{1}{2} \times \theta_{env}(t)$ for the particular case of study.

Therefore, the Deep NN policy allocates a probability distribution over joint actions to every possible state of the system (year, budget, adopters). This distribution can be sampled whenever we require an action to apply to the system.

6. Results

This section presents the experiments performed to validate the model (Section 6.1) and the framework proposed for policy design using reinforcement learning (Section 6.2) on the adoption of communication water meters in the farmers use case.

The model has been implemented in the open-source platform GAMA [33].[10] The choice of this platform is due to the ease of implementation of models with it, but also, in a perspective of evolution of the model. Indeed, we plan to enrich the model with geographical data to represent real farms and thus be able to calculate in a more advanced way the economic context of the farm and what the communicating water meters can bring them. We also plan to enrich the model by taking up work of Sadou et al. [11] on argumentation to allow a more detailed calculation of the attitude from the knowledge and point of view of the farmer. And GAMA offers integrated tools to support both extensions, namely to integrate geographical data and to explicitly represent arguments in the model [34]. Finally, GAMA also supports communication with external software using message exchange which has facilitated the coupling with the learning module (implemented in Python to take advantage of the existing deep learning libraries). We perform experiments using our implementation of the RL logic and the Proximal Policy Optimization (PPO) algorithm [35] but building on the *Tensorflow* framework[11] for the deep learning part. For the sake of reproducibility, we have made publicly available[12] the source code

9 Note that how the budget left is distributed between the financial aid dimension and budget not spent at time t and thus transferred to time $t+1$ is not controlled by the institution policy.

10 https://gama-platform.org/

11 https://www.tensorflow.org/

12 https://github.com/ptaillandier/policy-design

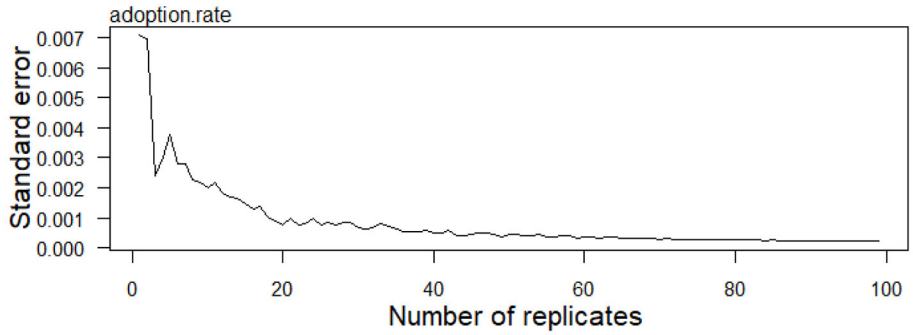

FIGURE 4
Comparison of the standard errors of the adoption rate for the different numbers of replicates.

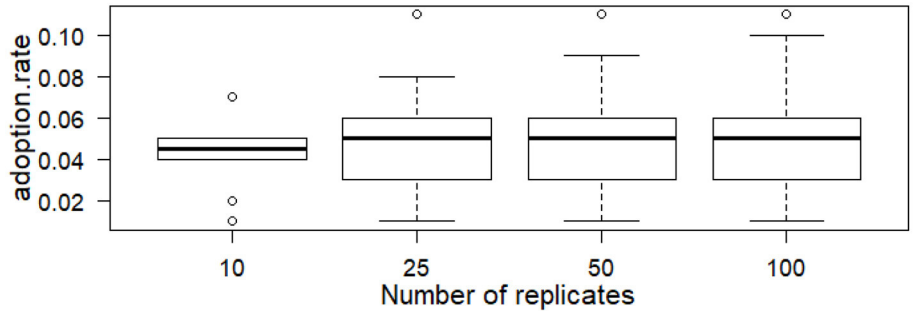

FIGURE 5
Whiskers plots of the adoption rate for the different numbers of replicates. The black lines represent the median values; the boxes represent the interquartile range (IQR), the whiskers represent the minimum/maximum excluding 1.5 IQR outliers. Points are outliers beyond that distance.

of the agent-based model and the reinforcement learning methods used in the experiments.

