Journal of Experimental & Theoretical Artificial Intelligence

ISSN: 0952-813X (Print) 1362-3079 (Online) Journal homepage: https://www.tandfonline.com/loi/teta20

A computational model of scientific discovery in a very simple world, aiming at psychological realism

Francisco Lara-Dammer, Douglas R. Hofstadter & Robert L. Goldstone

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE
https://doi.org/10.1080/0952813X.2019.1592234

Taylor & Francis
Taylor & Francis Group

ARTICLE

A computational model of scientific discovery in a very simple world, aiming at psychological realism

Francisco Lara-Dammer[a,b,c], Douglas R. Hofstadter[b] and Robert L. Goldstone[c]

[a]Departamento de Matemática, Escuela Politécnica Nacional, Quito, Ecuador; [b]Center for Research on Concepts and Cognition, Indiana University, Bloomington, IN, USA; [c]Department of Psychological and Brain Sciences, Indiana University, Bloomington, IN, USA

ABSTRACT
We propose a computational model of human scientific discovery and perception of the world. As a prerequisite for such a model, we simulate dynamic microworlds in which physical events take place, as well as an observer that visually perceives and makes interpretations of events in the microworld. Moreover, we give the observer the ability to actively conduct experiments in order to gain evidence about natural regularities in the world. We have broken up the description of our project into two pieces. The first piece deals with the interpreter constructing relatively simple visual descriptions of objects and collisions within a context. The second phase deals with the interpreter positing relationships among the entities, winding up with elaborated construals and conjectures of mathematical laws governing the world. This paper focuses only on the second phase. As is the case with most human scientific observation, observations are subject to interpretation, and the discoveries are influenced by these interpretations.

ARTICLE HISTORY
Received 29 December 2017
Accepted 27 December 2018

KEYWORDS
Human perception; context; scientific discovery; computational model; visual observation; conjectures

Introduction

Everyone, from children to scientists, makes discoveries, and every discovery is the result of perception of some situation in the world. Situations come in all sorts of forms, ranging from tangible objects to computer-simulated events, from rudimentary to highly sophisticated. The path to discovery starts out in a place common to all discoverers: all worlds seen by discoverers are unfamiliar to them at first (otherwise, there is nothing to be discovered). By studying various mechanisms or techniques of discovery, including those of novel perceptual interpretation, we hope to shed light on the intricacies of discovery. However, not all types of discovery are covered herein. Our focus will be on *scientific* discovery.

This paper describes a computational model of scientific discovery that simulates visual perception in a context-sensitive manner. Our approach is based on two premises: (1) that discoveries are influenced by perception, and (2) that perception depends on context.

This paper constitutes the second part of our project on computational modeling scientific discovery. The first part (Lara-Dammer, Hofstadter, & Goldstone, 2017) focused on how the context-dependent perception of properties of events unfolds over time. In that paper, we showed that how a human perceives situations gives rise to interpretations, and we proposed a system to simulate the activity of discovery. We named this system NINSUN, which stands for 'Novel Interpretations of Scientific Understanding.' NINSUN is an interpreter of world situations, somewhat

CONTACT Francisco Lara-Dammer flaradam@indiana.edu Department of Psychological and Brain Sciences, Indiana University, Bloomington, USA

in the spirit of an interpreter of dreams, of music, or of any other world that is observable. The observed world is created using *Tricycle*, a separate program, although NINSUN has the ability to visually observe and explore Tricycle-realized worlds via experimentation and manipulation.

In keeping with our philosophy of psychological realism, we have focused on providing NINSUN with the ability to derive simple discoveries akin to those made by people. Given that we are not trying to automate the process of discovery, our goal is not to show that our system can make a large number of highly sophisticated discoveries. Instead, NINSUN has discovered a few simple patterns, such as the fact that for molecules of a gas moving in a container, the number of wall collisions per minute multiplied by the perimeter of the container is a constant.

This might sound humble, but this simple discovery was made by NINSUN without presupposing properties like molecule velocity, direction, or even persisting molecules over time. Instead, these properties were derived by implementing psychologically plausible processes for determining object identities, motion, and interactions. If, for instance, a ball hits an object and the ball changes direction immediately after this event, what causes the change of direction can be inferred using formal techniques for the determination of causality. For example, Ayazoglu, Yilmaz, Sznaier, and Camps (2013) use graph theory and topology. These approaches usually focus on issues such as algorithm efficiency and psychological plausibility is of only peripheral interest. In Lara-Dammer et al. (2017), we stated that the perception of causality is modeled using a technique related to Michotte's work on causal perception (Michotte, 1963), which is more faithful to our psychological approach. In the next section we will address our philosophy of discovery, which stands in stark contrast to some artificial-intelligence (AI) philosophies.

We will be referring to simulations of discoveries throughout this paper (which are recorded in videos), in the same way we did in Lara-Dammer et al. (2017). For viewing convenience, we have posted these videos in the archives located in the following website:

http://www.indiana.edu/~pcl/2016/03/ninsun/

Our project was motivated in part by the large number of possibilities for cognitive exploration that are offered by the activity of discovery. This activity manifests itself in multiple ways, such as a desire to observe the world carefully and to manipulate it for long periods, and the occurrence of occasional stimulating moments that make an observer curious, as well as of rarer and more exhilarating 'aha' moments.

A second motivation underlying the creation of this computational model is to help us understand and predict humans' interpretations (or misinterpretations) of scientific situations when they are watching computer simulations. Since our model simulates visual scientific discovery (i.e., information is extracted from spatially and temporally structured events, as opposed to being extracted from symbols and formal expressions), the interpretations that we focus on involve visual perception.

Another motivation for our modeling approach is to tackle some of the important challenges involved in the compression of information. Once a visual display is observed, the system infers the mechanisms that give rise to what it sees. These mechanisms can hopefully be described by a small set of rules. Furthermore, capturing the essence of a situation might lead to a generalization of the situation and consequently a scientific discovery. The program then can create variations of the original situation based on the generalization it has formed.

Some AI approaches to the modeling of discovery

Given the technological advances of recent years in computers, one might well wonder if existing artificial intelligence programs are not already making compelling and psychologically realistic scientific discoveries. There are, in fact, AI programs that are making some kinds of discoveries, but the key question is *how* they do so, rather than *what* they do. AI models of discovery need not be inspired by the perceptual activity of the brain. This is partly because they are often designed to carry out certain tasks without any regard to how humans (or animals) do so. Simulating human behavior is often not relevant to their goals. A good example is the domain of chess. Today there are programs that excel at playing chess, but their underlying mechanisms bear no similarity to *human* chess-playing. A crucial aspect of human chess-playing is error-making. This aspect of human thinking in chess is not part of current computer chess programs (Kasparov, 2010; Lara-Dammer, 2009; Linhares & Brum, 2007).

In this regard, our computational model of scientific discovery differs from AI programs such as BACON (Langley, Simon, Bradshaw, & Zytkow, 1987) and AM (Lenat, 1979), which do not simulate the perceptual aspect of discovery, or at least this aspect is secondary for these programs. Their discoveries are not based on observation of the world but on tables of data that are given to them. For example, BACON is claimed to simulate how chemist Robert Boyle (1627–1691) derived his famous law $pV = C$, where p stands for the pressure exerted by the gas inside a container, V for its volume, and C represents a numerical constant. Boyle's original numerical data were given to BACON as a table with columns p and V. As Langley et al. (1987) reported, BACON was able to derive Boyle's law from the table by following certain heuristics. Using the same techniques, BACON was able to make other discoveries as well.

Douglas Lenat's AM program worked with a different sort of symbol manipulation and in a different domain. As Lenat explained (Lenat, 1979), the domain of number theory was chosen in order to avoid 'uncertainties in the raw data.' However, the existence of such uncertainties in data is precisely one of the key components of perception that lead different scientists to different interpretations, and consequently to different theories. Given the hard-edged quality of integers and their relations of divisibility, Lenat was successful to a certain extent in letting AM apply heuristics and thereby discover some theorems of number theory. However, some cognitive scientists have argued that human mathematical thinking is grounded in the human experience of space and time, and in the way that humans perceive objects, events, and transformations, rather than merely making use of formal operations (see, for example, Dehaene, 1997; Lakoff & Núñez, 2000). Indeed, perception plays a role even in mathematics, a domain which to some people seems perception-free. (The mathematician David Hilbert (1999) was one of the first defenders of perception-free mathematics: 'Geometry, like, arithmetic, requires only a few simple principles for its logical development.') For example, a psychological aspect of the positive integers is that they can be conceived of as lying on a straight line, and along this line the prime numbers are fairly dense at the outset and then grow sparser and sparser as one's gaze moves away from the origin. Yet the set of prime numbers is infinite (a rather surprising fact), and they generate all the integers in the sense that any integer can be represented as a product of primes. Being rare is a key psychological component of discovery because it triggers surprise and curiosity, and motivates people to search for further rare entities of the same sort (see Lara-Dammer, 2009).

The heuristic used by BACON to derive Boyle's law and other laws (e.g., Kepler's third law) can be expressed roughly as 'Whenever one variable increases and another decreases, examine their product,' and it is psychologically plausible. The model's psychological implausibility lies not in such heuristics, but in the way it obtains its data. BACON was fed pre-digested data that summarized large amounts of thought, experimentation, trial and error, and observation, which in some cases came from more than one generation of scientists. These efforts were preceded by enormous amounts of trial and error. A famous hypothesis made by Kepler captures the complex psychological forces and often self-contradictory efforts that are such common ingredients of scientific discovery. This was his attempt to find a law explaining the radii of the orbits of all the planets by imagining that the orbits of the planets were equators of concentric spheres in which the five regular polyhedra were inscribed. What gave rise to this strange image in Kepler's mind?

