Hindawi
Complexity
Volume 2022, Article ID 1889348, 18 pages
https://doi.org/10.1155/2022/1889348

Research Article

A New Quantitative Definition of the Complexity of Organized Matters

Tatsuaki Okamoto

NTT Research, 940 Steward Dr, Sunnyvale, CA 94040, USA

Correspondence should be addressed to Tatsuaki Okamoto; tatsuaki.okamoto@gmail.com

Received 3 October 2021; Revised 25 February 2022; Accepted 11 March 2022; Published 7 June 2022

Academic Editor: Eric Campos

One of the most fundamental problems in science is to define the complexity of organized matters quantitatively, that is, organized complexity. Although many definitions have been proposed toward this aim in previous decades (e.g., logical depth, effective complexity, natural complexity, thermodynamics depth, effective measure complexity, and statistical complexity), there is no agreed-upon definition. The major issue of these definitions is that they captured only a single feature among the three key features of complexity, descriptive, computational, and distributional features, for example, the effective complexity captured only the descriptive feature, the logical depth captured only the computational, and the statistical complexity captured only the distributional. In addition, some definitions were not computable; some were not rigorously specified; and any of them treated either probabilistic or deterministic forms of objects, but not both in a unified manner. This paper presents a new quantitative definition of organized complexity. In contrast to the existing definitions, this new definition simultaneously captures all of the three key features of complexity for the first time. In addition, the proposed definition is computable, is rigorously specified, and can treat both probabilistic and deterministic forms of objects in a unified manner or seamlessly. The proposed definition is based on circuits rather than Turing machines and ε-machines. We give several criteria required for organized complexity definitions and show that the proposed definition satisfies all of them.

1. Introduction

1.1. Background. More than seven decades ago, an American scientist, Warren Weaver, classified scientific problems into three classes: problems of *simplicity*, problems of *disorganized* complexity, and problems of *organized* complexity [1]. For example, classical dynamics can be used to analyze and predict the motion of a few ivory balls as they move about on a billiard table. This is a typical problem of simplicity. Imagine then, a large billiard table with millions of balls rolling over its surface, colliding with one another and with the side rails. Although to be sure the detailed history of one specific ball cannot be traced, statistical mechanics can analyze and predict the average motions. This is a typical problem of disorganized complexity. Problems of organized complexity, however, deal with features of an organization such as living things, ecosystems, and artificial things. Here, cells in a living thing are interrelated into an organic whole in their positions and motions, whereas the balls in the above illustration of disorganized complexity are distributed in a helter-skelter manner.

In the tradition of Lord Kelvin, the quantitative definition of complexity is the most fundamental and important notion in problems of complexity [2].

> *"I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of science, whatever the matter may be."*
>
> Lord Kelvin, 1883

The quantitative definition of *disorganized* complexity of physical systems has been established to be *entropy*, which is defined in thermodynamics and statistical mechanics. In a similar manner, the disorganized complexity of information sources (distributions) can be quantitatively defined by Shannon entropy [3, 4].

In contrast, there is no agreed-upon quantitative definition of *organized* complexity. The difficulty comes from the notion that organized complexity could be greatly dependent on our senses and the context or that the objects of organized complexity like living things, ecosystems, and artificial things may be recognized only by intelligent organisms such as human beings and they are often heavily dependent on the context in the environments (e.g., previous knowledge and understanding), that is to say, it is vastly different from the measures of disorganized complexity such as entropy that simply quantifies the randomness of the objects.

We may therefore wonder whether such sensory, vague, and context-dependent things can be rigorously defined in a unified manner covering various living things to artificial things. Many investigations nonetheless have been pursued toward this aim in the last decades, for example, logical depth by Bennett [5], effective complexity by Gel-Man [6–8], natural complexity by Gentili et al. [9–12], thermodynamic depth by Lloyd and Pagels [13], effective measure complexity by Grassberger [14], and statistical complexity by Crutchfield et al. [15–18], although no existing definition has been agreed on in the field [2, 19, 20].

In Section 4, we explain our understanding of why no existing definition is satisfactory to be agreed upon. Roughly speaking, these definitions have the following problems (Section 4.3).

(i) They captured only a single feature among the three key features of complexity; *descriptive*, *computational*, and *distributional* features. The effective complexity (and Kolmogorov complexity [21]) captured only the *descriptive* feature; the logical depth captured only the *computational*; and the statistical complexity and effective measure complexity captured only the *distributional*.

(ii) They treated either a probabilistic or deterministic form of objects, but none of them could treat both forms of objects in a unified manner or seamlessly.

(iii) Some definitions were not computable.

(iv) Some definitions were not rigorously specified.

We then have the following question:

Is there any quantitative definition of organized complexity with all of the following properties?

(i) It simultaneously captures all of the *descriptive*, *computational*, and *distributional* features

(ii) It can treat both probabilistic and deterministic forms of objects in a unified manner or seamlessly

(iii) It is computable

(iv) It is rigorously specified

1.2. Contribution. In this paper, we positively answer this question.

The major results of this paper are as follows:

(1) Summary

This paper presents a new quantitative definition of organized complexity. In contrast to existing definitions, this definition simultaneously captures the three key features of complexity: descriptive, computational, and distributional features. It is also computable, is rigorously specified, and can treat both probabilistic and deterministic forms of objects in a unified manner or seamlessly.

(2) The Proposed Definition of Organized Complexity

The proposed definition is based on *circuits* [22, 23] rather than Turing machines [5–8, 21, 22] and ϵ-machines [15–18]. The new definition for an object (distribution) is given by the shortest size of a stochastic automaton form of circuit, *oc-circuit*, for simulating the object (i.e., it is based on Occam's razor principle and rigorously specified).

Here, we show the proposed definition roughly (see Section 5.2 for the precise definition).

(a) The *oc-circuit* (stochastic automaton form of circuit) is defined as below. Let C be a circuit with N input bits and L output bits (C: $\{0,1\}^N \longrightarrow \{0,1\}^L$), and let $\overline{C}$ be a specified form of C. $(\overline{C}, u, n, (m_1, \ldots, m_K))$ is oc-circuit $\mathscr{C}$ such that

$$\begin{aligned} &(s_{i+1}, y_i) \longleftarrow C(u, \cdot) \longleftarrow (s_i, m_i, r_i), \quad i = 1, 2, \ldots, K, \\ &Y \coloneqq (y_1, \ldots, y_K)_n, \quad (y_1, \ldots, y_K)_n \\ &\quad \text{is the } n\text{-bit prefix of } (y_1, \ldots, y_K), \end{aligned} \tag{1}$$

(see Section 1.3) where s_i is a state at step i, u is an a priori input (universe), m_i is an input at step i, r_i is random bits at step i, $K \coloneqq \lceil n/L \rceil$, and $L = |y_i|$ (bit length of y_i) for all i (Definition 1).

(b) *Organized complexity OC* of an object, distribution X over n bit strings, at precision level δ ($0 \le \delta < 1$) is defined as follows:

$$OC(X, \delta) \coloneqq \min\left\{|\mathscr{C}| \,|\, X \overset{\delta}{\approx} Y \xleftarrow{R} \mathscr{C}\right\}, \tag{2}$$

where $\mathscr{C}$ is an oc-circuit, and $X^\delta \approx Y \xleftarrow{R} \mathscr{C}$ means the output (simulation result) Y of oc-circuit $\mathscr{C}$ is close (with precision level δ) to object X ($|\mathscr{C}|$ denotes the bit length of $\mathscr{C}$). That is, the organized complexity of object X is the shortest size of oc-circuit $\mathscr{C}$ that simulates X (with precision level δ; Definition 2).