6.1. Analysis of the model

6.1.1. Stochasticity sensitivity analysis

In a first experiment, we analyze the impact of the stochasticity of the simulations on the results and in particular on the number of adopters. The main objective is to find a threshold value of replications beyond which an increase in the number of replications would not imply a significant marginal decrease of the difference between the results. To do this, we compare the number of adopters at the end of the simulation (i.e., after 5 years) between different numbers of replications of the simulation. We undertake this exploration with the simplest possible scenario, i.e., without any action implemented by the *Institution* agent.

Figure 4 shows the standard error of the adopters rate obtained with different numbers of replicates. Figure 5 shows the impact of the number of replicates on adoption rate: the black lines show the median, the boxes show the second and third quartiles (IQR), the whiskers show the minimum and maximum excluding outliers (simulation results that differ from the median by more than 1.5 times the IQR).

The results suggest that increasing the number of replications beyond 50 does not have a great impact on the aggregate trend of the simulations.

6.1.2. Analysis of the impact of the institution's actions

We propose here to study the impact of the different possible institution's actions individually. In more detail, we calculate, for the three possible types of action defined (financial support, environmental sensibilisation and training), the average values of adoption obtained for 50 replications, depending on the level of these actions. We consider in these experiments that the same action is applied at each institution's action implementation stage (every 6 months) - the action is thus applied 10 times during the simulation period (5 years).

Table 5 presents the results in terms of adoption rate and cost for different levels of financial support. Tables 6, 7 show, respectively the adoption rate and the cost of the training action according to the level of training and proportion of farmers concerned. Finally, Tables 8, 9 show, respectively the adoption rate and the cost of the environmental sensibilisation action according to the level of sensibilisation and proportion of farmers concerned.

A first result is that it is the *training* action that is the most effective in bringing new adopters. This result is not surprising since this action allows the farmer to better understand the interest

TABLE 5 Adoption rate by level of financial support: mean value for the 50 replications and standard deviation in brackets.

θ_{aid}	Adoption rate	Cost
0.0	0.045 (0.02)	4.5 (2)
0.2	0.059 (0.025)	5.9 (2.5)
0.4	0.078 (0.034)	7.8 (3.4)
0.6	0.116 (0.046)	11.6 (4.6)
0.8	0.175 (0.066)	17.5 (6.6)
1.0	0.242 (0.078)	24.2 (7.8)

TABLE 6 Adoption rate by level of training support (θ_{train}) and number of farmers concerned (N_{train}): mean value for the 50 replications and standard deviation in brackets.

N_{train} \ θ_{train}	0.0	0.2	0.4	0.6	0.8	1.0
0	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)
20	0.045 (0.026)	0.184 (0.067)	0.521 (0.198)	0.764 (0.194)	0.801 (0.171)	0.805 (0.179)
40	0.045 (0.026)	0.373 (0.134)	0.736 (0.187)	0.805 (0.176)	0.824 (0.167)	0.85 (0.154)
60	0.045 (0.026)	0.491 (0.181)	0.778 (0.182)	0.809 (0.174)	0.842 (0.162)	0.872 (0.14)
80	0.045 (0.026)	0.558 (0.186)	0.792 (0.179)	0.825 (0.172)	0.85 (0.156)	0.886 (0.133)
100	0.045 (0.026)	0.574 (0.186)	0.804 (0.169)	0.831 (0.163)	0.866 (0.142)	0.893 (0.13)

of the innovation for managing the farm (and thus, increase her attitude toward adoption) and at the same time to remove the technical obstacles to adoption (increase the perceived behavior control). However, this action is the most expensive, as in this case of application we consider a budget of 10 per year and that for 5 years, the action allows at best to bring the adoption percentage to 18.4% (training of 20% of farmers to improve their level of technicality of 0.2). At the same time, the *financial support* action allows for an average adoption percentage of around 24.2% for a budget of 24.2.

Indeed, the advantage of the *financial support* action is that it is only spent when one farmer adopts the innovation, which greatly limits the cost in the case where few farmers adopt the innovation. Note that this cost could have been much higher if more agents had adopted the innovation.

The *environmental sensibilization* action had a much smaller effect on results with at best a 4% increase in the adopter percentage for a budget of 50.