Such thoughts probably arose because at that time it was believed that the solar system had six planets, and Kepler, who knew geometry well, knew that there are exactly five regular polyhedra, and so he imagined a correspondence. He started with Jupiter, imagining that a sphere contained its circular orbit, and then he inscribed a cube inside this sphere. This second sphere corresponded to the orbit of Saturn. He repeated this process, using the remaining polyhedra in the order:

tetrahedron, dodecahedron, icosahedron, and octahedron (see Kline, 1972). For some time, Kepler believed in his hypothesis because the numbers in his model very closely matched the observed data.

What is crucial to note here is how Kepler's perception of the solar system and his knowledge of geometry influenced his discovery (which was right within a 5% of error, but even at that time an error of that size was not acceptable). His mystical beliefs about the creation of the universe and his great familiarity with polyhedra strongly influenced his ideas about what mathematical ingredients might go into scientific laws. He was not willing to give up on this hypothesis for some time. Eventually, however, he abandoned it because of its discrepancies with observational data.

Kepler also eventually abandoned the idea of the planets moving in circular paths, an idea that was very deeply ingrained at that time. Leaping from the idea of circular orbits (possibly including epicycles) to that of elliptical ones was a giant step of imagination requiring extreme boldness. The notion of planets moving in circular orbits had come from the ancient Greeks and had been deeply reinforced by Copernicus (1473–1543), and very few scientists were willing to question the legitimacy of these ideas. 'Is it possible that the planets do not move in circular orbits but in elliptical ones?' is a question that would not have sprung to anyone's mind without immense psychological pressures. After all, no one had asked such a question in all the 1800 years that followed the development of the theory of conics by the Greeks (Eves, 1969). The circle had been considered a perfect shape since the Greeks. Imagining that orbits were not 'perfect' would have required great courage, since circles were so deeply built into astronomical tradition.

Kepler was not given a table from which he could derive elliptical orbits by applying some heuristics. Instead, he had to observe the world carefully and for a long time, taking notes, organizing his measurements, until he finally realized a better model for the trajectory of the planets. We are not suggesting that our model is capable of emulating complex mental conflicts of the sort that arose in Kepler's mind when he was making his discoveries. We are merely suggesting that perception influences the way humans make discoveries, and that error-making is an important part of the process.

The way BACON was designed to use pre-digested data without engaging in perception of situations in the external world has been described as '20–20 hindsight' (Chalmers, French, & Hofstadter, 1995). Obviously, the human activity of perception is still too complex a phenomenon to be faithfully modeled by a computer. However, it would be a significant contribution to make a model of discovery that attempts to take into account current theories of human perception.

Our approach to the modeling of discovery

It was stated above that our model of discovery differs fundamentally from traditional symbolic AI programs (such as BACON (Langley et al., 1987) and AM (Lenat, 1979)) both in its imagery based motivation and in its psychologically driven mechanisms. How the data are processed is perhaps the clearest manifestation of this difference. Let us turn again to the example of BACON's discovery of Boyle's law, to compare the two approaches. Whereas BACON starts its discovery from a numerical table with columns p and V, our interpreter starts out a possible path of exploration by observing a world consisting of a container and various small balls bouncing off of the container's walls and also possibly off of each other. Whereas BACON applies its heuristics to the static numerical data in the table, our interpreter observes as many variables and relations as it can in the container. In the interpreter, variables are created based on a perceptually constrained apparatus that seeks patterns in a temporally and spatially structured environment with individual objects. For example, until persistent objects are posited, speed is not even a definable variable. Persistent objects are determined by spatial and physical similarity constraints and one-to-oneness.

Among the variables and relations that have a chance to be observed by NINSUN are the size of the container and how frequently the balls strike its walls. Upon increasing the size of the container, it might observe that the number of hits against the wall in a given time decreases.

From these observations, it might suspect the existence of a possible relation '*pV* is constant,' where *p* in this case means the frequency of the balls hitting the walls and *V* means the size of the container. This possibility is due to the fact that NINSUN has heuristics such as 'when one variable increases and another variable decreases then these variables might have a constant sum or a constant product' (a heuristic with some similarity to BACON's heuristics).

Of course, there are other variables and relations that can be observed, such as the sizes and speeds of the balls, and the shape of the container. However, *p* and *V* are the crucial variables for the discovery of Boyle's law. Should NINSUN pay attention to those two variables, there is a good chance that the relation $pV = C$ can be discovered as a 2-D version of Boyle's law. It is worth mentioning that for NINSUN, *p* is not just an atomic quantity (i.e., a fixed descriptor associated with a situation). It is grounded in the perception of actual collisions of balls against walls in the simulation. NINSUN uses perceptual heuristics to interpret scenes as consisting of persistent, moving objects that interact with each other.

In order to be able to apply the above-described heuristic, NINSUN has to be able to act on the world and observe consequences when it is modified. For instance, in the example, NINSUN needs to try increasing or decreasing the size of the container, and then it has to make new observations.

Although the heuristic helps NINSUN make the discovery, there is nonetheless a chance that Boyle's law might not be conjectured, because the observed numerical values or *p* and *V* might not fit the equation $pV = C$ closely enough. However, if they are close enough, NINSUN will take a risk and make a guess. The interpreter has parameters of credibility for accepting or rejecting a possible conjecture. Letting credibility be a parameter inevitably means that NINSUN might sometimes make erroneous guesses, similar to Kepler's model of the solar system using nested regular polyhedra inscribed in spheres. The relatively small numerical error allowed Kepler to believe for some time that his conjecture was true.

The credibility parameter is hard to tune because it depends on the situation. This is exemplified by Boyle's conjecture of his law and Kepler's conjecture of the solar system as involving polyhedra. Most of the *pV* values in the data used by Boyle to derive his law of gases were as far as 10% away from the mean (i.e., what would be expected if *pV* were a constant), and yet he accepted the idea that *pV* was constant see Langley et al. (1987). Kepler's solar-system values, on the other hand, involved errors of less that 5%. This contrast shows that in astronomy, to have credibility a conjecture must have great precision, but that when gases are concerned, the credibility of a guess does not require so much precision (or at least that was the case a few hundred years ago).

All of this implies, of course, that discovery is a process subject to error. This is a natural consequence of how humans perceive the world. For this reason, in our model, scientific discovery is simulated by taking into account certain facts about human perception (for the sake of simplicity, we limit ourselves to visual perception). Thus, NINSUN's observation of the world starts out with the object-tracking problem. This kind of low-level vision process is bypassed in most AI programs because it has a high computational cost and can easily give rise to errors. However, it is precisely the possibility of making such errors that lends it such crucial importance in the simulation of human perception and discovery.

Although the simulation of a microworld is computationally difficult, since it involves large amounts of sophisticated computation, it is actually the *interpreter* that is much more intricate to model. This is due in part to the fact that the worlds Tricycle creates are crisp and well-defined, whereas to interpret them in real time, in the way a human mind does, is not. Also, cognitive modeling requires large amounts of sophisticated computation. The total amount of computation (counting both Tricycle and NINSUN) makes our model very expensive in terms of computation. This is to be expected, however, since it is central to this project that the interpreter should be a *psychologically realistic* model. This means that *how* it gives its interpretations is what counts, not merely the fact that it comes up with interpretations. On the other hand, the fact that the interpreter was intended to be a model of human cognitive abilities does not mean that it actually performs like a human. Our claim is simply that we have implemented a set of computational cognitive processes whose degree of similarity to *human* cognitive processes is the object of our study.

An example illustrating our approach to psychological realism is collision detection. If collision detection *per se* were all that we wished to model, it could be taken care of by Tricycle instead of in the interpreter. However, what matters in this project is the interpreter's observations and what it derives from them. NINSUN does not know the precise laws of physics that underlie the motions of the objects it is observing. Detection of collisions is not trivial, especially given that even the tracking of a moving ball by itself is error-prone. As was stated above, Tricycle does not pass information about relations of the objects in the world to NINSUN; it is rather the job of NINSUN to *derive* such relations.

The inherently complex nature of human perception means that this project is constantly evolving. This paper presents some of the features that have been developed so far. Even though these are only a small fraction of our original goal, the current model already exhibits some interesting cognitive properties.

Among the advantages of a computer simulation of discovery is the easy manipulation of variables and the observation of their effects on the entire system. One of the features of this project is the fact that the physical world is not fixed. It can be modified by NINSUN during run time.

The shapes, tools, and commands available in Tricycle and the way that they facilitate the creation of new worlds is reminiscent of the theory of scientific discovery as *dual search* (SDDS) proposed by Klahr and Dunbar (1987). According to these authors, the generation of hypotheses in scientific discovery is performed by simultaneously exploring the space of possible hypotheses and the space of possible experiments. For instance, in the example of the world of three balls inside a container given above, NINSUN can change the world in several ways, such as these:

- balls can be created or destroyed;
- the container can be made bigger or smaller;
- the container's shape can be modified;
- the speeds of the balls can be modified;
- walls can be added or removed.

