*Information Modelling and Knowledge Bases XXXV*
*M. Tropmann-Frick et al. (Eds.)*
*© 2024 The Authors.*
*This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).*
*doi:10.3233/FAIA231161*

# A Knowledge Model Based on "Dark-Matter" and Parallel Spaces

Xing Chen[1] and Yasushi Kiyoki[2]
[1] *Department of Information & Computer Sciences*
*Kanagawa Institute of Technology, Japan*
[2] *Graduate School of Media and Governance, Keio University, Japan*

**Abstract.** This paper aims to analyze the phenomenon of inapplicability of experience, which means that sometimes we make mistakes when we use our past experience to solve current problems. We propose a knowledge model based on the concept of "dark-matter", which is a term used to describe the time-related data that is hidden from our observation. We use a two-dimensional matrix to represent both time-related and non-time-related data, and we call it space. We also introduce the concept of parallel spaces, which are composed of several spaces that can explain different situations and outcomes. We use case studies to illustrate how knowledge is generated and expressed using "dark-matter" and parallel spaces. We also reveal the reason for the inapplicability of experience and suggest some solutions. The contribution of this paper is that we provide a new perspective and a new model to understand and process knowledge based on "dark-matter" and parallel spaces.

**Keywords.** Knowledge, knowledge presentation, knowledge generation, machine learning, semantic space, spatiotemporal space

## 1. Introduction

People make judgments and decisions based on experience and knowledge to solve problems. However, it often happens that wrong judgments and decisions are made based on experience and knowledge. For example, in the stock trading process, people will determine the current trade based on the stock's past ups and downs and trading experience. However, decisions based on experience are not always correct. After you decide to buy, the stock may fall, or after you decide to sell, the stock may rise. This phenomenon is referred to as "the phenomenon of inapplicability of experiences" in this paper. The aim of this paper is to create a knowledge model to analyze this phenomenon. It is known that knowledge and experience build up over time. In other words, if we want to express knowledge and experience in computers, we need to create a time related knowledge model. In our previous research, we proposed a knowledge model based on a concept of "dark matter" [1].

The concept of "dark-matter" is developed based on the research works of semantic computing models [2, 3, 4, 5]. In the semantic computing models, matrixes are used to present "meaning" of data. In the models, input data are mapped through mapping matrixes into semantic spaces and presented as points in semantic spaces. Another data

[1] Xing Chen, 1030 Simo-Ogino, Atsugi-shi, Kanagawa 243-0292, Japan; chen@ic.kanagawa-it.ac.jp
[2] Graduate School of Media and Governance, Keio University, Japan

which present different meanings, referred to as meaning data, are also mapped to the same space. Distances of the points among input data and meaning data are performed. In this way, the semantic calculation is transmitted to calculate Euclidean distances of those points. For example, in the case for implementing semantic query, query data presenting query keywords are mapped into a semantic space and summarized as a point in the space. Retrieval candidate data are also mapped into the semantic space as other points. Euclidean distance is calculated between the query point and each retrieval candidate point. When the distance of a retrieval candidate is shorter than a given threshold, its relative retrieval candidate is extracted as the query output.

Two methods, Mathematical Model of Meaning (MMM) [4, 5] and Semantic Feature Extracting Model (SFEM) [2, 3], are developed on semantic space creation. In MMM, a common data set, for example an English-English diction is used to create the semantic space. In SFEM, the semantic space is created based on special defined data sets according to the requirement of applications.

After the semantic space is created, input data will be mapped to the space and the Euclidean distance calculation between the mapped data points in the space will be performed. Mapping matrixes are required to map input data into the semantic space. In SFEM, mapping matrixes are defined according to the applications of the models [6- 14]. Many methods are developed to create mapping matrixes for applying the model in the areas of semantic information retrieving [9, 10, 14], semantic information classifying [11], semantic information extracting [12], and semantic information analyzing on reason and results [13], etc. Furtherly a method is developed to create the mapping matrixes through deep-learning [15]. Elements of the matrixes presenting semantic spaces are previously defined, which are different from those other models, such as the artificial neural network model and deep-learning artificial neural network model [16-19].

A mechanism is furtherly developed based on the semantic space model to implement the basic logic computation to implement true and false judgement which is the foundational mechanism required by machine learning [20]. The mechanism is applied to simulate unmanned ground vehicle control [21]. Temporal data processing is required for the ground vehicle control. In paper [1], a model is presented for the temporal data processing. In the model, the word “matter” is used to represent features of spaces which are related to the non-temporal data. The word “dark-matter” is used to represent of spaces which elements are temporally changed and “dark-matter” is also referred to as a matrix which elements are generated during training processing. In the paper [22], an exploratory research was presented on the expression of knowledge and its generation process based on the concept “dark-matter.”

In this paper, a new knowledge model is created to analyze “the phenomenon of inapplicability of experiences” and a new concept of “parallel spaces” is presented based on the previous researches. Case studies are used to illustrate why “the phenomenon of inapplicability” occurred. The concept of “dark-matter” is used in this research. It is reviewed in Section 2. In Section 3, the relationship between “knowledge” and “dark-matter” is described. The phenomenon of inapplicability is also analyzed. with examples in this section. Then, the concept of “parallel spaces” is presented and illustrated with examples. Finally, the conclusions of this paper are presented.

## 2. The machine learning model created based on the concept "dark-matter"

In this section, the machine learning model created based on the concept "dark-matter" is briefly reviewed through an example as shown in Figure 1. This model is called "dark-matter learning model" in the following. In this model, matrix multiply is referred to as "space mapping" or simply "mapping." For example, the following equation (1) is described as to mapping a matrix **X** by a matrix **C** to a new matrix **E**.

$$\mathbf{E} = \mathbf{X}*\mathbf{C} \tag{1}$$

The matrix **X** is further divided into two parts as shown in Figure 1. The first part is the first column of the matrix **X,** which is referred to as "***matter***". The second part of **X** is referred to as the "***dark-matter***", which is constructed by the second to the last column of the matrix **X**. In Figure 1, "***matter***" is an index matrix correlated with sensor data. As sensor data are "visible", it is referred to as matter. Elements in the matrix of "***dark-matter***" are randomly filled. As the matrix of the dark-matter is created randomly filled, it is referred to as "chaotic space".