(3) Properties of the Proposed Definition (See Section 5.3 for the Precise Descriptions)

(a) For any distribution X over $\{0,1\}^n$ ($n \in \mathbb{N}$) and any precision level $\delta > 0$, (the proposed definition of organized complexity) $OC(X, \delta)$ can be computed (Theorem 1).

(b) Any ϵ-machine can be simulated by an oc-circuit (as a special case with no descriptive nor computational feature; Theorem 2).

(c) We introduce a structured variant of the proposed definition, $OC^S(X, \delta)$, for object X and

precision level δ (Definition 3) and show the following properties:

(i) For any distribution X over $\{0,1\}^n$ $(n \in \mathbb{N})$ and any precision level $\delta > 0$, $OC^S(X, \delta)$ can be computed (Theorem 3).

(ii) Let $OC^S_{\{\mathrm{NAND}\}}(X, \delta)$ be defined on NAND-based circuits, while $OC^S(X, \delta)$ is defined on (AND, OR, NOT)-based circuits. Then, for a constant $O(1)$ in n,

$$\left|OC^S_{\{\mathrm{NAND}\}}(X, \delta) - OC^S(X, \delta)\right| \leq O(1). \quad (3)$$

(Theorem 4)

(4) Comparison

We give five criteria required for organized complexity definitions in Section 3.3 and summarize the comparison of our definition (Section 5) with the existing definitions (Section 4) in Table 1 in terms of the criteria and Table 2 in terms of the approach and machinery. (The comparison is described in Sections 4.3 and 5.1.)

As shown in Table 1, the proposed definition satisfies all of the criteria for the first time. As shown in Table 2, the proposed definition is based on Occam's razor principle and the oc-circuit machinery. As a result, it is rigorously specified, is computable, and captures all three key features of complexity.

1.3. Notations. The sets of natural, rational, and real numbers are denoted by $\mathbb{N}$, $\mathbb{Q}$, and $\mathbb{R}$, respectively. The set of n-bit strings is denoted by $\{0,1\}^n$ $(n \in \mathbb{N})$ and $\{0,1\}^* := \cup_{n\in\mathbb{N}}\{0,1\}^n$, and the null string (0 bit string) is denoted by λ. When $x \in \{0,1\}^*$, $|x|$ denotes the bit length of x. When $a, b \in \mathbb{R}$, $[a, b]$ denotes set $\{x | x \in \mathbb{R}, a \leq x \leq b\} \subset \mathbb{R}$. When $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

When x is a variable and y is a value, $x := y$ denotes that x is substituted or defined by y. A probability distribution over $\{0,1\}^n$ is $\{(a, p_a) | a \in \{0,1\}^n, p_a \in [0,1], \sum_{a\in\{0,1\}^n} p_a = 1\}$. When A is a probability distribution, or the source (machinery) of the distribution, $a \xleftarrow{R} A$ denotes that element $a \in \{0,1\}^n$ is randomly selected from A according to its probability distribution. When A is a set, $a \xleftarrow{U} A$ denotes that a is randomly selected from A with a uniform distribution.

When X and Y are two distributions, the statistical distance of X and Y, $\mathrm{SD}(X, Y)$, is defined by $1/2 \cdot \sum_{\alpha\in\{0,1\}^*} |\Pr[\alpha \xleftarrow{R} X] - \Pr[\alpha \xleftarrow{R} Y]|$, and $X^\delta \approx Y$ denotes that $\mathrm{SD}(X, Y)$ is bounded by δ. Then we say X and Y are statistically δ-close.

When Y is a distribution, $(Y)_n$ denotes the n bit restriction (n bit prefix) of Y, that is, $(Y)_n$ is a distribution over $\{0,1\}^n$ and $\Pr[y \xleftarrow{R} (Y)_n] = \sum_{z\in\{0,1\}^*} \Pr[(y, z) \xleftarrow{R} Y]$.

When S is a set, $\#S$ denotes the number of elements of S.

2. Objects

The existing complexity definitions can be categorized into two classes. One is a class of definitions whose objects are *deterministic strings*, and the objects of the other class of definitions are *probability distributions*. As for the traditional complexity definitions, the Kolmogorov complexity is categorized in the former, and entropy is in the latter. Among the above-mentioned organized complexity definitions, logical depth and effective complexity are in the former, and thermodynamic depth, effective measure complexity, and statistical depth are in the latter.

Which is more appropriate as the objects of organized complexity?

The objects of complexity are everything around us, stars and galaxies in space, living things, ecosystems, artificial things, and human societies. The existence of everything can be recognized by us only through observations. For example, the existence of many things are observed through devices such as the telescope, microscope, and various observation apparatus and electronic devices. We can take things around us directly into our hands and sense them, but they are also recognized by our brains as electronic nerve signals transmitted from the sensors of our five senses through the nervous system. That is, all objects of complexity are recognized as the result of observations by various apparatus and devices, including the human sensors of our five senses.

Since quantum mechanics governs the microworld, observed values are determined in a probabilistic manner. This is because observed values (data) obtained when observing microphenomena (quantum states) in quantum mechanics are randomly selected according to a certain probability distribution corresponding to a quantum state (e.g., entangled superposition).

How, then, are observed values in macrophenomena? For example, if some sort of radio signals are received and measured, they would almost certainly be accompanied by noise. There are various reasons why noise becomes mixed in with signals, and one of them is thought to be the probabilistic phenomena of electrons, thermal noise. Similarly, various types of noise will be present in the data obtained when observing distant astronomical bodies. This can be caused by the path taken by the light (such as through the atmosphere) and factors associated with the observation equipment.

Even in the case of deterministic physical phenomena, chaos theory states that fluctuations in initial conditions can lead to diverse types of phenomena that behave similarly to those of random systems. In short, even a deterministic system can appear to be a quasi-probabilistic system. However, even a system of this type can become a true (nonquasi) probabilistic system if initial conditions fluctuate due to some noise, for example, thermal noise. There are also many cases in which a quasi-probabilistic system associated with chaos cannot be distinguished from a true probabilistic system depending on the precision of the observation equipment. Here, even quasi-probabilistic systems may be treated as true probabilistic systems.

Table 1: Comparison of the proposed definition with the existing ones regarding criteria.

Definitions of complexity	Objects (Prob./Det.)	Simply regular	Simply random	Key features of highly organized objects			Computability
				Descriptional	Computational	Distributional	
Kolmogorov complexity	×	✓	×	✓	×	×	×
Effective complexity	×	✓	✓	✓	×	×	×
Logical depth	×	✓	✓	×	✓	×	×
Effective measure complexity	×	✓	✓	×	×	✓	✓
Statistical complexity	×	✓	✓	×	×	✓	✓
Proposed definition	✓	✓	✓	✓	✓	✓	✓

Table 2: Approach and machinery of the proposed definition and the existing ones.

Definitions	Approach	Rigorously specified	Machinery	Computability	Key features
Kolmogorov complexity	Occam's razor	✓	Turing machine	×	Descriptive
Effective complexity	Occam's razor	✓	Turing machine	×	Descriptive
Logical depth	Occam's razor	✓	Turing machine	×	Computational
Statistical complexity	Occam's razor	✓	Stochastic finite-state automaton (hidden Markov; ϵ-machine)	✓	Distributional
Effective measure complexity	Information theoretical	✓	None	✓	Distributional
Natural complexity	Framework	×	None	—	—
Thermodynamics depth	Framework	×	None	—	—
Proposed definition	Occam's razor	✓	Stochastic automaton form of circuit (oc-circuit)	✓	All (of the three)

—: Not easy to characterize.