6.2. Experiment on policy design

In this experiment the *Institution* agent uses (deep) RL to optimize its policy, following the approach and architecture detailed in Section 5. The *Institution* agent uses a two-layer (64 neurons each) feed-forward neural network for the policy approximation. The policy is trained using the Proximal Policy Optimization (PPO) algorithm [35] with the agent learning by batches, each batch containing experiences of 100 complete[13] episodes[14]. Since each complete episode is composed of 10

TABLE 7 Cost by level of training support (θ_{train}) and number of farmers concerned (N_{train}).

N_{train} \ θ_{train}	0.0	0.2	0.4	0.6	0.8	1.0
0	0	0	0	0	0	0
20	0	40	80	120	160	200
40	0	80	160	240	320	400
60	0	120	240	360	480	600
80	0	160	320	480	640	800
100	0	200	400	600	800	1,000

TABLE 8 Adoption rate by level of environmental sensibilisation (θ_{env}) and number of farmers concerned (N_{env}): mean value for the 50 replications and standard deviation in brackets.

N_{env} \ θ_{env}	0.0	0.2	0.4	0.6	0.8	1.0
0	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)	0.045 (0.026)
20	0.045 (0.026)	0.062 (0.03)	0.074 (0.032)	0.076 (0.033)	0.077 (0.033)	0.077 (0.033)
40	0.045 (0.026)	0.071 (0.031)	0.078 (0.031)	0.079 (0.032)	0.081 (0.033)	0.084 (0.031)
60	0.045 (0.026)	0.072 (0.031)	0.078 (0.03)	0.079 (0.031)	0.081 (0.032)	0.084 (0.032)
80	0.045 (0.026)	0.073 (0.03)	0.079 (0.03)	0.082 (0.033)	0.084 (0.033)	0.087 (0.032)
100	0.045 (0.026)	0.073 (0.03)	0.08 (0.03)	0.083 (0.033)	0.085 (0.033)	0.088 (0.033)

TABLE 9 Cost by level of environmental sensibilisation (θ_{env}) and number of farmers concerned (N_{env}).

N_{env} \ θ_{env}	0.0	0.2	0.4	0.6	0.8	1.0
0	0	0	0	0	0	0
20	0	20	40	60	80	100
40	0	40	80	120	160	200
60	0	60	120	180	240	300
80	0	80	160	240	320	400
100	0	100	200	300	400	500

13 Since our problem is characterized by a fixed finite time horizon there is no need to truncate the episode before its end as happens in other domains.

14 Following RL standard notation, an episode tracks all experiences obtained during the whole time horizon.

TABLE 10 Training hyperparameters.

	Parameter	Value
RL	Discount factor (γ)	0.99
	Advantage function	GAE
	GAE-lambda	0.95
	Training algorithm	PPO
PPO	Sampling horizon	10 (until episode termination)
	Number of learning iterations	30
	Number of episodes per batch	100
	Number of training epochs per update	10
	Number of training minibatches per update/epoch	10
	Clipping ratio	0.2
	Early termination with KL divergence	No
	Mini-batch splitting	Shuffle transitions
	Recomputation of advantages at epoch level	yes
	Separated networks for policy and value	yes
	Optimizer	Adam
	Adam learning rate	0.0003
	Adam epsilon parameter	0.0000007
Policy NN	Activation function for hidden layers	Tanh
	Number of (hidden) fully-connected layers	2
	Fully connected layer dimension	64
	Last layer scaling	0.01
	Initializer (hidden layers)	Orthogonal with gain=1.41
Value NN	Activation function for hidden layers	Tanh
	Number of (hidden) fully-connected layers	2
	Fully connected layer dimension	64
	Last layer scaling	1.0
	Initializer (hidden layers)	Orthogonal with gain=1.41

The *Last layer scaling* parameter rescales the network weights of the last layer after initialization [36].

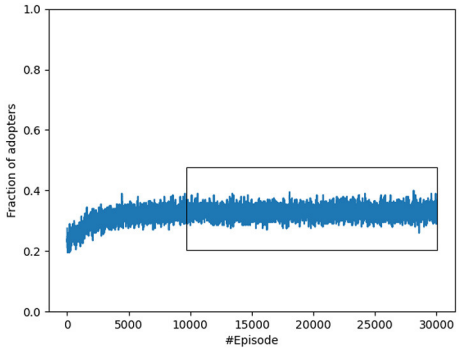

FIGURE 6
Fraction of adopters reached at the end of each simulation episode with the stable training period marked with a box.