The new situations that emerge are key aspects of our modeling of understanding and creativity in the process of discovery. Modifying the world and observing it before and after the modification is an essential aspect of experimentation. In order to simulate experimentation, one needs to take into account various aspects of these two essential processes. For example, in order to compare the states that occurred before a modification with a current state, the interpreter needs ways to memorize these already observed states. Also, since memory is a limited resource, it needs to store the most relevant changes in a compressed manner. Therefore, our simulated observer needs to have a sophisticated way of storing data as well as a sophisticated mechanism for retrieving data. NINSUN's storage system is described in the next section.

Experimentation through world modification

The way NINSUN is designed to make scientific discoveries is based on the premise that in order for people to come to know their world, they need to watch what happens when various changes are made to the world. Modifying the world is thus at the very heart of human everyday exploration and scientific experimentation (Weinberg, 2015).

In accord with this fundamental principle, we designed the interpreter to be able to make changes in the world it observes and see what happens. Of course, *how* people manipulate their world is very complex, and different people do it in very different ways. The interpreter has different modes, allowing it to modify the world in either a naive way (possibly in the way children or inexperienced people do), changing more than one variable at a time, or in a more sophisticated way, by changing only one variable at a time (Klahr, 2000).

Regardless of how NINSUN manipulates its world, it is crucial for it to have access to the state of the world *before* the modification and to compare that to the *current* state of the world. People's observations are subject to errors, but in order to compensate for their errors, they can modify the world and return it to its original state, and they can carry out similar modifications several times. Our interpreter thus possesses a mechanism allowing it to manipulate the world several times and to recover from potentially erroneous observations. The data structure that records its explorations (which are in fact *experiments*) is given the name *history of events*.

The history of events

The history of events in the interpreter is a dynamic data structure that grows whenever NINSUN modifies the world, and is used by NINSUN to remember its past observations. This structure does not have a pre-set size. Its size starts at zero, and depends thereafter on how NINSUN decides to modify the world.

One dimension of the history of events is a list called 'modifications.' This list stores the actions that NINSUN carries out (e.g., 'modify the area of the container'). The modifications list also stores the number of times NINSUN needs to make a modification before it observes a change in the category of any dependent variable.

The second dimension is called 'observations,' and is also a list. The observations list stores the actual observations of the dependent variables of the system (e.g., the number of ball–wall collisions per minute), and the value of the independent variable that was modified (e.g., the length of the perimeter of the container). Each entry in the modifications list points to the observation just before the modification was carried out for the first time.

For the sake of clarity, we will describe this structure pictorially. It is a large structure, and a full visualization would cause confusion rather than clarity. Showing a small but representative part of it is the best way to get the idea across.

Suppose NINSUN is running a sequence of observations, and at this point it has modified the length of the container's perimeter three times, the number of balls four times, and now is modifying the shape of the container from circular to triangular. (The number of observations is one more than the number of modifications of the same type (which we will refer to as *repetitions*) because the first observation is made before the first modification is carried out.) At the time when NINSUN is modifying the shape of the container from circular to triangular, we cannot know how many times it is going to change the shape. This will be up to NINSUN and will depend on the way it is observing, as is explained below (in Section 4.3). For every modification of the independent variable *shape*, the *number-of-repetitions* variable will be increased by 1. So far the value is 1.

At this specific moment, the history of events might look something like Figure 1. On the top row, all the modifications are stored. This row also contains the address of the location of the first observation before the modification (shown by the arrows from the modifications list to the observations list) as well as the number of repetitions of each type of modification.

The values of the dependent variable 'wall collisions/minute' have been made explicit. NINSUN also keeps track of the values of the independent variables. To suggest this idea, the changes of the independent variables are shown just below the list labeled 'Wall collisions/min.' The diagram informs us that the first four observations of the independent variable 'area,' after the size of the container was modified three times, were *medium, large, medium*, and *small*. The modifications for the independent variable 'number of balls' were 3, 2, 5, and 4. The values for the independent variable 'shape' have so far been *circular* and *triangular*.

The red arrows on the right side of the lists indicate that the history is probably going to expand in the future. Theoretically, the expansion can continue indefinitely, but it is convenient to stop the simulation after NINSUN has made a hypothesis and confirmed it.

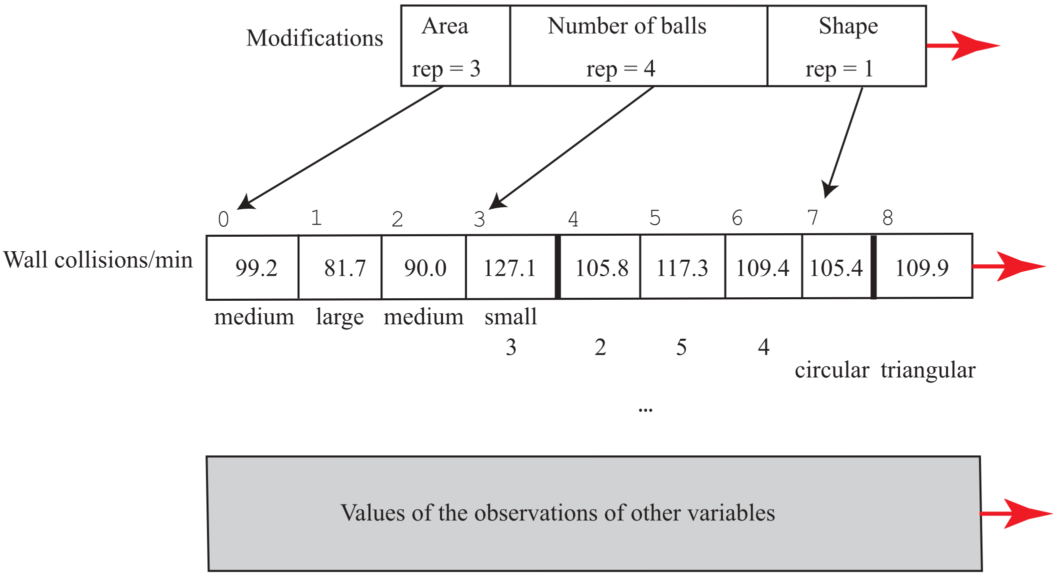

Figure 1. A representation of a portion of the history of events (in memory space and time) just after the shape of the container is modified.

The data in the history of events are recorded in precise synchrony with the other processes that are running in parallel. For example, consider the act of observing the stability of the system. (See Section 4.2.) NINSUN needs to wait for the system to become stable before writing the data into its memory. Likewise, certain modification processes take some time, such as altering the size of a container. NINSUN needs to gradually modify the size before making the next observation of the number of wall collisions per minute. Modifying the number of balls is special because the deletion of a ball might chance to coincide in time with a collision involving that ball. In this case, NINSUN needs to adjust to the new reality without producing contradictions (such as a nonexistent ball colliding with another ball). One possible strategy for attaining logical consistency is to ignore such a collision.

World stability

When the world is modified, there is usually a transition period during which it is unstable. While the instability lasts, there is no point in collecting data. In many everyday situations, humans are familiar with this type of instability and know how to deal with it. For example, when someone adds ice to a drink, they need to wait a while, until the drink is cool. Based on experience, humans know that the heat needs some time to be transferred before an equilibrium is reached, at which point they can enjoy the cool drink.

In general, in order to modify their world and observe the effects of the modifications, humans need to have a sense for when the world is unstable and when it is stable. The computational task of recognizing an approximate equilibrium is neither obvious nor simple, but what counts is finding a moment when it is *likely* that this point has been reached. Sometimes humans make mistakes in guessing about equilibrium, and they proceed either before equilibrium has been reached or long after it has been reached. It is not always easy to tell when the most recent modification has ceased to affect the current state of the world.

In our model, we need the notion of stability in order to be able to make reliable qualitative measurements; otherwise, the data measured by the interpreter could be meaningless. Our interpreter model is similar in this aspect to the qualitative physics described in (Forbus, 1984). The main heuristic for detecting an equilibrium is repeatability of measurements; that is, the interpreter makes sure that observing the world several times results in roughly the same observations.

A rigorous and general definition of 'stable world' would be too difficult to implement and too expensive to compute. We therefore take a rather simplified and intuitive approach, applicable to the microworld of balls and walls, in which we assume that a world is stable if the range of values taken on by dependent variables lies below a fixed threshold. We define a world as *unstable* when the most recent observation of (at least) one of the dependent variables differs from the current observation by more than a specified small value δ, assuming that those observations both occurred within a small time interval Δt.

If the most recent value of a dependent variable is u and the value of the same variable in the current observation is v, and r is the actual average value of that variable over Δt, then NINSUN assumes that the world is stable over the interval Δt if for any observation, the values u and v satisfy the inequality

$$|v - u| \leq 2\delta r.$$

This inequality can be obtained from the relative error formula as follows. We define the relative errors δ_1 and δ_2 as $|v - r|/|r| = \delta_1$ and $|u - r|/|r| = \delta_2$. By the triangle inequality, we have $|v - u| = |v - r + r - u| \leq |v - r| + |u - r| = \delta_1|r| + \delta_2|r| \leq 2\delta|r|$, with $\delta = \max\{\delta_1, \delta_2\}$. (In an unstable world, the inequality would be violated for some values of u and v.)