Thus, when attempting to give a quantitative definition of the complexity of observed data, the source of those observed data would be a probability distribution, and the observed data themselves would be randomly selected values according to that distribution. If we now consider the complexity of a phenomenon observed using certain observation equipment, the object of this complexity should not be the observed data selected by chance from the source but rather the probability distribution itself corresponding to the source of the observed data.

It is known that some parts of genome patterns appear randomly distributed over a collection of many samples (over generations). Here, we can suppose a source (probability distribution) of genome patterns from which each genome pattern is randomly selected. Also, in this case, the object of complexity should not be each genome pattern but rather the source (probability distribution) of the genome patterns.

Therefore, hereafter, we consider that an object of complexity is a *probability distribution* in this paper. Here, note that we treat a deterministic string as an object since a deterministic string is a special case of a probability distribution (where only a value occurs with probability 1 and the others with 0).

How can we identify a source or a probability distribution from observed data? It has been studied as the *model selection theory* in statistics and information theory, for example, AIC (Akaike's information criterion) by Akaike [24] and MDL (minimum description length) by Rissanen [3, 25, 26]. Here, given a collection of data, find the most likely and succinct model (source, i.e., probability distribution) of the data.

In this paper, however, it is outside the scope, that is, we do not care how to find such a source from a collection of observed data (through the model selection theory). We here suppose that a source (probability distribution) is given as an object of organized complexity and focus on how to define the complexity of such a given source quantitatively.

There are roughly two types of observed data: one type is data observed at a point in time, and the other is time-series data. Genome pattern data are examples of the former, and data obtained from an observation apparatus for a certain period are examples of the latter. In any case, without loss of generality, we here assume that observed data x is bounded and expressed in binary form, that is, $x \in \{0,1\}^n$ for some $n \in \mathbb{N}$, since any physically observed data have only finite precision (with rounding error). Then the *source* of the observed data, X, which is an object of organized complexity in this paper, is a probability distribution over $\{0,1\}^n$ for some $n \in \mathbb{N}$ such that $X \coloneqq \{(x, p_x) | x \in \{0,1\}^n, 0 \leq p_x \leq 1, \sum_{x \in \{0,1\}^n} p_x = 1\}$.

3. Criteria

In this section, we will show the criteria required for the quantitative definitions of organized complexity.

3.1. Examples. To begin with, let us consider the following example. We give a chimpanzee a computer keyboard and prompt the chimpanzee to freely hit the keys, resulting in a string of characters. Let us assume an output of 1,000 alphabetical characters. At the same time, we select a string of

1,000 characters from one of Shakespeare's plays. Naturally, the character string input by the chimpanzee is gibberish possessing no meaning, which undoubtedly makes it easy for us to distinguish that string from a portion of a Shakespearean play.

However, is there a way to construct a mathematical formulation of the difference between these two strings that we can easily tell apart? Why is it so easy for us to distinguish these two strings? The answer is likely that the chimpanzee's string is simply random (or disorganized) and meaningless to us, while Shakespeare's string is highly organized and meaningful. In short, if we can mathematically define the amount of organized complexity (or meaningful information), we should distinguish between these two strings.

What then are the sources of the observed data, the chimpanzee's string and Shakespeare's string. Let us return to the source of the chimpanzee's string creation without thinking of it as simply a deterministic string. Here, for simplicity, we suppose that the scattered hitting of keys by the chimpanzee is the same as a random selection of hit keys. At this time, the source of the chimpanzee's string is the probability distribution in which any particular 1,000-character string can be randomly selected from all possible 1,000-character strings with equal probability.

What, then, would be the source of Shakespeare's string? We can surmise that when Shakespeare wrote down this particular 1,000-character string, a variety of expressions within his head would have been candidates for use and that the 1,000-character string used in the play would have finally been selected from those candidates with a certain probability. The candidates selected must certainly be connected by complex semantic relationships possessed by English words. Such complex semantic relationships of human language have been extensively studied by the modeling with complex networks [27]. Accordingly, the source of Shakespeare's string must be the complex probability distribution of candidate expressions connected by complex semantic relationships.

Here, note that the source of Shakespeare's string (distribution of candidate expressions within his head) is an imaginary example of distribution. As we will observe below, we may easily perceive the features of the organized complexity in such an example. As mentioned in Section 2, this paper supposes a source is given and focuses on defining the complexity of the given source. Hence, it is outside the scope of this paper how to find the source (distribution) of Shakespeare's strings (for practical applications of our complexity definition, however, we may need to consider how to identify a source. See Section 6.3.)

We now consider three imaginary examples, Cases 1, 2, and 3. In Case 1, candidate expression 1 has the probability of 0.017, candidate expression 2 has the probability of 0.105, . . ., candidate expression 327 has the probability of 0.053. The other expressions have the probability of 0, that is, hundreds of candidate expressions occurred in his head consciously or unconsciously. Finally, one of them was randomly chosen according to the distribution. As more simplified cases, Case 2 is where candidate expression 1 has the probability of 2/7, candidate expression 2 has the probability of 5/7, and the others have the probability of 0, that is, only two candidate expressions occurred in his head. Finally, one of them was randomly chosen. Case 3 is where only a single expression has the probability of 1 and the others have the probability of 0, that is, a deterministic string case; he selected the expression without hesitation. These expressions and distributions should be highly organized and structured with complex semantic relationships.

3.2. Three Key Features. We can find three key features, descriptive, computational, and distributional features, in the examples of Shakespeare's string above.

Case 1 may have a highly complicated distribution with complex semantic relationships. The *distributional* feature of complexity captures such distributional complexity. In contrast, Case 3 has no such feature since it is deterministic. (Note that the distributional feature of complexity for distribution can be represented by the statistical complexity, Section 4.2.3, with an ϵ-machine.)

Instead, Case 3 (and Cases 1 and 2) may have two other features, descriptive and computational ones. The *descriptive* feature of complexity captures how much description (information or knowledge) is needed to produce an expression. For example, Shakespeare should need a tremendous amount of knowledge about culture, history, and language to write his plays. Its complexity (amount of knowledge) can be characterized as the descriptive feature. The complexity of the plays' semantic information (story, etc.) can also be characterized as the descriptive feature. (Note that the descriptive feature of complexity for a deterministic string can be represented by the Kolmogorov complexity, Section 4.1.1, with a Turing machine.)

The *computational* feature of the complexity of an expression captures how much amount of computation (work) is needed to produce the expression. Suppose that Shakespeare came up with an idea of what he wanted to express. He then tried to find a proper literary expression or rhetoric. It might be easy (less thinking) for him in some cases and might require much work (hard thinking) in other cases. As mentioned above, we can analyze complex semantic relationships of human language with complex network models [27]. To find a proper expression with complex semantic relationships can be modeled as solving a mathematical problem in complex networks for human language. We can consider such computational complexity (to find a proper expression or to solve a mathematical problem in complex networks) as the computational feature. (Note that the computational feature of complexity for a deterministic string can be represented by the logical depth, Section 4.1.2, with a Turing machine.)

In general, the organized complexity of an object (e.g., Case 1) can have all three key features, descriptive, computational, and distributional features. Therefore, a desirable quantitative definition of organized complexity should capture these three features simultaneously.

3.3. Criteria for the Quantitative Definition of Organized Complexity. Considering the observation mentioned above, we provide the following criteria for specifying the quantitative definition of organized complexity.