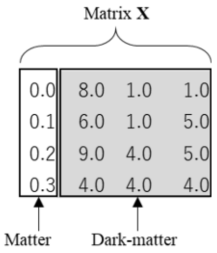


**Figure 1**. Creating a "chaotic space"

As pointed in the paper [22], the expression of knowledge and the knowledge generation process are correlated with "dark-matter." A machine learning model is also presented based on the concept "dark-matter" [22]. This model is referred to as the "dark-matter learning model."

The learning process is to change the "chaotic space" to "ordered space." The learning model is considered as a state machine. A state in the state machine will be changed from one to others, which is referred to as state transition. State transition diagrams are used to present the state transition. Figure 2 is an example of state transition diagram. In the figure, a circle and the number in the circle represents a state, an arrow represents a state transiting from one to another one. The numbers by the arrows represent the condition required for the state transition. For example, the state will be transited from state "0.0" to the state "0.1" if the input data is "0.0".

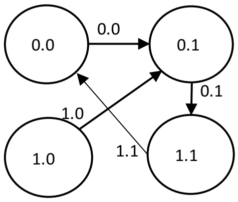


**Figure 2**. An example of state transition

Figure 3 presents how to create the "ordered space" from the "chaotic space" based on the state transition. The example of the state transition diagram shown in Figure 2 is used to illustrate it. In Figure 2, state "0.0" is the start state and the state transition from "0.0" to "0.1" is the first step of the state transition. Figure 3 (a) shows a chaotic space and Figure 3 (b) shows the first step state transition. In the second step, the state "0.1" transits to the state "0.3". as shown in Figure 3 (c). When all the state transition diagram shown is represented as Figure 3 (d), we say that we created an "ordered space" from the "chaotic space." In the figure, the "matter" is the first row of the matrix, and the "dark-matter" is the matrix constructed by the second, third and fourth row, painted in gray.

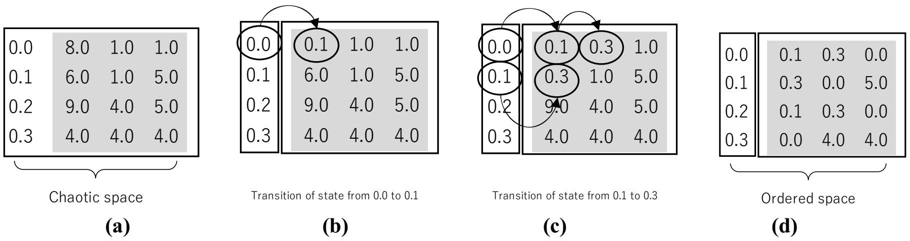


**Figure 3**. Creating an "ordered space" from a "chaotic space"

If $\mathbf{X}^{-1}$ is an inverse matrix of the matrix **X**, the matrix $\mathbf{X}^{-1}$ is referred to as an "***antimatter space***." By applying antimatter space to a matrix **E**, which represents actions of agents, a new matrix **C** is created as shown in equation (2). The calculation presented by the equation (2) is referred to as "*learning*" or "*training*" calculation.

$$\mathbf{C} = \mathbf{X}^{-1} * \mathbf{E} \qquad (2)$$

In the equation "mass and energy equivalent," "**mass**" is a measure of the amount of matter that an object contains. That is, it is visible, or it can be sensed by sensors. As explained in Figure 1, "**matter**" is defined as a vector that correlates with sensor data. That is, "**matter**" can be considered the same as "**mass**." It can also be seen that the concept of energy in physics is a vector **E**. In equation (1), the values of the elements of the vector **E** and the mass in the matrix **X** are measurable. That is, the measurable values of **E** are the energy in physics and the measurable values of mass in physics is the "**matter**" that we referred to.

Next, we will take an example to analyze the relationship between equation (1) and the "energy mass equivalent" equation. Suppose that space X is a five-by-five matrix, as shown in Figure 4 (a). This assumption means that there are five different types of matter in space. The elements in the first column of the matrix are the mass of matter. In the example, the values of the "**matter**" and "**dark-matter**", which are the elements of the matrix **X**, are shown in Figure 4 (a). The elements of the vector **E** are shown in Figure 4 (b). The $\mathbf{X}^{-1}$, which is the inverse matrix of **X**, is show in Figure 4 (c).

$$\begin{array}{ccccc} \begin{pmatrix} 6.00 & 14.16 & 72.84 & 49.42 & 26.72 \\ 3.00 & 83.37 & 75.00 & 36.21 & 28.21 \\ 9.00 & 27.55 & 32.70 & 59.96 & 20.56 \\ 2.00 & 28.83 & 34.27 & 35.39 & 74.98 \\ 7.00 & 55.43 & 86.68 & 38.21 & 87.91 \end{pmatrix} & \quad & \begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix} & \quad & \begin{pmatrix} -0.09 & -0.07 & 0.11 & -0.15 & 0.16 \\ -0.02 & 0.01 & 0.01 & 0.00 & 0.00 \\ 0.02 & 0.00 & -0.01 & -0.01 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.02 & -0.03 \\ -0.01 & -0.01 & 0.00 & 0.01 & 0.01 \end{pmatrix} \\ \text{mass} \qquad \text{dark-matter} & & \mathbf{E} & & \mathbf{X}^{-1} \end{array}$$