Since r is unlikely to be known by an observer, it needs to be estimated in some manner. A natural estimate is given by: $r \approx (u + v)/2$. Thus, the world is probably unstable if

$$|v - u| > \delta|u + v|.$$

The value δ is a parameter in the interpreter program called *sensitivity to change* and is linked to how sensitive an observer is to the perception of change in the microworld of balls and walls. This value influences NINSUN's perception in two ways, which we now explain with the help of a small world consisting of three balls moving at medium speeds. One way consists of observing this world for a few seconds, after which we increase the speed of the balls to some degree, without telling the observer that we made that change. Would the observer notice the change? (If δ is large, it is likely that the answer is no.) The other way is that we do not make any change in the world. Is it possible nonetheless that the observer might think that there had been a change? (If δ is small, it is likely that the answer is yes.)

If we make the parameter δ small, NINSUN becomes sensitive to small changes. However, being too sensitive is problematic since NINSUN will be prone to seeing the world as unstable merely because of the stochastic nature of the world built by Tricycle. For instance, in a world of three moving balls, one can reasonably expect to see 18 collisions in one minute and 14 collisions in the next minute. In this case, there is a variation of about 25% between observations. But if the value of δ is too low, NINSUN might see this world as unstable. A large value of δ is also problematic. NINSUN becomes insensitive to change, which can result, for example, in its making observations before the world has reached stability, and hence reaching incorrect conclusions.

Roughly, δ represents half of the theoretical maximal percentage of change in each of the dependent variables that can be observed by the interpreter without noticing the change, and hence without noticing any instability in the system. The selection of a value of δ that allows effective discovery depends on the nature of the microworld. For the microworld of walls and balls, the value $\delta = 0.2$ results in acceptable behavior by the interpreter, when the number of balls is between three and five. Occasional observations of instability happen with this value, but this is acceptable, since the world is indeed stochastic, and since collisions are sometimes seen as pass-through events (defined in (Lara-Dammer et al., 2017) as a situation in which two objects pass through each other; for example, if they are made of a gaseous substance) and vice versa.

The perception of stability is also connected to the time interval Δt mentioned above. For example, the ability to remember how many ball–ball collisions have happened recently on average depends on the length of this time interval, and consequently on how many observations we are considering. NINSUN updates the number of collisions every five seconds. In order to calculate the average, it needs to remember what happened in the previous five seconds, and in prior intervals, as many as it can (which is a parameter of the system). If humans observe the situation for a period of time, say twenty seconds, they will easily get a sense for how many collisions happen per minute.

In general, the longer humans observe a situation, the higher the accuracy of their stability observations, even though they are not actually counting the number of collisions. This suggests that it is plausible to allow the interpreter to average the number of collisions per minute observed during several updates. The maximum number of observations that NINSUN can make to find averages is called *collisions-per-minute memory*, and this number is a parameter of the model, which can be set anywhere from 2 to 12. A value selected in the range from 5 to 10 gives reasonable behavior of the perception of stability in the world with three balls. Lower values affect the stability, and higher values work better but need longer periods of time to become effective.

Although 'stability of the world' is not a quality of a situation that we humans consciously pay attention to, our unconscious minds must be frequently detecting stability in some manner, since not doing so would jeopardize our survival. Consider, for example, an earthquake. Such an event would cause almost any human to realize that their environment had lost stability, and they would start to run for safety. Likewise, NINSUN is at all times observing the stability of its microworld, and is constantly sending signals to the outside world that reflect its perception of the state of stability of its microworld.

In order to illustrate how NINSUN detects instability, there is a video of a simulation with three moving balls available under 'Stability V1.' In this simulation, the parameter δ (controlled by the 'Sensitivity to Change' slider) was initially set to the value 0.4. Also, the balls were initially moving with medium speed (0.3). The speed of the balls was then slightly increased to 0.35, and NINSUN did not notice the change. (This can be inferred by observing the cone on the left top resting on its base during all this time.)

The weight of the direction condition and the chance of confusing collisions with pass-through events (discussed in (Lara-Dammer et al., 2017), in the section entitled 'The Ambiguity of Collisions as Opposed to Pass-through-each-other Events') are also linked to stability. We just mentioned that setting the δ parameter at 0.2 gives good performance for the interpreter in recognizing stable worlds. However, this presumes that the network parameters are also well calibrated for the situation. In this particular example dominated by collisions, the interpreter will be better able to correctly interpret the world as containing many collisions when the weight given to the parameter biasing object correspondences in which objects move in a single direction is set to a low value. Collisions cause balls to change their direction of movement frequently, so a strong bias to see balls as moving in constant directions will lead to misinterpretations.

An abrupt change of this parameter was made in order to show the disruption of stability in this simple microworld. In this simulation, NINSUN initially sees a stable world where collisions are dominant. For this, the parameter δ was set to 0.2 and the weight of the direction condition was manually set to 0. After a few seconds, the weight of the direction condition was reset to the high value of 0.7. As a consequence, NINSUN started seeing some pass-through events, and a few seconds later, it decided that the world was unstable (at least for some time). A video of this simulation is available under 'Stability V2.'

We close this section by noting that the addition of noise to the system does not affect the behavior of stability, as long as this addition does not exceed 20%, approximately. The reason is that, as we stated in Section 3 of our paper (Lara-Dammer et al., 2017), at the level of object identification, our system utilizes an auto-associative connectionist network based on a selection-of

-winners philosophy. The winner units are what determine which object corresponds to which other object. This selection is probabilistic in nature. Adding noise is one of the techniques that helps the network to select the winners. This architecture is in agreement with the capability of the mind to recognize blurred objects. In addition, as we have emphasized in our project, context plays an important role in the identification of objects.

As an illustration to help understand how the presence of noise relates to stability in our computational model, consider the following description recounted by second author. One day of the year 2018, above the sky of Silicon Valley, a cloud displaying the shape of the symbol '10' appeared, leaving many of its viewers perplexed, at least for a short time. Of course, the cloud in itself does not have any shape because is a gaseous substance and it is intrinsically blurry. However, a human can easily associate those shapes to the binary symbols, especially if those clouds were seen in the context of the Silicon Valley. The 'noise' is ignored and the interpreters see the binary symbols with no ambiguity.

Number of repetitions

Since visual observation of the world affords only a very rough sense of the numerical values of the variables, human explorers would tend to repeat their experiments and observations in order to optimize the quality of the data they obtain. Even a simple task will be carried out several times in order to gain certainty about the judgments derived from an observation. To mimic this, our interpreter was designed taking into account the fact that it should not rely on a single observation, since one observation could give erroneous values or judgments.

On the other hand, a human observer cannot carry out endless repetitions, since that would take endless energy and time. A compromise must be found between the opposing pressures of wasting time and making errors. Such compromises are frequent in everyday life. For example, in chess, players need to try to optimize their thinking time. If they move too fast, they are more prone to making mistakes and thus losing the game. But if they move too slowly, they will run out of time and lose the game for that reason.

What strategy does a human follow to find a reasonable error/time trade-off? This question has no simple answer, because there are many factors involved in making decisions. Among these factors are: one's degree of interest in a project, one's supply of physical and mental energy on a particular day, etc. However, what is crucial for human observers is that they reach some degree of certainty before they stop making measurements. The strategy used by our interpreter is similar: it is to repeat a given type of modification until the observations of all the dependent variables allow NINSUN to observe a systematic pattern that reliably arises as a consequence of the modifications made to the world. The most important kind of patterns that we have been focusing on in our project are those of *growth* and *shrinking* of some dependent variable, which we will henceforth call simply 'growth.'

There are essentially just three possible kinds of 'growth patterns' connecting the behavior of a dependent and an independent variable. Of course in some situations things are more complex than this, but we made this assumption in line with our policy of keeping the world simple:

(1) either both variables increase or both decrease; or
(2) one increases while the other decreases; or
(3) the dependent variable remains roughly constant with variation of the independent variable.

We will use the terminology *simultaneous growth* to refer to the first type of pattern, *opposite growth* to refer to the second type of pattern, and *constancy* or *no growth* to refer to the third type of pattern.

Table 1. A set of observation values used to illustrate NINSUN's strategy of repetition. (The abbreviation 'og' refers to *opposite growth* of the independent variable 'area' and the corresponding dependent variable.).

	Conclusions about Monotonicity						
	obs. 1	obs. 2	obs. 3	obs. 4	rep. 1	rep. 2	rep. 3
Area	Medium	Large	Medium	Small	inc.	dec.	dec.
Wall collisions/min	99.0	79.2	108.0	127.2	dec. (og)	inc. (og)	const.
Ball collisions/min	78.0	32	48.0	102.0	dec. (og)	inc. (og)	inc. (og)
Pass-through events/min	12.0	0.0	6.0	12.0	dec. (og)	inc. (og)	inc. (og)

We illustrate this idea in the microworld of walls and balls, using an actual run with the parameter $\delta = 0.2$. At the beginning of this run, NINSUN decided to explore modifications of the *area* of the container. The area has been changed several times, as is indicated in Table 1.

First, let us draw attention to the dependent variable 'wall collisions/min' (for brevity, 'wall collisions'). One thing that jumps to the eye in the table is that apparently it does not show categories, but numbers. This is not so. These numbers become categories in the model after some more computation. The reason is that the category of 'wall collisions per minute' is not as simple as the category of 'area.' Consider for example, a container of medium size that has three identical small balls moving, and where a human can judge that the number of collisions with the wall is moderate (or medium). If we add three more balls under the same circumstances, the number of collisions with the wall doubles, and yet the observer still judges that there is a moderate number of collisions with the wall. This indicates that numbers alone are not sufficient to model the categorization of this type of variable.