(1) Object (Prob. and Det.)

The objects of the complexity definition should be probability distributions. In addition, the complexity definition should treat any deterministic strings (as a special case of distributions) and any general distributions in a unified manner or seamlessly. For example, it should treat Cases 1 and 2 (probabilistic distributions) and Case 3 (deterministic strings) of Section 3.1 in a unified manner or seamlessly.

(2) Simply Regular

Simply regular objects, "problems of simplicity" by Weaver [1], should have low-organized complexity.

(3) Simply Random

Simply random objects, "problems of disorganized complexity" by Weaver [1], for example, the source of the chimpanzee's string, should have low-organized complexity.

(4) Key Features of Organized Complexity

Highly organized objects, "problems of organized complexity" by Weaver [1], for example, the source of Shakespeare's string (Cases 1, 2, and 3), should have high-organized complexity. In addition, the organized complexity definition should simultaneously capture all of the three key features (descriptive, computational, and distributional) of complexity.

(5) Computability

Given an object, the organized complexity of the object should be computable (or recursive in the computation theory).

4. Existing Quantitative Definitions of Complexity

We now survey the typical quantitative definitions of organized complexity in the literature using the criteria above. Here, we can categorize them into two classes. One is a class of definitions whose objects are *deterministic strings*, and the objects of the other class of definitions are *probability distributions*.

4.1. Objects Are Deterministic Strings

4.1.1. Kolmogorov Complexity. The notion of the Kolmogorov complexity was independently proposed by Solomonoff, Kolmogorov, and Chaitin [3, 21, 28–30].

Roughly, the Kolmogorov complexity of string x is the size of the shortest program (on a Turing machine) to produce string x, that is, it is based on Occam's razor principle (see (iv) in Section 4.3) and rigorously (uniquely) specified. This definition captures the descriptive feature of the complexity of deterministic strings.

More precisely, let U be a reference universal prefix Turing machine (see [21] for the reference universal prefix machine). Then, the Kolmogorov complexity, $K(x)$, of string $x \in \{0,1\}^*$ is defined by

$$K(x) = \min\{|z| \mid U(z) = x, z \in \{0,1\}^*\}. \tag{4}$$

Note that the Kolmogorov complexity is a quantitative definition for *disorganized* complexity or the degree of randomness for deterministic strings (the counter notion of entropy for distributions).

The criteria in Section 3.3 show that the Kolmogorov complexity has the following properties (summarized in Tables 1 and 2).

(1) Object (Prob. & Det.): The objects are only deterministic strings (bad).

(2) Simply Regular: Simple (or very regular) objects have low Kolmogorov complexity (good).

(3) Simply Random: Simply random objects, deterministic n bit strings, that are uniformly and randomly chosen from $\{0,1\}^n$ have high Kolmogorov complexity (bad).

(4) Key Features of Organized Objects: For highly organized objects that focus on the computational feature (e.g., highly organized objects generated from small strings through long running-time computations), their Kolmogorov complexities are low (bad). In other words, the Kolmogorov complexity cannot capture the computational feature of organized complexity, while it captures the descriptive feature. In addition, the Kolmogorov complexity cannot capture the distributional feature because its objects are only deterministic strings.

(5) Computability: The Kolmogorov complexity of an object (string) is not computable (bad).

4.1.2. Logical Depth. Bennett [5] introduced *logical depth* with the intuition that complex objects are those whose most plausible explanations describe long causal processes. To formalize the intuition, Bennett employs the methodology of algorithmic information theory, the Kolmogorov complexity. The logical depth can be considered as one of the (computational) resource-bounded variants of the Kolmogorov complexity [21].

The logical depth of a deterministic string, Bennett's definition for measuring organized complexity, is dependent on the running time of the programs that produce the string and whose length is relatively close to the minimum in a sense. This definition captures the computational feature of the complexity of deterministic strings.

More precisely, the logical depth of string x at significance level $\epsilon := 2^{-b}$ [21] is

$$\min\left\{t \,\middle|\, \frac{m_t(x)}{m(x)} \geq \epsilon\right\}, \tag{5}$$

where we define $m_t(x)$ and $m(x)$ by

$$m_t(x) \coloneqq \sum_{U^t(p)=x} 2^{-l(p)}, \quad m(x) \coloneqq \sum_{U(p)=x} 2^{-l(p)}, \tag{6}$$

that is, it is based on Occam's razor principle (see (iv) in Section 4.3) and rigorously (uniquely) specified. Here, U is the reference universal prefix Turing machine (for the Kolmogorov complexity), and U^t is a specific class of U whose running time is bounded by t steps. $l(p)$ is the length of program p.

In terms of the criteria shown in Section 3, the logical depth has the following properties (summarized in Tables 1 and 2).

(1) Object (Prob. and Det.): The objects are only deterministic strings (bad).

(2) Simply Regular: Simple (or very regular) objects have low logical depth (good).

(3) Simply Random: Simply random objects have low logical depth (good).

(4) Key Features of Organized Objects: For highly organized objects that focus on the descriptive feature (e.g., high Kolmogorov complexity objects generated from (compressed) long strings through very small running-time computations), their logical depths are low (bad). Hence, logical depth cannot capture the descriptive feature of organized complexity, while it captures the computational feature. In addition, the logical depth cannot capture the distributional feature because its objects are only deterministic strings. Note that other (computational) resource-bounded variants of the Kolmogorov complexity, such as computational depth [31] and Kt complexity [21, 32], have the same drawback as the logical depth because they also focus on the computational feature and cannot capture the descriptive feature of complexity.

(5) Computability: The logical depth of an object (string) is not computable because it is based on the Kolmogorov complexity or Turing machines (bad).

4.1.3. Effective Complexity. Effective complexity [6, 7] was introduced by Gell-Mann and is based on the Kolmogorov complexity. To define the complexity of an object, Gell-Mann considers the shortest description of the distribution in which the object is embedded as a typical member. Here, "typical" means that the negative logarithm of its probability is approximately equal to the entropy of the distribution. The effective complexity captures the descriptive feature of complexity since it is defined by the Kolmogorov complexity of the distribution induced by an object (deterministic string).

The effective complexity of string x is

$$\min\{K(E)| - \log \Pr_E(x) \approx H(E)\}, \tag{7}$$

where $K(E)$ is the Kolmogorov complexity of distribution E, that is, the length of the shortest program to list all members, r, of E together with their probabilities, $\Pr_E(r)$, and $H(E)$ is the Shannon entropy of E, that is, it is based on Occam's razor principle (see (iv) in Section 4.3) and rigorously (uniquely) specified.

In terms of the criteria shown in Section 3, the effective complexity has the following properties (summarized in Tables 1 and 2).

(1) Object (Prob. and Det.): The objects are only deterministic strings (bad). Technically, however, we can consider distribution E to be an object of the effective complexity in place of x.

(2) Simply Regular: Simple (or very regular) objects have low effective complexity (good).

(3) Simply Random: Simply random objects have low effective complexity (good).

(4) Key Features of Organized Objects: For highly organized objects that focus on the computational feature (e.g., a distribution, E, induced by object, x, that is generated from a small string through very long running-time computation), their effective complexities are low (bad). Hence, the effective complexity cannot capture the computational feature of organized complexity, while it captures the descriptive feature. In addition, the effective complexity cannot capture the distributional feature because its objects are only deterministic strings.

(5) Computability: The effective complexity of an object (string) is not computable, since it is based on the Kolmogorov complexity or Turing machines (bad).