**(a) Space X: created with five types of matter** **(b) Energy vector** **(c) Inverse matrix of X**

**Figure 4**. An example of a space with five different types of matter

The values of the element of the vector **C** can be calculated based on equation (2), multiplying $\mathbf{X}^{-1}$ by **E**, as shown in Figure 5. Therefore, we call vector **C** "*rule*" and vector **E** "*energy*".

$$\underset{\mathbf{C}}{\begin{pmatrix} 100.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}} = \underset{\mathbf{X}^{-1}}{\begin{pmatrix} -0.09 & -0.07 & 0.11 & -0.15 & 0.16 \\ -0.02 & 0.01 & 0.01 & 0.00 & 0.00 \\ 0.02 & 0.00 & -0.01 & -0.01 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.02 & -0.03 \\ -0.01 & -0.01 & 0.00 & 0.01 & 0.01 \end{pmatrix}} * \underset{\mathbf{E}}{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}}$$

**Figure 5**. Multiplying $\mathbf{X}^{-1}$ by **E** to calculate the rule vector **C**

As shown in Figure 5, only the first element of vector **C** is a non-zero value, and all the other values of the vector are zero. When the vector **C** is given, the vector **E** can be calculated based on equation (1), as shown in Figure 6 (a). Let $c^2$ represent the first value of **C**, the vector **E** can be calculated by multiplying the first column of **X** by $c^2$, as shown in Figure 6 (b).

$$\underset{\mathbf{E}}{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}} = \underset{\text{mass} \qquad \text{dark-matter}}{\begin{pmatrix} 6.00 & 14.16 & 72.84 & 49.42 & 26.72 \\ 3.00 & 83.37 & 75.00 & 36.21 & 28.21 \\ 9.00 & 27.55 & 32.70 & 59.96 & 20.56 \\ 2.00 & 28.83 & 34.27 & 35.39 & 74.98 \\ 7.00 & 55.43 & 86.68 & 38.21 & 87.91 \end{pmatrix}} * \underset{\mathbf{C}}{\begin{pmatrix} 100.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}} \qquad \underset{\mathbf{E}}{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}} = \underset{\mathbf{mass}}{\begin{pmatrix} 6.00 \\ 3.00 \\ 9.00 \\ 2.00 \\ 7.00 \end{pmatrix}} * \underset{c^2}{100.00}$$

**(a) Calculating E with vector $\mathbf{C}^2$** **(b) Calculating E with the scalar value $\mathbf{c}^2$**

**Figure 6**. The calculated result of **E** with the vector $\mathbf{C}^2$ and the scalar value $c^2$

In this example, if *e* is the *i*-th element of the vector **E** and *m* is also the *i*-th element of the vector **mass**, it can be found that $e = mc^2$. For example, for the first element of E and the first element of mass,

$E = 600;$
$m = 6;$
$c^2 = 100$, where, $c = 10;$
$600 = 6 * 10^2.$

That is,

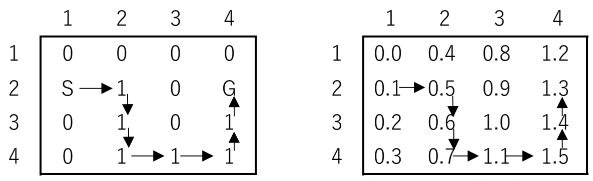
$$e = mc^2. \qquad (3)$$

In summary, if the vector **C** is defined such that the value of its first element is a non-zero value $c^2$ and all the other values are zero, as shown in Figure 6, the elements values of the vector **E** can be calculated by multiplying the vector **mass** by the scalar value $c^2$. That is, $e = mc^2$. The vector **C** is a special vector indicating that “dark-matter” is not used in the calculation. That is, the equation, $e = mc^2$, is suitable to the case that “knowledge” or “state transition” is not considered during the computation process.

## 3. A knowledge representation method with “dark-matter”

The relationship between “**dark-matter**” and “knowledge” is represented in the paper [22]. Let’s use an example to illustrate it. Suppose that there is a maze, represented by a 4 x 4 matrix, as shown in Figure 7 (a). An agent is created in this case study which should move from a start position to a goal position. The start position of the agent is at row 2 and column 1, which is marked with the character “S”. A position will be represented by row and column number as (*row number, column number*) in the following. Thus, the start position of the agent is represented as (2, 1) with the mark “S”. The goal position of the agent is (2, 4) with the mark “G”. The agent will go along the path shown by the arrow-mark “→”. From the start position to the goal position. That is, the agent goes through the points (2,1), (2, 2), (3, 2), (4, 2), (4, 3), (4, 4) and (3, 4) and reaches to (2, 4).

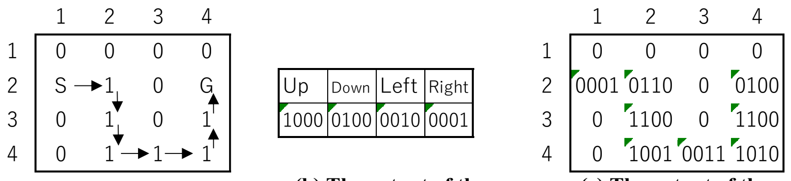
By defining states of the agent as its positions on the maze matrix, a space matrix can be defined. As there are 16 possible positions for the agent on the maze matrix, 16 values from 0.0 to 1.5 are used to represent the index as shown in Figure 7 (b).



**(a) A maze matrix** **(b) The position index values of the maze matrix**

**Figure 7**. A maze matrix shown for an agent moving from “S” to “G”

However, it is not always possible to calculate the action index value without the dark-matter. If the agent does not have the position sensor, but it has a laser sensor which output shows which direction that the agent can move as shown in Figure 8. The output values of the sensor are defined as follows: As shown in Figure 8 (b), the agent can move in four directions, moving to the “Down”, “Left” and “Right” direction, the outputs of the sensor are defined “0100”, “0010” and “0001”, respectively. The direction, in which the agent can move at each position, is shown in Figure 8 (a). The sensor's output values when the agent is in different positions are shown in Figure 8 (c). For example, at the position (3, 2), the agent can move to “Up” direction, therefore, the output value of the sensor is “1000”. The agent can go back from the current position to its previous position. If the agent is at the point (2, 2), the agent can move to two different positions. The agent can move back to the start position (2, 1) when it move “Left”. It can also move “Down” to the next position (3, 2). The output value of the sensor at the position (2, 2) is “0110”, which is the summary of the two direction “0010”, “Left” and “0100”, “Down”.