The strategy that we used for this categorization is based on the notion of *variation of the variable*, comparing the difference between the values before and after the modification with the value before the modification. (This is similar to how relative errors are computed in numerical analysis.)

Observation 1 of the variable 'wall collisions' is 99.0, and after the first modification, observation 2 is 79.2. The fact that 99.0 is larger than 79.2 is only the first step in drawing a conclusion. The actual values 99.0 and 79.2 are not as important as the *variation* taking the value δ into account. What matters the most is how much we have *now*, as compared to what we had *before* the modification. Since the variation $|99.0 - 79.2| = 19.8$ is greater than the value $\delta * 79.2 = 15.8$, only now does NINSUN conclude that the number of wall collisions decreased as a result of increasing the area of the container.

The two steps just described make NINSUN see that the wall-collisions variable *decreased* as a result of the first modification. Since the independent variable 'area' *increased*, the conclusion is that the area of the container and the number of collisions are in *opposite growth*. However, since this is just one observation, this first conclusion is not solid. It is just the beginning of a suspicion.

After a second modification of the area, which this time consists in *reducing* its value, NINSUN observes that the number of wall collisions is now 108.0. After the second step (as in the first modification), this time it sees the number of wall collisions *increasing* as a consequence of the modification. Since the area was *decreasing*, the conclusion is once again that *area* and *number of collisions* are in *opposite growth*. Still, the conclusion is not solid, but this time the suspicion of an opposite-growth pattern is greater than the first time.

After the third modification, NINSUN's fourth observation of wall collisions is 127.2. Since the change is relatively small for the parameter $\delta = 0.2$ because $|108.0 - 127.2| = 19.2$ is less than $\delta * 108.0 = 21.6$, NINSUN sees this as a case of *no change*, and declares it constant. Another way to put this is that the numbers (of wall collisions) '100.8' and '127.2' fall in the same category for NINSUN. At this stage of its processing (i.e., when NINSUN is doing rough modifications of the world), NINSUN is designed to see *categories* rather than *numbers*, and that is why the *area* variable in the table shows categories of size such as 'medium' and 'small,' rather than numerical values.

This is in agreement with the qualitative physics mentioned in Section 4.2. (It is also desirable that our program, after reaching a conclusion in a qualitative style, should enter a new phase with a more quantitative approach in which the interpreter could measure variables with higher precision by using Tricycle. This would increase the level of certainty of its conclusions (Lara-Dammer, 2009).)

Even though the last observation on its own would suggest that the number of wall collisions was constant, the overall conclusion about wall collisions and area is that they are in *opposite growth*. After all, the two previous conclusions were that these variables exhibit opposite growth. Therefore, a majority vote for the relation 'opposite growth' between *area* and *wall collisions* has been reached.

Similar conclusions can be drawn about the other two dependent variables in regard to their pattern of growth. However, 'pass-through events' is especially interesting because all such events were 'hallucinated' by NINSUN, in the sense that this particular world had only collisions and no actual pass-troughs. But as we know, NINSUN has to believe in what it 'sees,' independently of reality.

In this example, the number of modifications was only three, because there were almost no conflicts among the conclusions drawn by NINSUN. In general though, there can be conflicts. For instance, it is possible for NINSUN to conclude that *area* and *number of wall collisions* are in *opposite growth* in one observation, and in *simultaneous growth* in another observation. When such conflicts arise, NINSUN will continue to modify the area of the container until its observations of the dependent variables 'wall collisions,''collisions among balls,' and 'pass-through events' exhibit a reliable pattern of relatedness.

Our interpreter's exploration strategy requires modifying the world at least twice. (In the case just described, it did so three times.) After having made at least the mandatory number of modifications, NINSUN tries to find a reliable pattern, and if it fails to do so, it will continue making world modifications.

Data compression and rule derivation

After trying out various modifications of the world, a human would want to record the key observations that had been made. However, the relevant aspects of the experience would have to be *compressed* to be useful. Otherwise, the person would have to carry out more repetitions of observations and more modifications of the world, with a significant time cost. Having access to compressed information, on the other hand, allows one to derive conclusions with the help of simple heuristics, rules, and logic. This simple principle was incorporated in our interpreter model.

When NINSUN has reached a state of confidence about the type of growth pattern relating two variables, it stops its exploration. It does not need to recall all the observations that it made while carrying out the changes. It suffices to recall the dominant type of pattern. The act of reaching such a conclusion (a pattern relating two variables) will henceforth be referred to as (a first form of) *data compression*. For example, for the set of explorations summarized in Table 2, NINSUN compresses the information as is shown in Table 2.

From this table NINSUN knows, for example, that if the area of the container goes down, then the number of collisions with the wall will go up. It does not need to go back to Table 1 and re-infer the conclusions or carry out more modifications of the area.

Table 2. Compressed information derived from Table 1. The independent variable in this simulation is the area of the container.

Dependent variable	Pattern of growth (with area)
Wall collisions/min	Opposite growth
Ball collisions/min	Opposite growth
Pass-through events/min	Opposite growth

Once NINSUN has determined the type of growth pattern that relates two variables, it is ready to try to guess a mathematical formula that describes the relation between those variables, by the application of heuristics. Of course, there is no guarantee that such a mathematical formula can be found; NINSUN is in a state of uncertainty at this point and, in a sense, at the mercy of Dame Fortune.

One heuristic is that if two variables are observed as manifesting opposite growth, then there is a decent chance they have a *constant product*. But this is not the only possibility. In fact, the possibilities are many. If the variables observed to manifest opposite growth are called a and b, and if κ denotes a constant, then the equations $a^2b = \kappa$, $ab^2 = \kappa$, $a + b = \kappa$, $a^3b = \kappa$, and $ae^b = \kappa$ are a few possibilities that illustrate the idea, showing that the possibilities are limitless. A similar explosion of possibilities would happen if a and b are variables that have been observed to manifest *simultaneous* growth: $a = \kappa b$, $a = \kappa b^2$, and so on.

Would a human explorer carry out an exhaustive search through these kinds of possibilities in order to discover a quantitative law to relate the variables? Definitely not, although people are willing to try a few cases. One reason is that this would be a tremendously time-consuming action, especially given that more than two variables are often involved in scientific experiments. The combinatorial explosion would be prohibitive.

In contrast to human behavior, the program BACON had a *system of rules* that considered possibilities until one equation was confirmed (normally, this is the case because the program is fed with data from a law already discovered by humans, such as Boyle's law or Kepler's law.) While this strategy may work to some degree for a brute-force automated system, it is certainly not psychologically plausible. Humans are guided by principles that are far more selective when they guess a quantitative formula. For example, in the discovery of a geometric theorem, conceptual symmetry and analogy play fundamental roles (Lara-Dammer, 2009). There is always a reason *why* a formula holds, and the variables that are involved in the formula can be linked to the observer's perceptual activity.

In order to test, for example, whether $ab = \kappa$, NINSUN's strategy is to select a few specific values of a and b, which it obtained by observation. Having chosen these sample values, it calculates the product ab in each case. If ab is indeed constant, the products ab should have roughly equal values (actually, close to the constant κ, whose value the discoverer is unlikely to know at this stage). Notice that this stage of explicit calculational testing would be reached only if NINSUN had previously observed that a and b are in *opposite growth* in the world-modification stage. This is a way to reduce the potential explosion of possible mathematical formulas that can arise from following a brute-force strategy. If $ab = \kappa$ fails, should NINSUN try $a + b = \kappa$? Not necessarily. It should try this only if the variables a and b have the same units (for example, length), because it makes no sense to add incommensurable variables such as pressure $[\mathrm{Newton/cm^2}]$ and area $[\mathrm{cm^2}]$. A sensible rule-derivation system should not explore mathematical formulas involving the addition of dimensionally incommensurable variables.

This kind of strategy can be characterized as a stage-by-stage approach, in the sense that NINSUN observes events, modifies worlds, compresses information, applies rules, and draws conclusions. Notice that the application of rules and derivation of conclusions come only after the data have been compressed. This stage-by-stage approach is reminiscent of Hofstadter's parallel terraced scan (Hofstadter & Mitchell, 1995), where simpler tests are carried out on a large number of objects, while more complex tests are reserved for the fewer candidates that passed earlier stages of testing.

This type of approach also has the advantage that it allows NINSUN to easily discard conclusions that came from mistaken conjectures of patterns. For example, consider the observation of pass-through events in simulations where NINSUN 'hallucinated' these events. In this case, NINSUN might derive the false conjecture that the smaller the container, the more pass-through events occur. Since no such events actually happened, the actual data turn out to be very scattered (which leads to the conclusion that there is no systematic pattern of growth involving this variable), allowing NINSUN to discard its false conjecture.

Our system's first discovery

In Section 4.3, we explained how NINSUN can arrive at certain conclusions about the world from rough estimates coming from observations made after simple manipulations of the world. These conclusions are essentially statements that under certain conditions, two or more variables in the world might follow a pattern (e.g., opposite growth). We now give a specific case, using data from an actual simulation, to show how an observed pattern led the system to the discovery of a physical law. This example is similar to, but not the same as, the one described in Section 4.3. In that example, NINSUN was manipulating the independent variable 'size of the container,' focusing on its *area*. In this new example, during the simulation, NINSUN manipulates the size of the container, focusing specifically on its *perimeter*.