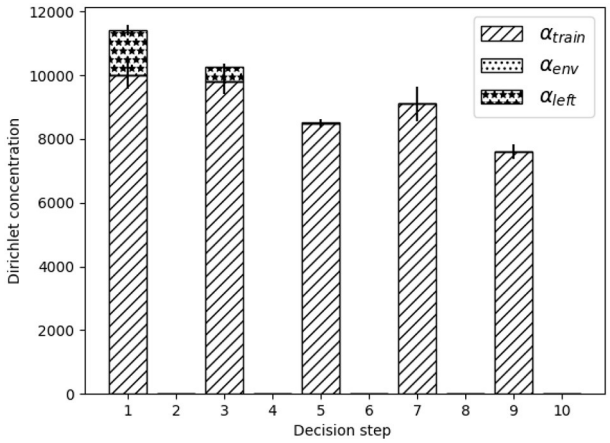

FIGURE 7
Mean values of the Dirichlet concentration parameters, output of the NN to define budget distribution, per institution decision step for the training stable period. Error bars stand for standard deviation.

experiences, this leads to 1000 experiences sampled from the environment in each training iteration using the last policy parameters. In every training iteration, we perform 10 epochs (i.e., the learner will perform 10 passes over the whole batch training dataset). In each epoch, the indices of experiences in the batch are randomly shuffled (i.e., shuffling transitions) and shuffled experiences are partitioned into 10 mini-batches (i.e., each minibatch containing 100 experiences). Table 10 details the training hyperparameters used in our experiments. The hyperparameters were defined following the recommendations issued after the large experimental study carried out for online deep RL methods in Andrychowicz et al. [36]. Regarding the execution time, as it is frequent in deep reinforcement learning the execution time is dominated by the generation of experiences on the simulation module (the simulation of one complete episode[15] takes around 10 s[16]). The time spent to actual training is negligible in comparison (one training iteration takes less than 1 s[16]).

Figure 6 shows the empirical training progress by plotting the adoption rate reached at the end of each episode, resulting from the application of the current policy learned by the institution. The first thing we observe is that the *Institution* agent is able to learn efficient policies (converging to policies with more than 33% of adopters) after 10 learning iterations (recall that with the used parameters the learning occurs every 1000 simulation episodes). For the whole training period the minimum fraction of adopters is 0.09, the maximum is 0.57 and the mean is 0.32 (with standard

15 One complete episode contains 10 decision steps for the institution but also the 365(days)*5(years) simulation steps for individual agents.

16 Experiments where run on an Intel(R) Xeon(R) W-2133 CPU @ 3.60 machine with 128GiB System memory and GPU GeForce RTX 2080.

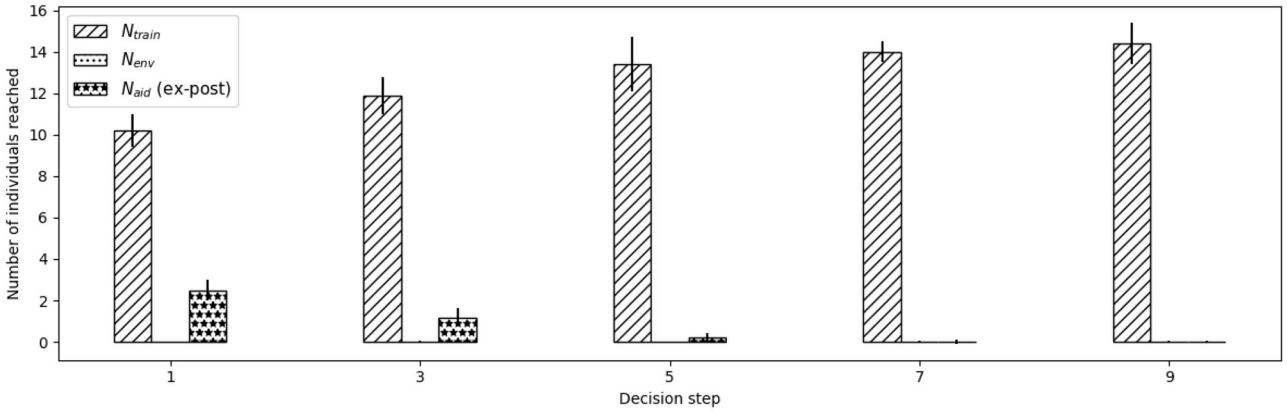

FIGURE 8
Mean values of the N_{train} and N_{env} parameters chosen by the institution and of the observed N_{aid} on odd decision steps during the training stable period. Error bars stand for standard deviation.