(a) The maze matrix

| Up | Down | Left | Right |
|---|---|---|---|
| 1000 | 0100 | 0010 | 0001 |

(b) The output of the laser sensor

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0001 | 0110 | 0 | 0100 |
| 3 | 0 | 1100 | 0 | 1100 |
| 4 | 0 | 1001 | 0011 | 1010 |

(c) The output of the laser sensor in the maze

**Figure 8**. The output of the laser sensor of the agent

Figure 8 (c) shows a situation where the agent has the same sensor output of "1100" at two different positions: (3, 2) and (3, 4). This means that the agent can either move up or down from either position. However, if we want to define a function, f(x), that takes the sensor output as input and gives the agent’s action as output, we will have a problem. The same input of "1100" will have two possible outputs: "Moving Down" or "Moving Up". This is not a valid function, because a function should have only one output for each input. Therefore, we need to use some additional information, called dark-matter, to help the agent decide which action to take.

It is not always possible to know the path at the start position as that shown in Figure 6 (b). For example, when reinforcement learning is applied to find the path from the start position to the goal position, the path is unknown at the start position. The path will be found after many trys are performed. Sometimes, many paths are found. Rewards are assigned to found paths. If the length of a path is shorter than the others, higher reward than the others is assigned to the path. During the reinforcement learning, the agent is trained to obtain the maximum reward. In this way, the optimal path from the start position to the goal position can be found. At the same time, actions of the agent at the positions on the path are determined.

When the path is unknown at the start position, it is impossible to create the space matrix as that shown in Figure 8 (a). Here, ***a new method is proposed for generating the space matrix***. At the same time, ***a new method for knowledge expression is also proposed***. In the methods, passed positions are recorded instead of the next positions. For example, when the agent moves from the start position (2, 1) to the position (2, 2), the passed position (2, 1) is recorded. That is, dark-matter matrix can be created based on events that have occurred instead of the events which will occurred in the future. In the following, an example is used to illustrate the method in detail. In the example, the agent has a laser sensor.

In the example, a matrix is used to show a maze as shown in Figure 9 (a). The agent can be in any positions where the “1” is marked in the matrix. The agent cannot be in the positions where “0” is marked in the matrix. Figure 9 (b) shows the output of the sensor of the agent in each position, and Figure 9 (c) shows the index values of the sensor outputs at each position. For example, when the agent is at the position (2, 2), as the agent can move to “Up”, “Down” and “Left”, output of the sensor is “1110”, which is the summary of the value “Up”, “1000”, “Down”, “0100” and “Left”, “0010”. As the agent can not be in the position (3, 1), (4, 1), (2, 3), (3, 3) and (1, 4), the output of the sensor at those positions is marked with an “X”.

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 |
| 4 | 0 | 1 | 1 | 1 |

**(a) The maze matrix**

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0101 | 0111 | 0111 | X |
| 2 | 1001 | 1110 | X | 0100 |
| 3 | X | 1100 | X | 1100 |
| 4 | X | 1001 | 0011 | 1010 |

**(b) The output of the laser sensor**

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.5 | 0.7 | 0.7 | X |
| 2 | 0.9 | 1.4 | X | 0.4 |
| 3 | X | 1.2 | X | 1.2 |
| 4 | X | 0.9 | 0.3 | 1.0 |

**(c) The index of the laser sensor**

**Figure 9**. The output of the laser sensor of the agent

The reinforcement learning is performed as the follows. At the position (2, 1), the agent can move to two different directions “Up” and “Right”. The probability value 0.5 is assigned to the two directions. That is, the agent has a 50% chance of moving “Up” and a 50% chance of moving to the “Right”. When the agent moves to the position (2, 2), it has three moving directions, “Up”, “Down” and “Left”. Therefore, 0.33 is assigned as the probability value to the three directions. The maze is set up as follows. The target position is set at position (2, 4), and the starting position can be any position marked “1", as shown in Figure 9 (a). When the agent moves from a position to its neighbor position, it is said to have "moved a step". The number of the steps, which the agent is used to move from the start position to the goal position, is used for the reward calculation. The fewer steps the agent used, the higher the score the agent got. If the start position is at the point (2, 1), the minimum number of steps required is seven for the agent moving from the start position the goal position. When the agent moves to a direction where higher reward were obtained than the other directions, higher probability value is assigned to the direction. For example, at the position (2, 1), as moving to the “Right” can obtain higher reward than moving “Up”, 0.01 is added to the probability value of moving to that “Right”. Experiments were performed to find the path from the start position to the goal position. Based on the experimental results, trying about 200 times, the best path from the start position (2, 1) to the goal position (2, 4) can be found by the agent. After 2000 times trying, the probability values to the high reward directions are increased to 0.99.