After modifying the size of the container three times, which allows NINSUN to make four observations, NINSUN concludes that the variables 'perimeter' and 'number of wall collisions' obey a pattern of *opposite growth*. Once NINSUN has settled on this pattern, it then undertakes an exploration of various formulas, using the approximate values provided by its observations. It is important to keep in mind that these values are intended to imitate the qualitative nature of human observations. For example, the value of the variable 'number of wall collisions' that NINSUN obtained during a particular time interval is subject to errors (as can be recalled from the section called 'Relation Identification' in Lara-Dammer et al. (2017)), and is merely an approximation to the actual number of collisions that occurred during this time interval. The same is true for other variables.

For convenience, let us use the letters a and b to represent the dependent variable 'number of wall collisions per minute' and the independent variable 'perimeter of the container,' respectively. If all the products of the form $a_i b_i$ corresponding to the manipulations of the perimeter of the container fall within a certain range close to the average $\sum_{i=1}^{n} a_i b_i / n$ (where n is the number of observations of the independent variable 'perimeter of the container'), then it is very probable that the conclusion $ab = \kappa$ will be drawn. To be more explicit, the hypothesis $ab = \kappa$ will be accepted if, for all $j = 1, 2, \ldots, n$,

$$\left| a_j b_j - \frac{1}{n} \sum_{i=1}^{n} (a_i b_i) \right| < \alpha \frac{1}{n} \sum_{i=1}^{n} (a_i b_i),$$

where α represents a parameter of certainty. Otherwise, when these products are scattered, the hypothesis $ab = \kappa$ is rejected.

In most of the simulations, the value $\alpha = 0.20$ was used. (It seems reasonable that this parameter should have the same value as the parameter of stability δ (see Section 4.2). We have not been able to confirm that this is the case, and therefore we keep two separate parameters, even though we are giving them the same value.) Also, the condition 'all the products of the form $a_i b_i$' could be relaxed slightly and replaced by something like 'the majority of these products.' There are of course pros and cons for any such choice. If we insist on 'all,' quite plausible conclusions may wind up being discarded because of a single discrepant observation. On the other hand, if we use 'the majority,' then some unreasonable conclusions may be drawn. The use of 'all' can be justified to some extent because NINSUN is making only a few observations.

Because NINSUN has a stochastic nature, the output of a simulation, including the conclusions drawn, might differ from run to run. One of the parameters that affect NINSUN's conclusions is the updating time in object–tracking. This is because the observations and interpretations are affected by the distance between two frame positions for a single ball. When using an updating time of 50 milliseconds (the default value), NINSUN arrives at the conclusions given in the following description.

If the radius of every ball is 16 pixels, then the hypothesis $ab = \kappa$ is accepted with a probability slightly higher than 0.5. If the radius is 8 pixels, then the hypothesis $ab = \kappa$ is accepted with a higher probability than in the previous case. If the radius is 4 pixels or smaller, the probability of this conclusion is similar. If the updating time for object–tracking is set to 100 milliseconds instead of 50, the description is similar except that for each of the three radii, NINSUN makes the hypothesis $ab = \kappa$ with a lower probability than with a 50-millisecond update time.

The values observed by NINSUN for the variables 'perimeter' and 'number of wall collisions per minute' in a simulation with balls whose radii are all 8 pixels and with an updating time of 50 milliseconds are shown in Table 3. This table also shows the statistical calculations that NINSUN uses in order to decide whether or not to accept the guess that the product ab is constant. When it accepts the constancy of this product, NINSUN has discovered that the number of wall collisions per minute multiplied by the perimeter of the container is a constant. Expressed as a mathematical formula, NINSUN has discovered that, for small balls,

$$\text{ball} - \text{wall collisions per minute} \times \text{perimeter} = \text{constant}. \quad (1)$$

Or, using more compact notation,

$$ab = \kappa.$$

This is our system's first discovery. Notice that this is not the same as Boyle's law for a two-dimensional gas, which states that the *pressure* multiplied by the *area* of the container is constant. However, there is a definite connection between the two formulas, which will be discussed below.

It is interesting that NINSUN arrived at the conclusion that the number of *ball–ball* collisions per minute and perimeter length are also in opposite growth. However, the last row of Table 3 shows how scattered the values of the variable 'ball–ball collisions/min' were. The precise statistics corresponding to these data are omitted, but the reader can easily see that the values are widely scattered. NINSUN, therefore, rejected the possibility that the product (number of ball–ball collisions times perimeter length) is constant.

In other runs of this sort, NINSUN usually concludes that the number of ball–ball collisions (or the number of pass-through events) per minute and the perimeter length are independent, which is to say, NINSUN does not observe any pattern of growth relating these variables. Only very rarely are wrong conclusions about these variables drawn by NINSUN.

The simulation whose results are shown in Table 3 is recorded in a video available under 'First Discovery 2.' It is worth mentioning that in that simulation, the conclusions were reached in a world having only three balls. It is remarkable that NINSUN's conclusions have a good chance of being correct, considering that reliable patterns and mathematical laws for these situations are far more likely to be seen in worlds with a large number of balls. However, as was mentioned before, there are instances of simulations where the desired conclusion was *not* reached. A simulation where this is the case can be found under 'First Discovery 1.' Increasing the size of the balls raises the likelihood of not reaching the conclusion $ab = \kappa$. This is actually quite reasonable (however, we do not discuss it in any detail), since NINSUN is able to derive a *different* formula

Table 3. Some data and calculations used by NINSUN to conclude that ab is constant. The check mark symbol '$\surd$' is used to indicate that NINSUN checks that the value is less than $\alpha = 0.2$. Notice that $\frac{1}{n}\sum_{i=1}^{n}(a_i b_i) = 143736.95$, $n = 4$.

	obs. 1	obs. 2	obs. 3	obs. 4
$a =$ perimeter length	1256.64	1700.0	1350.0	900.0
$b =$ wall collisions/min	120.0	75.43	96.0	184.8
$a_j b_j$	150796.8	128231.0	129600.0	166320.0
$\left\lvert a_j b_j - \frac{1}{n}\sum_{i=1}^{n}(a_i b_i)\right\rvert / \left\lvert\frac{1}{n}\sum_{i=1}^{n}(a_i b_i)\right\rvert$	0.05 $\surd$	0.10 $\surd$	0.10 $\surd$	0.16 $\surd$
ball–ball collisions/min	24.0	4.8	20.0	54.0

involving the size of the balls, with the formula $ab = \kappa$ being valid only when the radius of the balls approaches zero.

If the simulation is run with five balls instead of three, the probability of accepting the hypothesis $ab = \kappa$ is lower than 0.5, which is what it is when the balls have a radius of 16 pixels. In general, the probabilities of accepting this hypothesis are lower than their corresponding counterparts in the simulations with three balls and different ball radii, described above. Simulations with five balls and various ball sizes are recorded in videos available under 'First Discovery 3/4'. Interpreter scripts to run simulations with scenarios featuring three and five balls and various ball sizes are available with file names 'boylesLawNBallsX.txt,' where 'N' is either 3 or 5 and 'X' is either 'A,' 'B' or 'C.'

Annotated analysis of a run leading to the first discovery

In order to synthesize what has been said about NINSUN's multiple mechanisms that lead to a discovery, we will now show and explain the output of an actual run. Readers should keep in mind that the program's stochastic nature makes it extremely unlikely that it will produce exactly the same output twice. Additionally, the program is evolving and is subject to changes. With those caveats, the output shown in Figure 2 gives a clear idea of how NINSUN behaves.

In line 1, NINSUN chooses to observe and manipulate the independent variable 'perimeter of the container.' Currently, there are fewer than ten independent variables (some of which the interpreter observes during an experiment), and so developing criteria that would lead to a smart selection is not yet important, since we can explore the world modifications produced by all of the independent variables in a reasonable amount of time. For this reason, and for convenience, the user can make the initial choice by selecting an independent variable to be modified by NINSUN.

At the start of a simulation, the world is naturally unstable. It takes several seconds to become stable. This is reflected in lines 2 and 3. After the world becomes stable, NINSUN is ready to observe the variables (dependent and independent) for the first time (line 4). NINSUN then stores the values of the observed variables (lines 5 to 9).

Once NINSUN has taken note of the observed variables, it needs to *modify* the world (line 10). Since the perimeter was the chosen independent variable, it modifies the size of the container. Modifying the size of the container takes some seconds. The world is unstable during this modification, and after the modification a few seconds are needed to reach stability (lines 11 and 12). Once the world becomes stable, NINSUN is ready for a second observation of the variables (line 13). Then NINSUN observes the variables and records their values (lines 14 to 18).

At this point NINSUN has already carried out two sets of observations, and it can now compare them, to see if there are any reliable patterns (line 19). It finds out that some variables increased, others decreased, and some others remained roughly constant (lines 20 to 24). This gives NINSUN some hints about which variables might display which types of growth patterns. In order to look into these hinted possibilities, NINSUN needs to carry out further modifications to the size of the container. It does this several times (lines 25 to 104).

Recall from Section 4.3 that the heuristics helping NINSUN to decide when to stop making modifications are based on the criterion of reaching a majority in the patterns observed for each variable. Sometimes during the modifications of a single variable, these patterns are in conflict. For example, on line 23 the variable 'passthroughs' was observed to exhibit constancy when the perimeter increased (line 20). By contrast, on line 39, it was seen as *increasing* when the perimeter was decreasing (line 36). This type of inconsistency leads NINSUN to carry out a repetition of the modification of the world. (See lines 43 and 59.)