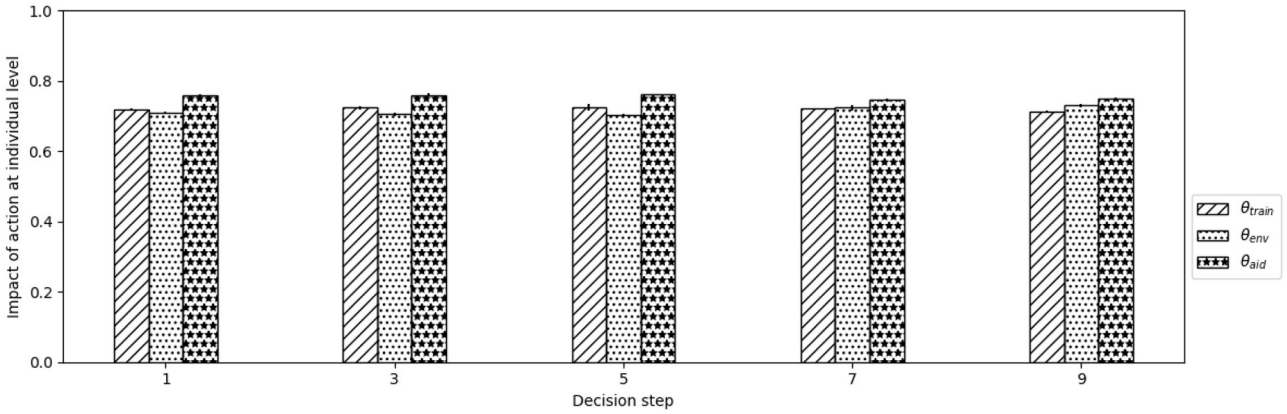

FIGURE 9
Mean values of the level of impact of each type of action (i.e., θ-parameters) chosen by the institution on odd decision steps during the training stable period. Error bars stand for standard deviation.

deviation of 0.0489). We consider that the learning is stabilized after 10,000 training episodes (this with no parallelisation, i.e., all simulations run sequentially, which requires 1–2 days of training[16]) and use the interval 10,000–30,000 to study the structure of the learned policies.

Figure 7 shows, for each institutional decision step, the average of the concentration parameters generated by the NN, i.e., used to define the Dirichlet probability distribution on the budget allocation among the different action types. The first thing we observe in this graph is that concentration values of odd decision steps (corresponding to the beginning of the year when the institution budget gets incremented by 10) are much larger than those of even steps. In fact, this is a consequence of the institution using the budget as soon as it becomes available (i.e., leaving no budget left for future time steps). This is very clear on steps 5, 8 and 9 for which the concentration parameter α_{left} is close to 0. Instead, steps 1 and 3 allocate some budget *via* α_{left} but as we will see later in Figure 8, this budget is not left for the next step but instead entirely spent on the financial support action (i.e., providing the financial aids to new adopters). As a result of this, budget allocation at even steps has no effect on the reward and the NN is unable to reduce the variance of the distribution leading to the observed smaller concentration values.

The second thing we observe in Figure 7 is that the institution learned to spend nearly all budget on the most effective type of action: the training action. As discussed in Section 6.1.2 when evaluating the baselines policies for each type of action, the effectiveness of this type of action is explained as it increases the attitude to adoption at the same time as it increases the perceived behavior control. Despite this dominance of the training dimension, in the first decision steps the institution finds profitable to reserve some budget for ex-post financial aids. Finally, no budget is allocated to environmental sensibilisation actions.