To record sensor values, ***a working memory mechanism is utilized***. In the mechanism, a vector is created which initial value is randomly assigned. The length of the vector is as same as the number of the steps required by the agent from the start position to the goal position. For example, when the agent started at position (2, 1) which is set as the start position, and the agent moved through positions, (2, 2), (3, 2), (4, 2), (4, 3), (4, 4), (3, 4), and reached the goal position (2, 4), there are seven steps are required. Thus, the vector is defined with seven elements with random values. Then, the value of the current sensor value is combined with the vector, thus a vector with eight elements is created, as shown in Figure 10. In Figure 10, index values of the sensor values are used. The background of the element recording the current sensor value is painted white, and the background of the elements recording the past sensor values are painted gray. For example, at the start position (2, 1), the current sensor value is 0.9, which was recorded at the first element of the vector, and the background of the first element was painted white, as shown in Figure 10 (a). The background of all the other elements were painted gray. The values of the elements from the second to the eighth were remained the random values because no sensor data were recorded. When the agent moved to position (2, 2), 1.4 which was the index value of the sensor at the position, was recorded to the first element and the previously recorded data 0.9 was moved to the second element as shown in Figure 10 (b). In the same way, when the agent moved to position (3, 2), 1.2, which was the index value of the sensor output, was recorded to the first element, and the previously recorded data were moved to the second and third elements, respectively, as shown in Figure 10 (c). After the agent moved to the goal position (2, 4), all index values of the sensor output were recorded as shown in Figure 10 (d). The index value of the sensor output at the goal position was recorded to the first element, and the index value of the sensor at the start position was recorded to the eighth element.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| **0.9** | **94.0** | **84.0** | **12.0** | **78.0** | **58.0** | **84.0** | **23.0** |

**(a)** At position (2,1)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| **1.4** | **0.9** | **84.0** | **12.0** | **78.0** | **58.0** | **84.0** | **23.0** |

**(b)** At position (2,2)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| **1.2** | **1.4** | **0.9** | **12.0** | **78.0** | **58.0** | **84.0** | **23.0** |

**(c)** At position (3,2)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| **0.4** | **1.2** | **1.0** | **0.3** | **0.9** | **1.2** | **1.4** | **0.9** |

**(d)** At position (2,4)

**Figure 10**. Working-memory and recorded index value of the sensor output

Actions of the agent are also recorded by action index value. For example, if the agent moved to the "Right" at the start position, the index value 0.1, which is the index value of moving to the "Right", was recorded. A vector **E** which records all the action index values from the start position to the end position is used, as shown in Figure 11 (c).

As shown in Figure 11 (a), the space **X** is a collection of the working-memory of the agent moved from the start position to the goal position. Its inverse matrix $\mathbf{X}^{-1}$ is shown in Figure 11 (b). By multiplying $\mathbf{X}^{-1}$ by **E**, a vector **C** is generated as shown in Figure 11 (d).

$$\begin{pmatrix} 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 58.0 & 84.0 & 23.0 \\ 0.4 & 1.2 & 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 \\ 0.9 & 94.0 & 84.0 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \\ 0.9 & 1.2 & 1.4 & 0.9 & 78.0 & 58.0 & 84.0 & 23.0 \\ 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 84.0 & 23.0 \\ 1.2 & 1.4 & 0.9 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \\ 1.2 & 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 23.0 \\ 1.4 & 0.9 & 84.0 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \end{pmatrix}$$

**(a) X**

$$\begin{pmatrix} -0.03 & 2.77 & -0.03 & 0.03 & 0.01 & -0.03 & -0.06 & 0.00 \\ 0.00 & 0.01 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & -0.01 \\ 0.00 & -0.01 & 0.00 & 0.00 & 0.00 & -0.01 & 0.00 & 0.01 \\ 0.00 & -0.08 & 0.00 & -0.09 & 0.00 & 0.09 & 0.00 & 0.00 \\ -0.01 & -0.02 & 0.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.02 & 0.03 & 0.00 & 0.00 & -0.02 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.00 & 0.00 & 0.01 & 0.00 & -0.01 & 0.00 \\ 0.00 & -0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.00 \end{pmatrix}$$

**(b) $\mathbf{X}^{-1}$**

$$\begin{bmatrix} 0.1 \\ 0.0 \\ 0.1 \\ 0.1 \\ 0.8 \\ 0.4 \\ 0.8 \\ 0.4 \end{bmatrix} \qquad \begin{bmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{bmatrix}$$

**(c) E** **(d) C**

**Figure 11**. Working-memory and recorded index value of the sensor output

When the vector **C** is generated, each action of the agent at each relative position can be calculated by multiply the vector of the working-memory by **C**, as expressed by equation (1) introduced in Section2.

When the dark matter matrix is utilized, a unique action index value can be calculated even if the output values of the sensor are the same. For example, at the position (3, 2) and (3, 4), the index values of the sensor output value are the same 1.2, as shown in Figure 12 (a) and (b), where the index values are recorded at the first elements of the two vectors. In the working-memory, the values of the dark-matter values in the second to the eight elements of the two vectors are different, as shown in Figure 12 (a) and (b).

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 1.2 | 1.4 | 0.9 | 12.0 | 78.0 | 58.0 | 84.0 | 23.0 |

**(a)** Values of the working-memory at position (3,2)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 1.2 | 1.0 | 0.3 | 0.9 | 1.2 | 1.4 | 0.9 | 23.0 |

**(b)** Values of the working-memory at position (3,4)

**Figure 12**. Working-memory and recorded index value of the sensor output

By multiplying the two vectors of the working-memory by the energy vector **E**, two different action index values 0.4 and 0.8 are obtained, as shown in equation (4) and (5). The index value 0.4 means that the agent should take the action “Moving Down” at the position (3, 2), and the index value 0.8 means that the agent should take the action “Moving Up” at the position (3, 4).

$$[1.2, 1.4, 0.9, 12.0, 78.0, 58.0, 84.0, 23.0] \times \begin{bmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{bmatrix} = 0.4 \tag{4}$$

$$[1.2, 1.0, 0.3, 0.9, 1.2, 1.4, 0.9, 23.0] \times \begin{bmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{bmatrix} = 0.8 \tag{5}$$

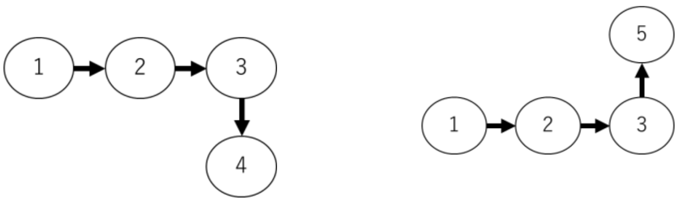
During the above process, the agent obtained skills to take appropriate actions at different positions leading it to move from the start position to the goal positions. That is, skills are acquired through experience of the agent. Considering the definition of *knowledge*, which is defined as “facts, information, and skills acquired through experience or education”, it can be found that ***knowledge is expressed in the dark-matter matrix*** of the space matrix **X**.