4.1.4. Natural Complexity. There are various definitions and frameworks of natural complexity in the literature [9, 11, 12, 33–37]. Here, we focus on the definition in [9] since it has the most general form of natural complexity among them.

The *natural complexity* (NaC) [9] is a multivariate function that depends on the following:

(i) The multiplicity (Mu) of the network, which is the number N of nodes. The nodes are described by the temporal functions of the type: $x_1(t), x_2(t), \ldots, x_N(t)$.

(ii) The interconnection (Ic) of the network, which is the number L of links among the nodes. All the potential edges can be represented by the $N \times N$ adjacency matrix A that describes the system's wiring and the interaction strength between the nodes.

(iii) The diversity of the nodes (DiN) when the functions, $x_1(t), x_2(t), \ldots, x_N(t)$, are not the same.

(iv) The diversity of links (DiL) when the elements of the matrix A are different.

(v) The variability of the nodes (VaN), which depends on how much the functions $x_1(t), x_2(t), \ldots, x_N(t)$ evolve in time.

(vi) The variability of links (VaL), which depends on how much the coefficients of the adjacency matrix A (i.e., $a_{11}(t), a_{12}(t), \ldots, a_{NN}(t)$) change over time.

(vii) The integration (Ig) of the nodes' and links' features, which gives rise to those properties that are called emergent because they belong to the entire network. As has been sustained in the previous paragraph, the emergent properties depend on the network's architecture. The network's topology can be inferred from parameters such as the degree distribution (P(d)), the clustering coefficient (C) of the nodes, and the mean path length ($\overline{\text{sp}}$) of the network. Furthermore, the emergent properties depend on the environment surrounding the system as proved by the Darwinian evolution of life [38].

In synthesis, natural complexity (NaC) results to be a seven-variable function of the type:

$$\text{NaC} = f(\text{Mu}, \text{Ic}, \text{DiN}, \text{DiL}, \text{VaN}, \text{VaL}, \text{Ig}). \tag{8}$$

Although the natural complexity (NaC) provides a good framework for studying the complexity of various systems expressed with network forms as shown in [9] and Section 6.2, it has the following problems as a quantitative complexity definition.

(1) NaC is a framework of definitions but not a rigorous definition because there is no concrete description/specification of f, DiN, DiL, VaN, VaL, and Ig (P(d), C, $\overline{\text{sp}}$, etc.) and there is no way to compute a concrete value of NaC given an object (network expression). Even if NaC were concretely specified (by specifying f, DiN, DiL, VaN, VaL, and Ig (P(d), C, $\overline{\text{sp}}$, etc.)), it would be ad hoc, and there would be many other concrete specifications of NaC. Therefore, it would be hard to specify NaC uniquely.

Note that this drawback is shared with the thermodynamic depth to be introduced below, which is not rigorously specified due to the arbitrariness of macroscopic states.

(2) The object of NaC is restricted to network expressions but not general. Network expressions have been used broadly for the characterization and analysis of complex systems. However, individual models or more direct analyses for complex systems rather than network expressions have also been employed [10, 39–41]. Therefore, a complexity definition is desirable to give no particular restriction on the objects. Otherwise, we would need many complexity definitions depending on different classes of objects, for example, network expressions and other individual models.

Here, note that if the object of a complexity definition is general (as described in pointer (1) of Section 3.3), it can treat any particular class of the object, for example, deterministic network expressions, as shown in Section 6.2. That is, if the object of a complexity definition is general, it can be applied to any particular class of the object, but if the object is restricted, it cannot be applied to objects outside the restriction.

Therefore, the natural complexity (NaC) is not rigorously specified, and it is not easy to characterize the definition in terms of the criteria in Section 3 (summarized in Table 2).

4.2. Objects Are Probability Distributions

4.2.1. Thermodynamic Depth. The *thermodynamic depth* was introduced by Lloyd and Pagels [13] and shared some informal motivation with logical depth, where complexity is considered a property of the evolution of an object.

We now assume the set of histories or trajectories that result in object (distribution) S_0. A trajectory is an ordered set of macroscopic states (distributions) $S_{-L-1}, \ldots, S_{-1}, S_0$. The thermodynamic depth of object S_0 is

$$H(S_{-L+1}, \ldots, S_{-1} | S_0), \tag{9}$$

where $H(A, \ldots, B|C)$ is the conditional entropy of combined distribution $(A, \ldots, B)$ with condition C.

The thermodynamic depth has the following problems as a quantitative complexity definition.

(1) The thermodynamic depth is not rigorously defined. This is because it is not defined how long the trajectories (what value of L) should be, and there is no description on how to select macroscopic states in the definition [13]. If there are thousands of possible sets of macroscopic states, we would have thousands of different definitions of the thermodynamic depth.

(2) The object is essentially limited by the selection of macroscopic states. In order to define the complexity of an object S_0, a set of macroscopic states $\mathcal{S}$ whose complexity is comparable to or more than that of S_0 should be established beforehand. Hence, if $\mathcal{S}$ is fixed, or the thermodynamic depth of $\mathcal{S}$ is concretely defined, it cannot measure the complexity of an object whose complexity is more than that of $\mathcal{S}$. That is, any concrete definition of this notion can measure only a restricted subset of objects, that is, any concrete and generic definition is impossible in the thermodynamic depth.

Therefore, the thermodynamic depth is not rigorously specified, and it is not easy to characterize the definition in terms of the criteria in Section 3 (summarized in Table 2).

4.2.2. Effective Measure Complexity. The *effective measure complexity* was introduced by Grassberger [14] and measures the average amount by which the uncertainty of a symbol in an (infinite) random string (distribution) decreases due to the knowledge of previous symbols. The effective measure complexity captures the distributional feature of complexity.

For distribution X^N over $\{0, 1\}^N$ ($N \in \mathbb{N}$), $H(X^N)$ is the Shannon entropy of X^N. Let $h_N \coloneqq H(X^{N+1}) - H(X^N)$ and

$h \coloneqq \lim_{N \longrightarrow \infty} h_N$. The effective measure complexity of $\{X^N\}_{N \in \mathbb{N}}$ is

$$\sum_{N=0}^{\infty} (h_N - h). \tag{10}$$

This difference quantifies the perceived randomness, which, after further observation, is discovered to be order [19]. It is information theoretically and rigorously (uniquely) specified.

In terms of the criteria shown in Section 3, the effective measure complexity has the following properties (summarized in Tables 1 and 2).

(1) Object (Prob. and Det.): The objects are only probability distributions, and deterministic strings are outside the scope of this definition (the complexity is 0 for any deterministic string; bad).

(2) Simply Regular: Simple (or very regular) objects have low effective measure complexity (good).

(3) Simply Random: Simply random objects have low effective measure complexity (good).

(4) Key Features of Organized Objects: For highly organized objects that focus on the descriptive or computational feature (e.g., highly organized complex objects (distributions), $\{X^N\}_{N \in \mathbb{N}}$, that are almost deterministic and generated from (compressed) long strings or from small strings through very long running-time computation), their effective measure complexities are low since they are almost deterministic (bad). In other words, the effective measure complexity cannot capture the descriptive and computational features of complexity, while it captures the distributional feature.

(5) Computability: The effective measure complexity of an object (distribution) is computable since it is information theoretically defined and not based on a Turing machine (good).