Figures 8, 9 show respectively for the N-parameters and the θ-parameters, the average values of the parameters chosen by the institution at each decision step to parameterize each type of action. For the sake of clarity, we only plot decision steps corresponding to the beginning of a year (i.e., odd time steps) since, as discussed

above, for the rest of the time steps there is no budget left to apply any action to any individual. The first thing we observe in Figure 8 is that, unsurprisingly and as a consequence of the budget allocated, the number of agents reached by environmental awareness actions (N_{env}) is 0 in all decision steps. For financial aids, there is some percentage of individuals that get financial aids on the first steps (i.e., around 2 individuals get the financial aid at step 1 and around a single individual at step 3). As we see in Figure 9 the level of aid (θ_{aid}) proposed to individuals is quite high (values around 0.75). Finally, for training actions, we observe that the level of training does not vary much across the decision steps and it is quite high (values around 0.72). In Figure 8, we observe that this results in around 10-11 individuals trained at each earlier time steps (in which some budget was allocated to financial aids) and in around 14 individuals trained at later time steps (in which the whole budget is allocated to that type of action).

7. Conclusions and future work

This paper proposes an AI framework for the design of innovation diffusion policies. The innovation diffusion policy design problem is a complex sequential decision-making task in which an institution needs to decide which policy actions to launch over time in order to maximize the number of adopters of an innovation after a finite time horizon. The actions available are constrained by the available budget at the decision time. The proposed framework builds on two distinguished components:

- Agent-based simulations, used as a virtual environment in order to conduct a large number of experiments that would be prohibitive on the real environment; and
- (Deep) reinforcement learning, used to automatically identify good-candidate policies in the extremely large search space of possible ones.

In our framework, the agent-based simulations make use of the theory of planned behavior to simulate the behavior of adopters in a credible way, while keeping the computational cost of simulations affordable. Then, a deep reinforcement learning agent interacts with these simulations in order to efficiently explore the exponential space of institutional policies, eventually learning the structure of efficient innovation diffusion policies. The learning represents the policy *via* a neural network architecture that guarantees the respect of the budget constraint by implicitly learning the budget allocation among the different types of actions.

The proposed framework is illustrated in the specific use case of the adoption of communicating water meters by farmers in the Louts region (South-West of France). Empirical results demonstrate the viability and soundness of our approach to identify good-candidate innovation adoption policies for this particular application.

We identify multiple directions that can be pursued as future work. The paper demonstrates for the first time that an AI framework that combines (deep) RL and agent-based simulations is sound and viable for the innovation policy design problem, learning effective policies in the presence of non mutually-exclusive parameterizable actions and budget constraints. However, these results are a first step since any real-support to policymaking will require simulations with ground in real-data as well as widespread consultation with policy-makers. So future research should calibrate the model with real-world data. For the specific use case of communicating water meters we are currently carrying out interviews with farmers in the South-West of France in order to better understand their opinion on these water meters and to be able to move from a random initialization of the agents' attributes to values based on real data. We also hope to get real data on the adoption of these new water meters so that we can also calibrate the model with this information. Second, regarding the NN internal architecture, as future work we plan to test Long Short-Term Memory (LSTM) networks, typically used [5, 6] for encoding the history of past observations in partially-observable environments, in order to analyze the performance impact, instead of the simpler feed forward NN architecture used in this paper. Finally, regarding the objective used in the policy optimization, in this work the budget of the policy maker is represented only as a constraint but not in the objective function. Future work would consider enhancing the reward function in order to minimize the budget as a secondary objective and analyzing the impact of this on the structure of optimal policies.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

The authors have worked together, each in their own specialty, to design and describe the work presented in the paper. MV and PT are the principal designers of this document and are responsible for much of the writing. The agent-based model was mainly developed by PT, LS, and KC. The reinforcement learning approach was developed by MV and RS. BL developed the methods allowing interactions between the simulator and the learning module. Finally, RT and SC, in addition to providing their expertise on the design of models and participated in the writing of the article. All authors contributed to the article and approved the submitted version.

Funding

This work had been funded by INRAE (MathNum department) and by the #Digitag convergence institute (ANR 16-CONV-0004).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Kiesling E, Günther M, Stummer C, Wakolbinger LM. Agent-based simulation of innovation diffusion: a review. *Central Eur J Operat Res*. (2012) 20:183–230. doi: 10.1007/s10100-011-0210-y

2. Bass FM. A new product growth for model consumer durables. *Manag Sci*. (1969) 15:215–27. doi: 10.1287/mnsc.15.5.215

3. Ajzen I. The theory of planned behavior. *Organ Behav Hum Decis Process*. (1991) 50:179–211. doi: 10.1016/0749-5978(91)90020-T