## 4. The phenomenon of inapplicability of experience and the parallel spaces

The phenomenon of inapplicability of experience occurs when we use our experience to make judgements, but it fails to guide us correctly. Sometimes, we make wrong judgements and decisions based on experience. For example, in stock trading, we decide the current trade based on the past trends of the stock and our trading experience. However, it often happens that the stock falls when we buy, or rises when we sell.

Figure 13 shows an example of the phenomenon of inapplicability of experience. In the figure, the circles are the positions of an agent and arrows indicate position transitions.

As shown in Figure 13 (a), initially, the agent is in position 1, then it moves to the next position, position 2. When it reaches position 3, it transitions to the next position, position 4. If an agent is trained and obtains the experience as shown in Figure 13 (a), it will always move from position 3 to position 4. However, if the next position of position 3 is position 5, the phenomenon of inapplicability of experience happens as shown in Figure 13 (b).



**(a)** The next position of position 3 should be the position based on the experience

**(b)** If the next position of position 3 is not position 4 but position 5, then the experience of (a) is inapplicable.

**Figure 13**. An example of the phenomenon of inapplicability of experience

When this phenomenon happens, it is impossible to determine which position will be the next in the current position, as shown in Figure 14. In Figure 14, both position 4 and position 5 are the next positions after position 3. In this case, we cannot empirically determine which position will be the next position of position 3.

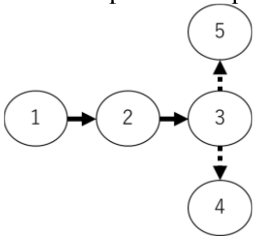
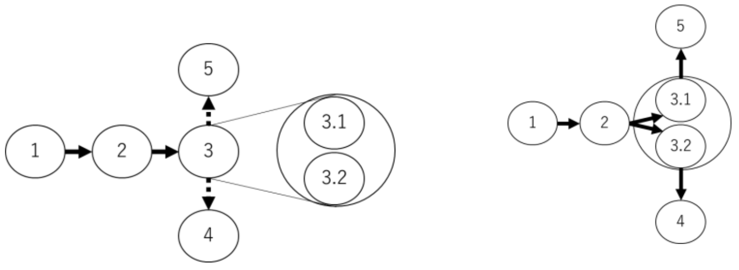


**Figure 14**. Both the position 4 and the position 5 will be the next position of position 3

One reason for this phenomenon is due to the lack of detection accuracy of the sensor. When the sensor's detection accuracy is insufficient, it cannot detect whether the current position should be position 3.1 or position 3.2, as shown in Figure 15 (a). Since it can only detect the current position as position 3, it cannot determine whether the next position should be position 4 or position 5. When the sensor has enough detection accuracy to detect whether the current position is position 3.1 or position 3.2, it can determine whether the next position should be position 4 or position 5, as shown in Figure 15 (b).



**(a)** Sensor's lack of detection accuracy

**(b)** The sensor has enough detection accuracy

**Figure 15**. Sensor's lack of detection accuracy causes the phenomenon of inapplicability of experiences.

The **parallel space model** proposed in this paper is a model to illustrate this phenomenon. Figure 16 is an example. In figure 16, there are an observation space and two parallel spaces, Space1 and Space2. The observation space is a projection space of the parallel spaces. That is, the observation space is a space that the sensor can detect. In Figure 16, the sensor can only detect *x* and *y* direction. It cannot detect *z* direction.

If two agents are trained in Space1 and Space2, respectively, they will move from position 3 to position 4 and position 5, respectively. The phenomenon of inapplicability of experience will not occur. But in the observation space, as shown in Figure 16, this phenomenon will occur. If the agent moves in Space2 based on the experience obtained from Space1, from position 3, it will move to a non-existent position in Space2, position 4. If the agent moves in Space1 based on the experience obtained from Space2, from position 3, it will move to a non-existent position in Space1, position 5. From the point of view in the observation space, from position 3, sometimes the agent moves to position 4 but sometimes it moves to position 5. In the observation space, we cannot predict what position the agent will move to from position 3 until it has moved.

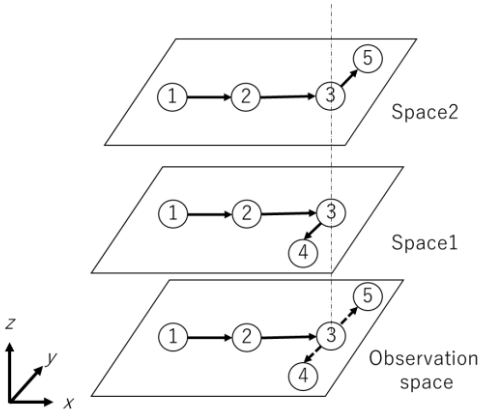


**Figure 16**. An example of parallel spaces

Suppose an agent is used to control an automated cleaning robot and it will clean two floors. The two floors are represented by Space1 and Space2, respectively. The robot's position on the floor plane is represented by *x* and *y* coordinates. The floor where the robot is located is represented by *z*-coordinates. The robot's sensors can detect positions of the robot in the *x* and *y* directions, but not the *z* direction. The positions of the robot are represented as 1, 2, 3 and 4 when the robot is on the Space1 floor plan. If the robot is on the Space2 floor plan, the positions of the robot are represented as 1, 2, 3 and 5. When the robot is on the floor plan Space1, as shown in Figure 17 (a), the matrix **X** is constructed with the elements of the first column are the position of the robot and the other columns are experiences and dark-matter. The inverse matrix of the matrix **X**, $\mathbf{X}^{-1}$ is represented in Figure 17 (b). The elements of vector **E** are the next positions of the current positions. As shown in Figure 17 (c), the next positions are 2, 3 and 4 when the current positions are 1, 2 and 3, respectively. When the robot reached to the position 4, its next position is also 4. The rule vector **C**, shown in Figure 17 (d), is calculated multiplying $\mathbf{X}^{-1}$ by **C**.