4.2.3. Statistical Complexity. The *statistical complexity* was introduced by Crutchffeld and Young [15]. Here, to define the complexity, the set of causal states S and the probabilistic transitions between them are modeled in the so-called ϵ-machine, which produces a stationary distribution of causal states, D_S. The mathematical structure of the ϵ-machine is a stochastic finite-state automaton or hidden Markov model. The statistical complexity of object (distribution) is the minimum value of the Shannon entropy of D_S, where the object is equivalent to the output of the ϵ-machine, that is, it is based on Occam's razor principle (see the footnote in Section 4.3) and rigorously (uniquely) specified because the minimum value of the Shannon entropy corresponds to the shortest size of expression. It captures the distributional feature of complexity.

Let S_i for $i = 1, .., k$ be causal states, $S \coloneqq \{S_1, \ldots, S_k\}$, and T_{ij} be the probability of a transition from state S_i to state S_j, that is, $T_{ij} \coloneqq \Pr[S_j|S_i]$. Each transition from S_i to S_j is associated with an output symbol, σ_{ij} (e.g., $\sigma_{ij} \in \{0, 1\}$). Then, $\Pr[S_i]$, the probability that S_i occurs in the infinite run of the ϵ-machine, is given by the eigenvector of matrix $T \coloneqq (T_{ij})$, since $\sum_{i=1}^{k} \Pr[S_i] \cdot T_{ij} = \Pr[S_j]$, that is, $(\Pr[S_1], \ldots, \Pr[S_k]) \cdot T = (\Pr[S_1], \ldots, \Pr[S_k])$. Hence, the machine produces a stationary distribution of states, D_S. The output of the ϵ-machine is the infinite sequence of σ_{ij} induced by the infinite sequence of the transition of states. That is, ϵ-machine outputs a distribution, Σ_S, over $\{0, 1\}^{\infty}$, induced by D_S.

The statistical complexity of object X (distribution), denoted C_1, is the minimum value of the Shannon entropy of D_S, $H_1(D_S)$, when $\Sigma_S = X$:

$$C_1 \coloneqq \min\{H_1(D_S)|\Sigma_S = X\}. \tag{11}$$

A more generalized notion, C_α $(0 \le \alpha \le \infty)$, is defined by the Reny entropy of D_S in place of the Shannon entropy, that is,

$$C_\alpha \coloneqq \min\{H_\alpha(D_S)|\Sigma_S = X\}, \tag{12}$$

where C_1 is the case where $\alpha \coloneqq 1$ as H_1 is the Shannon entropy and $C_0 \coloneqq \min\{\log \#S|\Sigma_S = X\}$ $(\alpha \coloneqq 0$; $\#S$ is the number of elements of set S; for $\alpha \coloneqq \infty$, H_∞ is the mini-entropy).

In terms of the criteria shown in Section 3, the statistical complexity has the following properties (summarized in Tables 1 and 2).

(1) Object (Prob. & Det.): Statistical complexity C_α for any α $(0 \le \alpha \le \infty)$ can measure only probability distributions as objects, since for any deterministic string, there is only one state S_1 with this deterministic string and $\Pr[S_1] = 1$, that is, $C_\alpha = 0$ $(0 \le \alpha \le \infty)$. Thus, none of C_α can treat probability distributions and deterministic strings in a unified manner (bad).

(2) Simply Regular: Simple (or very regular) objects have low statistical complexity (good).

(3) Simply Random: Simply random objects have low statistical complexity (good).

(4) Key Features of Organized Objects: For highly organized objects that focus on the descriptive or computational feature, their statistical complexities are low, that is, the statistical complexity cannot capture the descriptive and computational features (bad). Here, we show two typical cases below:

(i) For highly organized complex objects (on the descriptive or computational feature) that are almost deterministic and are generated from (compressed) long strings (information) or from small strings through very long running-time computations), their statistical complexities are low.

(ii) For a highly organized complex object (on the descriptive or computational feature) characterized by an ϵ-machine with $(T, S, D_S, \{\sigma_{ij}\})$ that has low statistical complexity, C_α, and that has $\{\sigma_{ij}\}$ generated from (compressed) long

strings (information) or from small strings through very long running-time computations), its statistical complexity is low. Note that C_α depends on only D_S but is independent of $\{\sigma_{ij}\}$.

That is, the statistical complexity cannot capture the descriptive and computational features of complexity, while it captures the distributional feature.

See Remark 3 for more precise observation about the properties of the statistical complexity.

(5) Computability: The statistical complexity of an object is computable since it is information theoretically defined and not based on a Turing machine (good).

4.3. Summary. In summary, the existing definitions have the following properties and shortcomings (see Tables 1 and 2):

(i) Any existing complexity definition can capture only a single feature of complexity; for example, the Kolmogorov complexity and effective complexity capture only the descriptive feature, the logical depth captures only the computational, and the statistical complexity and effective measure complexity capture only the distributional.

(ii) Any existing complexity definition can treat either a probabilistic or deterministic form of objects, but none of them can treat both in a unified manner or seamlessly.

(iii) The definitions on Turing machines (Kolmogorov complexity, effective complexity, and logical depth) are not computable. The definitions not on Turing machines (the effective measure complexity and the statistical complexity) are computable, but they can only capture the distributional feature (but not descriptive and computational ones) due to the lack of computational machinery in the definitions. Note that an ϵ-machine for the statistical complexity, which produces a stochastic or hidden Markov process, is information-theoretical but not computational machinery.

(iv) We can rigorously specify the complexity in Occam's razor principle and information-theoretical approaches (Kolmogorov complexity, effective complexity, logical depth, statistical complexity, and effective measure complexity), while the natural complexity (NaC) and the thermodynamics depth are not rigorously specified as described in Sections 4.1.4 and 4.2.1. Occam's razor principle can be described as "if presented with a choice between indifferent alternatives, then one ought to select the simplest one" [21]. We can specify a complexity definition on Occam's razor principle by adopting the shortest one (expression/time/entropy) for simulating the object. Due to the minimality principle, the definition can be specified uniquely.

5. Proposed Quantitative Definition of Organized Complexity

5.1. Introduction of the Proposed Definition. We now propose a new quantitative definition of organized complexity. This definition simultaneously captures the three key features of complexity: descriptive, computational, and distributional. It is also computable, is rigorously specified, and can treat both probabilistic and deterministic forms of objects in a unified manner or seamlessly. We give several criteria required for organized complexity definitions and show that the proposed definition satisfies all of them (Table 1).

Roughly speaking, the proposed quantitative definition is given by the shortest size of a stochastic automaton form of *circuit* (oc-circuit) that simulates an object (distribution). That is, in place of Turing machines and ϵ-machines, the proposed definition employs another class of computational machinery, a stochastic automaton form of *circuit*, *oc-circuit*. It is based on Occam's razor principle and rigorously specified (Table 2).

We now clarify the differences between oc-circuit (circuit), ϵ-machine, and Turing machine. We also show the advantage of using oc-circuit (circuit) in terms of the three key features of complexity.

(i) oc-Circuit versus ϵ-machine

Our approach by oc-circuits is more general than the approach by ϵ-machines for the statistical complexity because an oc-circuit can simulate any ϵ-machine as a special case without the descriptive and computational features. In other words, an ϵ-machine is information-theoretical machinery to produce a stochastic process or hidden Markov process with only the distributional feature. In contrast, an oc-circuit is computational machinery to produce a distribution with all three key features, descriptive, computational, and distributional features (Theorem 2 and Remark 3).

Another special case of oc-circuits lacks the distributional feature but has descriptive and computational features. Such an oc-circuit produces (simulates) a deterministic string. See pointer (1) in Section 5.5 for the description of such an oc-circuit.

Hence, oc-circuits include two typically specific cases. A special oc-circuit, ϵ-machine, produces information-theoretical distributions with only the distributional feature. Another is a special oc-circuit, pointer (1) in Section 5.5, which produces deterministic strings with only the descriptive and computational features. In general, an oc-circuit produces any distribution with the three key features.