4. Bourceret A, Amblard L, Mathias JD. Governance in social-ecological agent-based models: a review. *Ecol Soc*. (2021) 26:238. doi: 10.5751/ES-12440-260238

5. Zheng S, Trott A, Srinivasa S, Naik N, Gruesbeck M, Parkes DC, et al. The AI economist: improving equality and productivity with AI-driven tax policies. *arXiv preprint* arXiv:200413332 (2020). doi: 10.48550/arXiv.2004.13332

6. Trott A, Srinivasa S, van der Wal D, Haneuse S, Zheng S. Building a foundation for data-driven, interpretable, and robust policy design using the ai economist. *arXiv preprint* arXiv:210802904. (2021) doi: 10.2139/ssrn.3900237

7. Danassis P, Filos-Ratsikas A, Faltings B. Achieving diverse objectives with AI-driven prices in deep reinforcement learning multi-agent markets. *arXiv preprint* arXiv:210606060 (2021). doi: 10.48550/arXiv.2106.06060

8. Liu Y, Halev A, Liu X. Policy learning with constraints in model-free reinforcement learning: a survey. In: Zhou Z, editor. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021 Virtual Event/Montreal, Canada, 19-27 August 2021*. Montreal, QC: ijcai.org (2021). p. 4508–15.

9. Deffuant G, Huet S, Amblard F. An individual-based model of innovation diffusion mixing social value and individual benefit. *Am J Sociol*. (2005) 110:1041–69. doi: 10.1086/430220

10. Deffuant G, Neau D, Amblard F, Weisbuch G. Mixing beliefs among interacting agents. *Adv Complex Syst*. (2000) 3:87–98. doi: 10.1142/S0219525900000078

11. Sadau L, Couture A, Thomopoulos R, Taillandier P. Better representing the diffusion of innovation through the theory of planned behavior and formal argumentation. In: *Advances in Social Simulation: Proceedings of the 16th Social Simulation Conference*. Springer (2022). p. 423–35.

12. Rogers EM. *Diffusion of Innovations*. 5th ed. New York, NY: Free Press (1962).

13. Zhang H, Vorobeychik Y. Empirically grounded agent-based models of innovation diffusion: A critical review. *Artif Intell Rev*. (2019) 52:707–41.

14. Bourceret A, Amblard L, Mathias JD. Adapting the governance of social-ecological systems to behavioural dynamics: an agent-based model for water quality management using the theory of planned behaviour. *Ecol Econ*. (2022) 194:107338. doi: 10.1016/j.ecolecon.2021.107338

15. Beedell J, Rehman T. Using social-psychology models to understand farmers' conservation behaviour. *J Rural Stud*. (2000) 16:117–27. doi: 10.1016/S0743-0167(99)00043-1

16. Masson W, Ranchod P, Konidaris GD. Reinforcement learning with parameterized actions. In: Schuurmans D, Wellman MP, editors. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016*. Phoenix, AZ:AAAI Press (2016). p. 1934–40.

17. Hausknecht MJ, Stone P. Deep reinforcement learning in parameterized action space. In: Bengio Y, LeCun Y, editors. *4th International Conference on Learning Representations, ICLR 2016. San Juan, Puerto Rico, May 2-4, 2016 Conference Track Proceedings*. San Juan (2016).

18. He J, Ostendorf M, He X, Chen J, Gao J, Li L, et al. Deep reinforcement learning with a combinatorial action space for predicting popular reddit threads. In: Su J, Carreras X, Duh K, editors. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016 Austin, Texas, USA, November 1-4, 2016*. Austin, TX: The Association for Computational Linguistics (2016). p. 1838–48.

19. Delarue A, Anderson R, Tjandraatmadja C. Reinforcement learning with combinatorial actions: an application to vehicle routing. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 NeurIPS 2020 December 6-12, 2020*. (2020). Available online at: https://proceedings.neurips.cc/paper/2020/hash/06a9d51e04213572ef0720dd27a84792-Abstract.html

20. Huang S, Ontañón S. A closer look at invalid action masking in policy gradient algorithms. In: Barták R, Keshtkar F, Franklin M, editors. *Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022 Hutchinson Island, Jensen Beach, Florida, USA, May 15-18, 2022*. Jensen Beach, FL (2022).