$$\begin{pmatrix} 1.0000 & 94.0000 & 84.0000 & 12.0000 \\ 2.0000 & 1.0000 & 84.0000 & 12.0000 \\ 3.0000 & 2.0000 & 1.0000 & 12.0000 \\ 4.0000 & 3.0000 & 2.0000 & 1.0000 \end{pmatrix} \quad \begin{pmatrix} -0.0081 & 0.0020 & -0.0158 & 0.2629 \\ 0.0107 & -0.0107 & -0.0002 & 0.0028 \\ 0.0000 & 0.0119 & -0.0122 & 0.0032 \\ 0.0002 & 0.0003 & 0.0883 & -0.0665 \end{pmatrix} \quad \begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \end{bmatrix} \quad \begin{pmatrix} 0.9781 \\ -0.0002 \\ -0.0003 \\ 0.0889 \end{pmatrix}$$

**(a) X** **(b) $\mathbf{X}^{-1}$** **(c) E** **(d) C**

**Figure 17**. Space1

Figure 18 shows the matrices when the robot is on the floor plan Space2. Same as those of in Figure 17, the space **X**, its inverse matrix $\mathbf{X}^{-1}$, the next position vector **E** and the rule vector **C** are represented in Figure 18 (a), (b), (c) and (d), respectively.

$$\begin{pmatrix} 1.0000 & 94.0000 & 84.0000 & 12.0000 \\ 2.0000 & 1.0000 & 84.0000 & 12.0000 \\ 3.0000 & 2.0000 & 1.0000 & 12.0000 \\ 5.0000 & 3.0000 & 2.0000 & 1.0000 \end{pmatrix} \quad \begin{pmatrix} -0.0064 & 0.0016 & -0.0125 & 0.2082 \\ 0.0107 & -0.0107 & -0.0001 & 0.0022 \\ 0.0001 & 0.0119 & -0.0122 & 0.0025 \\ -0.0002 & 0.0004 & 0.0875 & -0.0526 \end{pmatrix} \quad \begin{bmatrix} 2 \\ 3 \\ 5 \\ 5 \end{bmatrix} \quad \begin{pmatrix} 0.9701 \\ -0.0003 \\ -0.0124 \\ 0.1752 \end{pmatrix}$$

**(a) X** **(b) $\mathbf{X}^{-1}$** **(c) E** **(d) C**

**Figure 18**. Space2

As shown in Figure 16, on both floor plans, Space1 and Space2, the agent moves in the same way from the position 1 to the position 3. Therefore, experiences of the agent at the position 3 are the same. The experience stored in the working memory is shown in Figure 19. It is a row vector. Its first element 3.0000 represents the current robot's position, the second element 2.0000 presents its previous position and the third element 1.0000 is the start position of the robot. The last element 12.0000 is the *dark-matter*.

$$\begin{pmatrix} 3.0000 & 2.0000 & 1.0000 & 12.0000 \end{pmatrix}$$

**Figure 19**. Experiences stored in the working memory

Although the same experience is stored in the working memory at position 3, different next position is obtained using different rule vectors. The rule vector **C** of Space1 and the rule vector **C** of Space2 are shown in Figure 17(d) and 18(d), respectively. Multiplying the row vector stored in the working memory by the rule vector **C** respectively, the next position, 4, in the Space1, and the next position, 5, are calculated respectively, as shown in Figure 20 (a) and (b).

$$\begin{pmatrix} 3.0000 & 2.0000 & 1.0000 & 12.0000 \end{pmatrix} * \begin{pmatrix} 0.9781 \\ -0.0002 \\ -0.0003 \\ 0.0889 \end{pmatrix} = 4$$

(a) Calculating the next position at position 3 using the rule of Space1

$$\begin{pmatrix} 3.0000 & 2.0000 & 1.0000 & 12.0000 \end{pmatrix} * \begin{pmatrix} 0.9701 \\ -0.0003 \\ -0.0124 \\ 0.1752 \end{pmatrix} = 5$$

(b) Calculating the next position at position 3 using he rule of Space2
**Figure 20**. Calculating the next position at position 3

**Space expansion** is necessary when the phenomenon of inapplicability of experience happens. In the example shown in Figure 16, if only the *x* and *y* dimensions are used, experience cannot be used to decide which position, position 4 or 5 will be the next position when the current position is position 3. Adding a new sensor is one of the methods for space expansion. For example, we can add a color sensor to the cleaning robot to detect which floor it is on. Painting the floor of Space1 as yellow and the floor of Space2 as green, the output of the color sensor will be different on different floors. As a result, the space is expanded from one space to two parallel spaces, on which it can be found out which floor the robot is on and which will be the next position when the robot is at position 3.

To find out which space the agent is on, we can use sensors that can detect the next possible moving position. For example, if the cleaning robot’s sensors can tell if it can move to position 4 or 5 from position 3, it can know which space it is on.

Another method is trial-and-error. Suppose the robot is on Space1. If it encounters an obstacle that prevents it from moving to position 4 from position 3, it can infer that robot is on Space2 instead of Space1.