When NINSUN stops modifying the world, it compresses the information according to the patterns that have been found. If a pair of variables is detected to exhibit a known type of pattern of growth, NINSUN can apply a formula-generating rule. For example, the variables 'perimeter' and

```
 1 Chose to manipulate Perimeter
 2 ... Waiting for stabilization
 3 ... Waiting for stabilization (extra time)
 4 ... Observations were (time = 1)
 5    Perimeter: 1256.64
 6    Wall Collisions: 124.0
 7    Ball Collisions: 30.0
 8    Passthroughs: 2.0
 9    Hit Period: 0.01
10 ... Modifying
11 ... Waiting for stabilization
12 ... Waiting for stabilization (extra time)
13 ... Observations were (time = 2)
14    Perimeter: 1700.0
15    Wall Collisions: 100.0
16    Ball Collisions: 22.0
17    Passthroughs: 2.0
18    Hit Period: 0.01
19 ... Checking Monotonicity
20    Perimeter (1256.64 1700.0)  Increasing
21    Wall Collisions (124.0 100.0)  Decreasing
22    Ball Collisions (30.0 22.0)  Decreasing
23    Passthroughs (2.0 2.0)  Approx. Constant
24    Hit Period (0.01 0.01)  Approx. Constant
25 ... Modifying
26 Repeating manipulation to gain more certainty on monotonicity
27 ... Waiting for stabilization
28 ... Waiting for stabilization (extra time)
29 ... Observations were (time = 3)
30    Perimeter: 1350.0
31    Wall Collisions: 132.0
32    Ball Collisions: 33.6
33    Passthroughs: 7.2
34    Hit Period: 0.01
35 ... Checking Monotonicity
36    Perimeter (1700.0 1350.0)  Decreasing
37    Wall Collisions (100.0 132.0)  Increasing
38    Ball Collisions (22.0 33.6)  Increasing
39    Passthroughs (2.0 7.2)  Increasing
40    Hit Period (0.01 0.01)  Approx. Constant
41 ... Modifying
42 Repeating manipulation to gain more certainty on monotonicity
43 Repeating because of observation conflict in Passthroughs
44 ... Waiting for stabilization
45 ... Waiting for stabilization (extra time)
46 ... Observations were (time = 4)
47    Perimeter: 900.0
48    Wall Collisions: 202.29
49    Ball Collisions: 60.0
50    Passthroughs: 5.14
51    Hit Period: 0.0
52 ... Checking Monotonicity
53    Perimeter (1350.0 900.0)  Decreasing
54    Wall Collisions (132.0 202.29)  Increasing
55    Ball Collisions (33.6 60.0)  Increasing
56    Passthroughs (7.2 5.14)  Decreasing
57    Hit Period (0.01 0.0)  Decreasing
58 ... Modifying
59 Repeating because of observation conflict in Passthroughs
60 ... Waiting for stabilization
61 ... Waiting for stabilization (extra time)
62 ... Observations were (time = 5)
63    Perimeter: 1350.0
64    Wall Collisions: 142.5
65    Ball Collisions: 38.0
66    Passthroughs: 2.0
67    Hit Period: 0.01
68 ... Checking Monotonicity
69    Perimeter (900.0 1350.0)  Increasing
70    Wall Collisions (202.29 142.5)  Decreasing
71    Ball Collisions (60.0 38.0)  Decreasing
72    Passthroughs (5.14 2.0)  Decreasing
73    Hit Period (0.0 0.01)  Increasing
74 ... Modifying
75 Repeating because of observation conflict in Hit Period
76 ... Waiting for stabilization
77 ... Waiting for stabilization (extra time)
78 ... Observations were (time = 6)
79    Perimeter: 1700.0
80    Wall Collisions: 120.0
81    Ball Collisions: 16.8
82    Passthroughs: 0.0
83    Hit Period: 0.01
84 ... Checking Monotonicity
85    Perimeter (1350.0 1700.0)  Increasing
86    Wall Collisions (142.5 120.0)  Approx. Constant
87    Ball Collisions (38.0 16.8)  Decreasing
88    Passthroughs (2.0 0.0)  Decreasing
89    Hit Period (0.01 0.01)  Approx. Constant
90 ... Modifying
91 ... Waiting for stabilization
92 ... Waiting for stabilization (extra time)
93 ... Observations were (time = 7)
94    Perimeter: 1350.0
95    Wall Collisions: 144.0
96    Ball Collisions: 30.0
97    Passthroughs: 0.0
98    Hit Period: 0.01
99 ... Checking Monotonicity
100    Perimeter (1700.0 1350.0)  Decreasing
101    Wall Collisions (120.0 144.0)  Approx. Constant
102    Ball Collisions (16.8 30.0)  Increasing
103    Passthroughs (0.0 0.0)  Approx. Constant
104    Hit Period (0.01 0.01)  Approx. Constant
105
106 ... Summarizing previous observations
107
108  .... Perimeter and Wall Collisions appear to be one
growing and the other shrinking
109
110    ... Confirming previous observations with rough
measures and calculations of Perimeter and Wall Collisions
111 Wall Collisions and Perimeter REALLY seem to be
inverse proportional which means that
112 numberOfWallCollisions * perimeter = constant
113  .... Perimeter and Ball Collisions appear to be one
growing and the other shrinking
114
115    ... Confirming previous observations with rough
measures and calculations of Perimeter
    and Ball Collisions
116 Ball Collisions and Perimeter FAIL to be
inverse proportional
117
118  .... Perimeter and Passthroughs appear to be one
growing and the other shrinking
119
120    ... Confirming previous observations with rough
measures and calculations of Perimeter
    and Passthroughs
121 Passthroughs and Perimeter FAIL to be
inverse proportional
122
123  .... Perimeter and Hit Period appear to be both
growing or both shrinking
124
125    ... Confirming previous observations with rough
measures and calculations of Perimeter
    and Hit Period
126 Hit Period and Perimeter REALLY seem to be direct
proportional which means that
127 wallHitPeriod / perimeter = constant
128 Finishing the simulation
```

Figure 2. Sample output from NINSUN.

'wall collisions' are found by NINSUN to exhibit *opposite growth* (line 108). At this point, it applies the rule that allows it to guess that the *number of collisions* multiplied by *perimeter length* might be a constant. However, it is not yet convinced. It needs to use its rough measurements and calculate the actual products to see if they appear to be constant, or close to a constant. (This stage of performing the multiplications and displaying the products does not appear in the output.) Since all the products turned out to be close enough to each other, NINSUN is now convinced that the number of collisions per minute multiplied by the perimeter length is indeed a constant. This conclusion is shown in lines 111 and 112.

The other variables that initially appeared to exhibit a pattern of growth could not be used to derive a formula, because their calculated products turned out to be too scattered (lines 116 and 121).

The variable Hit Period represents the time between two consecutive collisions. This variable is the reciprocal of the variable 'wall collisions (per minute),' and the conclusions involving Hit Period are very similar (lines 126 and 127) to the ones involving the number of collisions per minute. Finally, in line 128, NINSUN decides to stop the simulation. Theoretically, it can continue the simulation, but it stopped for convenience.

More discoveries and current limitations

We have described NINSUN's first discovery that the number of ball-wall collisions multiplied by the perimeter of the container is a constant. In spite of its simple appearance, this example has served its purpose of showing some fundamental and intricate conditions that need to be met for the realization of a discovery.

A second but related discovery that NINSUN can make is a 2-D version of Boyle's law, which states that the pressure multiplied by the area of the container is a constant. At first sight, the second discovery looks like a slight variant of the first. However, moving from the first discovery to the second one required a significant amount of work because it is necessary to understand how the concept of pressure relates to that of balls colliding against a container. Likewise, we developed NINSUN so that it can make a number of other related discoveries, such as Gay-Lussac's law of pressure-temperature, which states that when the mass and volume of a gas are constant, the pressure of the gas is directly proportional to its absolute temperature.

It sounds almost trivial, but these discoveries were the result of a significant amount of calculation in NINSUN, involving the creation of perceptually-grounded notions such as molecule direction, velocity, collision, and pressure. At the time of writing of this paper, the main focus of attention in our model of discovery was the conjecturing of Boyle's law and some variations. The most natural extension of this work was to enhance the interpreter so that it could make other discoveries about the laws obeyed by gases. In the process, however, it became clear that having NINSUN make other discoveries would require large amounts of new programming both in the interpreter and in Tricycle. Because this was such a big task, the current software is not yet able to run certain simulations, even though they were fully planned out and partially implemented. Some of NINSUN's discoveries have actually been made, some others would be made under certain conditions, and yet others remain as future challenges. We arrived at the conclusion that these other discoveries should be described in a separate paper, and we left only the first discovery in this paper. In order to address the future challenges, we will need to add fundamental new components, such as sorting balls into groups defined by various properties. Grouping has not reached a solid foundation at the moment of writing this paper.

Currently, and in part because of the small number of discoveries that NINSUN can handle, its interventions in the world are basically random. This limitation should be corrected, in order to allow NINSUN to deal with larger microworlds with a larger number of attributes in a meaningful way. The randomness should be replaced by a system capable of selecting actions in the manner of Copycat (Mitchell, 1990). Such a system would allow NINSUN to select an activity based on its observations, in much the same way as Copycat can find a plausible solution based on fundamental processes of cognition such as grouping objects into clusters and analogy-making.