Hence, the proposed definition can treat both probabilistic and deterministic forms of objects in a unified manner or seamlessly.

(ii) Circuit versus Turing machine

The major difference between circuits [22, 23] and Turing machines [22] is that a single (universal)

Turing machine can compute any size of input, while a single circuit can compute a fixed size of input. In spite of the difference, any bounded time Turing machine computation with n bit input can be computed by a circuit with n bit input. The finiteness of a circuit (oc-circuit) yields the computability of our definition (Theorem 1). It contrasts with the incomputability of Turing machine-based definitions (Kolmogorov complexity, logical depth, and effective complexity) [21], which is due to the incomputability (undecidability) of the halting problem of Turing machines.

(iii) Advantage of using circuits

An advantage of basing on circuits is that it can simultaneously capture the three key features: descriptive, computational, and distributional features.

(a) Descriptive feature: An input on a Turing machine for a descriptive feature can be realized for a circuit with the input or hard-wire of the description. If a descriptive feature requires d bit input to a Turing machine, it requires d bit input/hard-wire for a circuit to describe it. Hence, if an object has a d bit descriptive feature represented on a Turing machine (e.g., the Kolmogorov complexity of the object is d), the size of a circuit producing the object should be $\Omega(d)$. Then, the proposed complexity definition of this object (the shortest size of an oc-circuit simulating this object) should be $\Omega(d)$. Therefore, the proposed complexity definition can capture the descriptive feature of complexity.

(b) Computational feature: A t-time computation of a Turing machine can be computed by an $O(t^2)$-size circuit [22]. If a computational feature requires t-time computation on a Turing machine, the size of a circuit to compute it for the computational feature would be $\Omega(t^2)$. Hence, if an object has a t-time computational feature represented on a Turing machine (e.g., the logical depth of the object is t), the size of a circuit producing the object would be $\Omega(t^2)$. We now suppose that the data compression by the t-time computation is greater than the computation overhead in terms of the circuit size. Then, the proposed complexity definition of this object (the shortest size of an oc-circuit simulating this object) would be $\Omega(t^2)$. Therefore, the proposed complexity definition can capture the computational feature of complexity.

(c) Distributional feature: An oc-circuit includes an ϵ-machine structure as a partial structure, which characterizes a distributional feature (Remark 3). If an object has an s bit distributional feature represented on an ϵ-machine (e.g., the statistical complexity of the object is s), the size of an oc-circuit simulating this object should be $\Omega(s)$. Therefore, the proposed complexity definition can capture the distributional feature of complexity.

5.2. Proposed Definition. We now define a new measure of organized complexity. First, we define our computation model, *oc-circuit*.

Definition 1 (oc-circuit). Let circuit C with N input bits and L output bits be a directed acyclic graph in which every vertex is either an input gate of in-degree 0 labeled by one of the N input bits, or one from the basis of gates, $B := \{\text{AND}, \text{OR}, \text{NOT}\}$. Among them, L gates are designated as the output gates. That is, circuit C actualizes a Boolean function: $\{0,1\}^N \longrightarrow \{0,1\}^L$.

Let $s_i \in \{0,1\}^{N_s}$ be a state at step i ($i \in \mathbb{N}$), $u \in \{0,1\}^{N_u}$ be an a priori input (universe), $m_i \in \{0,1\}^{N_m}$ be an input at step i, $r_i \xleftarrow{U} \{0,1\}^{N_r}$ be random bits at step i, and $N := N_u + N_s + N_m + N_r$. Then

$$(s_{i+1}, y_i) \longleftarrow \boxed{C(u,\cdot)} \longleftarrow (s_i, m_i, r_i), \quad i = 1, 2, \ldots, K, \tag{13}$$

that is, $(s_{i+1}, y_i) := C(u, s_i, m_i, r_i)$, where $y_i \in \{0,1\}^{L_y}$, $N_m \le L_y$ is the output of C at step i, and $L := N_s + L_y$. Let V be the number of vertexes of C.

Let $\widetilde{C} := ((w_{ij})_{i=1,\ldots,V;j=1,\ldots,V}, (\ell_1, \ldots, \ell_V), (o_1, \ldots, o_L))$ be a canonical description of C, where $(w_{ij})_{i=1,..,V;j=1,..,V}$ is the adjacent matrix of directed graph C, that is, $w_{ij} := 1$ iff there is an edge from vertex i to vertex j, and $w_{ij} := 0$ otherwise, and ℓ_i ($i = 1, .., V$) is the label of the i-th vertex, $\ell_i \in \{1, .., N, \text{AND}, \text{OR}, \text{NOT}\}$, that is, each vertex i is labeled by ℓ_i, and $o_i \in \{1, \ldots, V\}$ is the vertex designated to the i's output, that is, $(o_1, \ldots, o_L)$ is the sequence of output gates. Hereafter, we abuse the notation of C to denote $\widetilde{C}$, the canonical description of C.

Let $\mathscr{C} := (\overline{C}, u, n, \vec{m})$ be an "oc-circuit," and Y be the output of $\mathscr{C}$, where $\overline{C} := (C, N_u, N_s, N_m, N_r, L_y, s_1)$, $K := \lceil n/L_y \rceil$, $\vec{m} := (m_1, .., m_K)$, and $Y := (y_1, \ldots, y_K)_n$ (see Section 1.3 for the notation of $(\ldots)_n$).

Here, $\mathscr{C}$ is expressed by a specified form $\mathbb{C} \subset \{0,1\}^*$, that is, $\mathscr{C} \in \mathbb{C}$ iff $\mathscr{C}$ is a valid form of oc-circuit, and it can be efficiently (polynomial-time in $|\mathscr{C}|$) verified whether $\mathscr{C} \in \mathbb{C}$.

The output, Y, of $\mathscr{C}$ can be expressed by $Y \xleftarrow{R} \mathscr{C}$, that is, $Y \xleftarrow{R} (\overline{C}, u, n, \vec{m})$, where the probability of distribution Y is taken over the randomness of $r_i \xleftarrow{U} \{0,1\}^{N_r}$ ($i = 1, \ldots, K$).

Then, $\overline{C}$, u, and $\vec{m}$ are called the "logic," "universe," and "semantics" of oc-circuit $\mathscr{C}$, respectively.

Remark 1. Circuit C of oc-circuit $\mathscr{C}$ is a probabilistic circuit, where uniformly random strings, $r_i \xleftarrow{U} \{0,1\}^{N_r}$ for $i = 1, \ldots, K$, are input to C and the output of $\mathscr{C}$ is distributed over the random space of $\{r_i\}_{i=1,\ldots,K}$.

Here, note that $\{r_i\}_{i=1,\ldots,K}$, which is an input to C, is not included in $\mathscr{C}$, while the other inputs to C, u and $\{m_i\}_{i=1,\ldots,K}$, are included in $\mathscr{C}$. In other words, the size of the randomness, $\sum_{i=1}^{K} |r_i|$, is ignored in the size of $\mathscr{C}$ or the definition of the organized complexity (see Definition 2), while

N_r and a part of C regarding the randomness are included in $\mathscr{C}$. This is because the randomness, $\{r_i\}_{i=1,\ldots,K}$, is just the random source of $\mathscr{C}$'s output distribution and has no organized complexity itself. Hence, simply random objects are characterized to have low-organized complexity based on the size of oc-circuit $\mathscr{C}$ (see pointer (3) in Section 5.5).