## 5. Conclusion and future work

In this paper, we have reviewed the concept of "***dark-matter***" and its relation to experience and knowledge. We have also reviewed the concept of "***space***" as a matrix that is derived from experience. We have shown the characteristic of the concept of space's "***rules***" represented as vectors and how the rules can be used to generate outputs for agents based on their inputs. We have used examples to illustrate how experience and knowledge can be expressed using the concept of "***dark-matter***". We have also reviewed the working memory mechanism that records the agent's experience and creates the space matrix. Furthermore, we have proposed a new model of "***parallel spaces***" that can explain the "phenomenon of inapplicability of experience". This model reveals why agents sometimes make wrong decisions based on their experience. The main contribution of this paper is that we have uncovered the cause of the "phenomenon of inapplicability of experience" and suggested solutions. In this paper, we have also proposed that for each space, only one rule vector is needed to calculate the output for a given input. A parallel space model can be represented by a three-dimensional matrix, where the first and the second dimensions of the matrix present "***spaces***". However, space expansion is also needed to create a new "***space***" when the "phenomenon of inapplicability of experience" happens. We have presented two methods for space expansion: adding new sensors and trial-and-error. As our future work, we plan to develop application systems based on the proposed methods and the mechanism.

## References

[1] Chen, X. and Kiyoki, Y., "*On Semantic Spatiotemporal Space and Knowledge with the Concept of "Dark-Matter"*," Information Modelling and Knowledge Bases XXXIII, IOS Press, pp.110-128, 2021.

[2] Chen, X. and Kiyoki, Y., "*A query-meaning recognition method with a learning mechanism for document information retrieval*," Information Modelling and Knowledge Bases XV, IOS Press, Vol. 105, pp.37-54, 2004.

[3] Chen, X. and Kiyoki, Y., "*A dynamic retrieval space creation method for semantic information retrieval*," Information Modelling and Knowledge Bases XVI, IOS Press, Vol. 121, pp.46-63, 2005.

[4] Kiyoki, Y. and Kitagawa, T., "*A semantic associative search method for knowledge acquisition*," Information Modelling and Knowledge Bases, IOS Press, Vol. VI, pp.121-130, 1995.

[5] Kitagawa, T. and Kiyoki, Y., "*A mathematical model of meaning and its application to multidatabase systems*," Proc. 3rd IEEE International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems, pp.130-135, April 1993.

[6] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A multi-language oriented intelligent information retrieval system utilizing a semantic associative search method,*" Proceedings of the 17th IASTED International Conference on Applied Informatics, pp.135-140, 1999.

[7] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A semantic metadata-translation method for multilingual cross-language information retrieval,*" Information Modelling and Knowledge Bases XII, IOS Press, Vol. 67, pp.299-315, 2001.

[8] Kiyoki, Y., Kitagawa, T. and Hitomi, Y., "*A fundamental framework for realizing semantic interoperability in a multidatabase environment*, " International Journal of Integrated Computer-Aided Engineering, Vol.2, No.1(Special Issue on Multidatabase and Interoperable Systems), pp.3-20, John Wiley & Sons, Jan. 1995.

[9] Kiyoki, Y., Kitagawa, T. and Hayama, T., "*A metadatabase system for semantic image search by a mathematical model of meaning*, " ACM SIGMOD Record, Vol.23, No. 4, pp.34-41, Dec. 1994.

[10] Kiyoki, Y, Chen, X. and Kitagawa, T., "*A WWW Intelligent Information Retrieval System Utilizing a Semantic Associative Search Method*," APWeb'98, 1st Asia Pacific Web Conference on Web Technologies and Applications, pp. 93-102, 1998.

[11] Ijichi, A. and Kiyoki, Y.: "*A Kansei metadata generation method for music data dealing with dramatic interpretation*," Information Modelling and Knowledge Bases, Vol.XVI, IOS Press, pp. 170-182, May, 2005.

[12] Kiyoki, Y., Chen, X. and Ohashi, H.: "*A semantic spectrum analyzer for realizing semantic learning in a semantic associative search space*," Information Modelling and Knowledge Bases, Vol.XVII, IOS Press, pp.50-67, May 2006.

[13] Takano, K. and Kiyoki, Y.: "*A causality computation retrieval method with context dependent dynamics and causal-route search functions*," Information Modelling and Knowledge Bases, ISO Press, Vol.XVIII, pp.186-205, May 2007.

[14] Chen, X. and Kiyoki, Y.: "*A visual and semantic image retrieval method based on similarity computing with query-context recognition*," Information Modelling and Knowledge Bases, IOS Press, Vol.XVIII, pp.245-252, May 2007.

[15] Nitta T, "*Resolution of singularities introduced by hierarchical structure in deep neural networks*," IEEE Trans Neural Netw Learn Syst., Vol.28, No.10, pp.2282-2293Oct. 2017.

[16] Wiatowski, T. and Bölcskei, H., "*A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction*," IEEE Transactions on Information Theory, PP(99) · Dec. 2015.

[17] Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. "*Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*," In Kremer, S. C. and Kolen, J. F. (eds.), *A Field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, 2001.

[18] Hochreiter, S. and Schmidhuber, J., "*Long short-term memory*,". Neural computation, Vol.9, No.8, pp.1735-1780, 1997.

[19] Kalchbrenner, N., Danihelka, I. and Graves, "*A. Grid long short-term memory*," CoRR, abs/1507.01526, 2015.

[20] Chen, X. and Kiyoki, Y., "*On Logic Calculation with Semantic Space and Machine Learning*," Information Modelling and Knowledge Bases XXXI, IOS Press, Vol. 321, pp.324-343, 2019.

[21] Chen, X. Prayongrat, M. and Kiyoki, Y., "*A Concept for Control and Program Based on the Semantic Space Model*," Information Modelling and Knowledge Bases XXXII, IOS Press, Vol. 333, pp. 26-44, 2020.

[22] Chen, X., "*An Exploratory Research on the Expression of Knowledge and Its Generation Process based on the Concept of "Dark-matter"*, Information Modelling and Knowledge Bases XXXIV, IOS Press, Vol. 364, pp. 110-124, 2023.