21. Chou P-W, Maturana D, Scherer SA. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In: Precup D, The YW, editors. *Proceedings of the 34th International Conference on Machine Learning, Vol.70*. Sydney, NSW: PMLR (2017). p. 834–43. Available online at: http://proceedings.mlr.press/v70/chou17a.html

22. Dalal G, Dvijotham K, Vecerík M, Hester T, Paduraru C, Tassa Y. Safe exploration in continuous action spaces. *CoRR*. (2018) abs/1801.08757. doi: 10.48550/arXiv.1801.08757

23. Bhatia A, Varakantham P, Kumar A. Resource constrained deep reinforcement learning. In: Benton J, Lipovetzky N, Onaindia E, Smith DE, Srivastava S, editors. *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2018*. Berkeley, CA: AAAI Press (2019). p. 610–20.

24. Chow Y, Nachum O, Duenez-Guzman EA, Ghavamzadeh M. A lyapunov-based approach to safe reinforcement learning. In: Bengio S, Wallach HM, Larochelle H, Grauman, Cesa-Bianchi N, Garnett R, editors. *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*. Montreal, QC (2018), p. 8103–12. Available online at: https://proceedings.neurips.cc/paper/2018/hash/4fe5149039b52765bde64beb9f674940-Abstract.html

25. Liu Y, Ding J, Liu X. IPO: Interior-point policy optimization under constraints. *Proc AAAI Conf Artif Intell*. (2020) 34:4940–7. Available online at: https://ojs.aaai.org/index.php/AAAI/article/view/5932

26. Yang T, Rosca J, Narasimhan K, Ramadge PJ. Projection-based constrained policy optimization. In: *8th International Conference on Learning Representations, ICLR 2020 Addis Ababa, Ethiopia, April 26-30, 2020*. Ababa: OpenReview.net (2020).

27. Grimm V, Railsback SF, Vincenot CE, Berger U, Gallagher C, DeAngelis DL, et al. The ODD protocol for describing agent-based and other simulation models: a second update to improve clarity, replication, and structural realism. *J Artif Soc Soc Simulat*. (2020) 23:7. doi: 10.18564/jasss.4259

28. Stoneman P, Diederen P. Technology diffusion and public policy. *Econ J*. (1994) 104:918–30. doi: 10.2307/2234987

29. Collard A-L, Garin P, Montginoul M. Un compteur guillemotleft intelligent guillemotright pour mesurer les usages de l'eau: l'entree en scene d'une nouvelle connaissance. *Developpement durable et territoires, Economie, geographie, politique, droit, sociologie*. (2019) 10.

30. Sadou L, Couture S, Thomopoulos R, Taillandier P. Simuler la diffusion d'une innovation agricole à l'aide de modèles à base d'agents et de l'argumentation formelle. *Revue Ouverte d'Intelligence Artificielle*. (2021) 2:65–93. doi: 10.5802/roia.10

31. Pardo F, Tavakoli A, Levdik V, Kormushev P. Time limits in reinforcement learning. In: *Proceedings of the 35th International Conference on Machine Learning, Vol. 80*. Stockholm: PMLR (2018). p. 4042–51. Available online at: http://proceedings.mlr.press/v80/pardo18a.html

32. Tian Y, Han M, Kulkarni C, Fink O. A prescriptive Dirichlet power allocation policy with deep reinforcement learning. *Reliabil Eng Syst Safety*. (2022) 224:108529. doi: 10.1016/j.ress.2022.108529

33. Taillandier P, Gaudou B, Grignard A, Huynh QN, Marilleau N, Caillou P, et al. Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica*. (2019) 23:299–322. doi: 10.1007/s10707-018-00339-6

34. Tailandier P, Salliou N, Thomopoulos R. Introducing the argumentation framework within agent-based models to better simulate agents' cognition in opinion dynamics: application to vegetarian diet diffusion. *J. Artif. Soc. Soc. Simul*. (2021) 24:1–6. doi: 10.18564/jasss.4531

35. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv preprint* arXiv:170706347 (2017). doi: 10.48550/arXiv.1707.06347

36. Andrychwicz M, Raichuk A, Stanczyk P, Orsin M, Girgin S, Marinier R, et al. "What matters for on-policy deep actor-critic methods? A large-scale study," in *9th International Conference on Learning Representation*. OpenReview.net (2021). Available online at: https://openreview.net/forum?id=nIAxjsniDzg