For instance, a human develops a mental model in which gas molecules are like billiard balls careening off of walls and each other. This might arouse the humans interest in factors like molecule size, which would influence the probability that balls would hit each other. If molecules

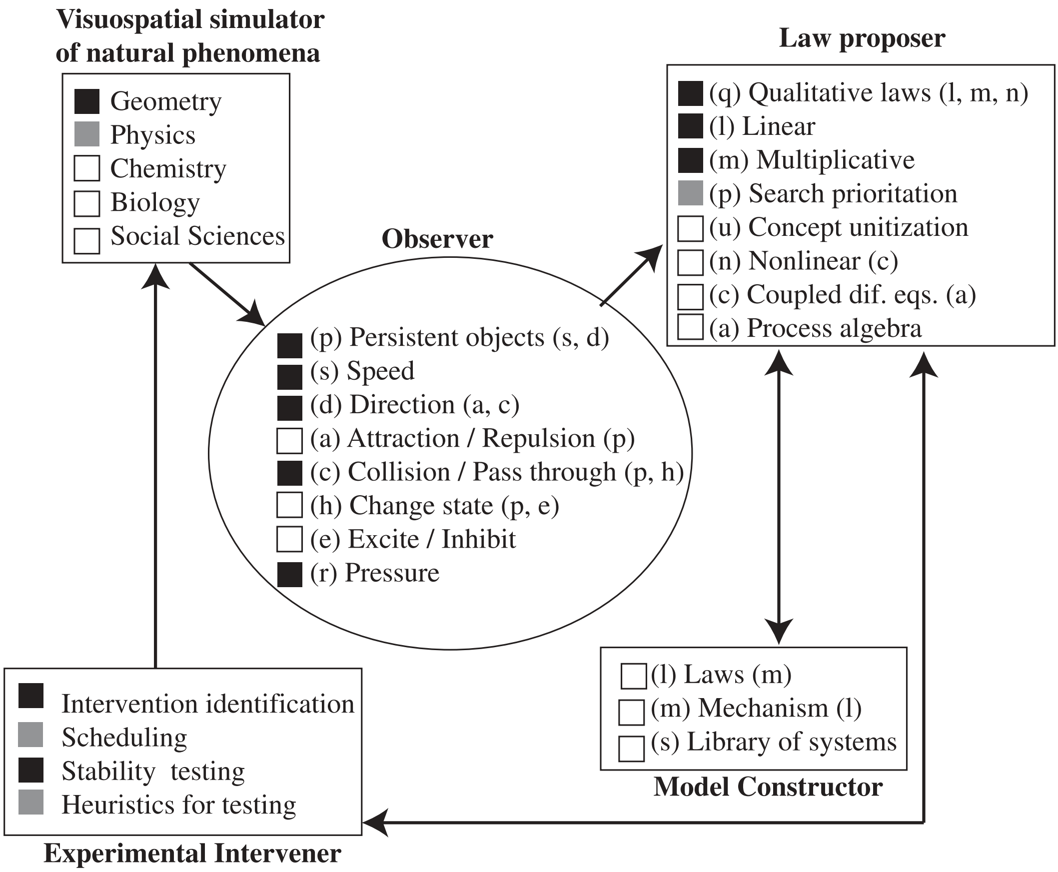

Figure 3. Schematic representation of some of the components that were originally planned for our project and the state they are at the moment of writing this paper. A black square indicates what is implemented, a gray square what is partially implemented, and a white square what has not yet been started. The arrows indicate the flow of information from one component to another. An attribute in the same component category is indicated by a letter on its left. The letters to the right correspond to attributes that depend on the calculation of the attribute. For example, the speed (s) of an object cannot be determined until its persistence (p) across time has been established.

never hit each other, for example, people might become interested in whether that fact increases, decreases, or leaves unchanged, the frequency of molecule-to-wall collisions.

We emphasize that we do not consider our model to be a complete product or even close to it. What is described in this paper is just a small part of a larger project. The diagram in Figure 3 indicates a sketch of this project, pointing out what is implemented (black squares), what is partially implemented (gray squares), and what has not yet been started (white squares). The diagram shows, for example, that model-building and model-testing (where most of the action might come in proposing motivated interventions) is yet to be realized. Additionally, the diagram indicates causal correlations between attributes within the same component, reflecting our earlier comment that the perception of causality in our system is akin to Michotte's work (Michotte, 1963).

Conclusion

We have simulated scientific discovery utilizing a very simple world. This is an artificial microworld consisting of fixed walls and moving balls. In spite of its simplicity, this humble microworld is capable of posing many serious challenges to a human observer who is willing to observe regularities and to make discoveries, or even to conjecture mathematical laws. This is because the collisions that the balls make (with the walls and with one another) have a high degree of randomness to a human observer.

Much like scientists who do experiments in the world that we inhabit our simulated system emulates the two inseparable elements of discovery – observer and microworld – and allows the observer to explore the microworld and to conduct experiments in it. Some events in this microworld, where context plays a significant role, are subject to interpretation, and this significantly increases the difficulty of making discoveries. Yet discovery is possible.

We describe our system NINSUN, which is a computational model of human scientific discovery that takes into account the complex process of human perception. Our system emulates the human visual system, and as such it

(1) observes the microworld,
(2) tracks objects in it,
(3) identifies relations between them (such as collisions),
(4) makes modifications of the world,
(5) memorizes events,
(6) compresses information,
(7) detects patterns among the objects, and last but not least,
(8) makes conjectures of mathematical laws.

Our simulation of these different stages of mental activity is based on selected cognitive theories of perception and scientific discovery. In this paper we address items (4), (5), (6), (7), and (8). (The first three were addressed in our earlier paper Lara-Dammer et al. (2017).)

Even though this microworld contains only a few balls, and is two-dimensional, it is a reasonable analogue to what happens in the real world and specifically to aggregates of gas molecules. In spite of the intrinsic uncertainty and the inevitable error-making of the interpreter, patterns of perception can be derived, and ultimately, the patterns lead to the formulation of mathematical equations – which is to say, scientific hypotheses.

Acknowledgments

The majority of this research was carried out at Indiana University, mostly in the Percepts and Concepts Laboratory of the Department of Psychological and Brain Sciences, but also partly at the Center for Research on Concepts and Cognition (CRCC). The most recent phases of the project were carried out at the Escuela Politécnica Nacional in Quito, Ecuador. Roughly seventy percent of the funding for this project came from National Science Foundation REESE grant 0910218, twenty percent from CRCC, and the remaining ten percent from the Escuela Politécnica Nacional.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation [REESE grant 0910218].

References

Ayazoglu, M., Yilmaz, B., Sznaier, M., & Camps, O. (2013). Finding causal interactions in video sequences. In *Computer Vision (ICCV), IEEE International Conference,* Sydney, NSW, Australia. (pp. 3575–3582).

Chalmers, D., French, R., & Hofstadter, D. (1995). High-level perception, representation, and analogy: A critique of artificial-intelligence methodology. In: Douglas Hofstadter and the Fluid Analogies Research Group, *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. New York: Basic Books.

Dehaene, S. (1997). *The number sense. how the mind creates mathematics*. New York: Oxford University Press.

Eves, H. (1969). *An introduction to the history of mathematics* (3rd ed.). New York: Holt, Rinehart and Winston.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, *24*, 85–168.

Hilbert, D. (1999). *Foundations of geometry* (2nd English ed. ed.). La Salle, Illinois: Open Court Classics.

Hofstadter, D., & Mitchell, M. (1995). The copycat project: A model of mental fluidity and analogy-making. In: Douglas Hofstadter and the Fluid Analogies Research Group, *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. New York: Basic Books.

Kasparov, G. (2010). The chess master and the computer. *The New York Review of Books*. http://www.nybooks.com/articles/archives/2010/feb/11/the-chess-master-and-the-computer/

Klahr, D. (2000). *Exploring science: The cognition and development of discovery processes*. Cambridge: MIT Press.

Klahr, D., & Dunbar, K. (1987). *Dual space search during scientific reasoning*. Tech. Rep. No. AIP-13. Pittsburgh, PA: Carnegie-Mellon University.

Kline, M. (1972). *Mathematical thought from ancient to modern times*. New York: Oxford University Press.

Lakoff, G., & Núñez, R. (2000). *Where mathematics comes from. How the embodied mind brings mathematics into being*. New York: Basic Books.

Langley, P., Simon, H., Bradshaw, G., & Zytkow, M. (1987). *Scientific discovery. Computational explorations of the creative processes*. Cambridge: The MIT Press.

Lara-Dammer, F. (2009). *Modeling human discoverativity in geometry* (Unpublished doctoral dissertation). Indiana University.

Lara-Dammer, F., Hofstadter, D., & Goldstone, R. (2017). A computer model of context- dependent perception in a very simple world. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*, 1247–1282.

Lenat, D. (1979). On automated scientific theory formation. A case study using the AM program. In J. Hayes, D. Michie, & Mikulich (Eds.), *Machine intelligence* (pp. 251–283). Chichester, U.K.: Hellis Horwood.

Linhares, A., & Brum, P. (2007). Understanding our understanding of strategic scenarios: What role do chunks play? *Cognitive Science*, 31(6), 989–1007.

Michotte, A. (1963). *The perception of causality*. London: Methuen & Co Ltd.

Mitchell, M. (1990). *Copycat: A computer model of high-level perception and conceptual slippage in analogy-making* (Unpublished doctoral dissertation). University of Michigan.

Weinberg, S. (2015). *To explain the world*. New York: HarperCollins Publishers.