Definition 2 (organized complexity). Let X be a distribution over $\{0,1\}^n$ for some $n \in \mathbb{N}$.

"Organized complexity" OC of distribution X at precision level δ $(0 \le \delta < 1)$ is

$$OC(X,\delta) \coloneqq \min\left\{|\mathscr{C}||X^{\delta} \approx Y \xleftarrow{R} \mathscr{C}\right\}, \tag{14}$$

where $\mathscr{C} \coloneqq (\overline{C}, u, n, \vec{m})$ is an oc-circuit and $|\mathscr{C}|$ denotes the bit length of the binary expression of $\mathscr{C}$ (see Section 1.3 for the notations of $^{\delta}\approx$).

We call oc-circuit $\mathscr{C}^X \coloneqq (\overline{C}^X, u^X, n, \vec{m}^X)$ the shortest (or proper) oc-circuit of X at precision level δ, iff $X^{\delta} \approx Y \xleftarrow{R} \mathscr{C}^X$ and $|\mathscr{C}^X| = OC(X,\delta)$. If there are multiple shortest oc-circuits of X, that is, they have the same bit length, the lexicographically first shortest one is selected as the shortest oc-circuit.

Then, $\overline{C}^X, u^X$, and $\vec{m}^X$ are called the "proper logic," "proper universe," and "proper semantics" of X at precision level δ, respectively. Here, $X^{\delta} \approx Y \xleftarrow{R} \mathscr{C}^X \coloneqq (\overline{C}^X, u^X, n, \vec{m}^X)$.

5.3. Properties of the Definition

5.3.1. Computability

Theorem 1. *For any distribution X over $\{0,1\}^n$ $(n \in \mathbb{N})$ and any precision level $\delta > 0$, $OC(X,\delta)$ can be computed.*

Proof. For any distribution $X \coloneqq \{(x, p_x) | x \in \{0,1\}^n, p_x \in [0,1], \sum_{x\in\{0,1\}^n} p_x = 1\}$ and any precision level $\delta > 0$, there always exists another distribution $X' \coloneqq \{(x, p'_x) | x \in \{0,1\}^n, 0 \le p'_x \le 1, p'_x \in \mathbb{Q}, \sum_{x\in\{0,1\}^n} p'_x = 1\}$ such that $X'^{\delta} \approx X$. (Here, note that $p_x \in \mathbb{R}$ is changed to $p'_x \in \mathbb{Q}$ provided that $X'^{\delta} \approx X$.)

We then construct the truth table of Boolean function $f: \{0,1\}^{\ell} \longrightarrow \{0,1\}^n$ such that $\Pr[x = f(r) | r \xleftarrow{U} \{0,1\}^{\ell}] = p'_x$ for all $x \in \{0,1\}^n$, where the probability is taken over $\xleftarrow{U}\{0,1\}^{\ell}$. Such a function, f, can be achieved by setting truth table $T_f \coloneqq \{(r, f(r))\}_{r\in\{0,1\}^{\ell}}$ such that $\#\{r | f(r) = x\}/2^{\ell} = p'_x \in \mathbb{Q}$ for all $x \in \{0,1\}^n$.

Since any Boolean function can be achieved by a circuit with basis $B \coloneqq \{\text{AND, OR, NOT}\}$ [42], we construct circuit C^* for oc-circuit $\mathscr{C}^*$ with $N_s \coloneqq 1, N_u = N_m \coloneqq 0$ (i.e., $u = m_i \coloneqq \lambda$), $N_r \coloneqq \ell$, $K \coloneqq 1$, $L_y \coloneqq n$, and $s_1 = s_2 \coloneqq 0$. That is, $(0, y_1) \coloneqq C^*(\lambda, 0, \lambda, r_1)$, and the output of $\mathscr{C}^*$ is $y_1 \in \{0,1\}^n$ with the same distribution as that of X' over the randomness of $r_1 \xleftarrow{U} \{0,1\}^{N_r}$. That is, $y_1 \xleftarrow{R} \mathscr{C}^*$ and $X^{\delta} \approx X' = y_1$.

From the definition of OC, $OC(X,\delta) \le |\mathscr{C}^*|$.

We then exhaustively check all values of Z with $|Z| < |\mathscr{C}^*|$ whether Z is a valid oc-circuit such that $X^{\delta} \approx Y \xleftarrow{R} Z$. Note that we can efficiently check whether or not Z is the correct form of an oc-circuit or $Z \in \mathbb{C}$. Finally, we find the shortest one among the collection of Z (and $\mathscr{C}^*$) satisfying the condition. Clearly, the size of the shortest one is $OC(X,\delta)$. □

Remark 2. As clarified in this proof, given object (distribution) X and precision level δ, the proposed definition of organized complexity uniquely determines (computes) not only organized complexity $OC(X,\delta)$ but also the shortest (proper) oc-circuit, $\mathscr{C}^X$, including proper logic $\overline{C}^X$, proper universe u^X, and proper semantics $\vec{m}^X$ of X. In other words, the definition characterizes the complexity features of object X, that is, it characterizes not only organized complexity $OC(X,\delta)$ but also structural complexity features of X, for example, the descriptive feature by the size of u^X and $\vec{m}^X$ and the computational and distributional features by the size of $\overline{C}^X$ and (N_r, N_s) of $\overline{C}^X$.

5.3.2. Relation between oc-Circuit and ϵ-Machine.

In the following theorem, we show that the notion of oc-circuit with the proposed organized complexity includes the ϵ-machine with statistical complexity introduced in Section 4 as a special case.

Theorem 2. *Any ϵ-machine can be simulated by an oc-circuit.*

Proof. Given ϵ-machine, $(\{S_1,\ldots,S_k\}, (T_{ij})_{i=1,..,k;j=1,..,k}, (\sigma_{ij})_{i=1,..,k;j=1,..,k})$, we construct oc-circuit $\mathscr{C} \coloneqq ((C, N_u, N_s, N_m, N_r, L_y, s_1), u, n, (m_1, m_2, \ldots))$ such that $N_s \coloneqq \lceil \log_2 k \rceil + 1$ (i.e., $S_i \in \{0,1\}^{N_s}$), $N_u \coloneqq 0$, $N_m \coloneqq 0$, $N_r \coloneqq \max_{i,j}\{|T_{ij}|\}$, $L_y \coloneqq \max_{ij}\{|\sigma_{ij}|\}$, $s_1 \coloneqq S_1$ (initial causal state), $n \coloneqq \infty$, $u \coloneqq \lambda$ (null string), $m_\iota \coloneqq \lambda$ $(\iota = 1, 2, \ldots)$, and C is achieved to satisfy $\Pr[(S_j, \sigma_{ij}) \coloneqq C(\lambda, S_i, \lambda, r)] = T_{ij}$ for $i, j = 1, .., k$, where $|T_{ij}|$ is the bit length of the binary expression of T_{ij}, and the probability is taken over the randomness of $r \xleftarrow{U} \{0,1\}^{N_r}$ in each execution of C. It is clear that the behavior of this oc-circuit with respect to the causal states is exactly the same as that for the given ϵ-machine. □

Remark 3 (features of the proposed complexity). Our complexity definition can treat more general cases with $N_u > 0$ and $N_m > 0$, while statistical complexity only considers a limited case with $N_u = 0$ and $N_m = 0$, that is, it ignores the descriptive feature as well as the computational feature.

For example, a sequence in a genome pattern that is common to all individuals is considered to be determined in the evolution process and has some biological meaning. The biological knowledge of DNA necessary to understand the DNA sequences can be captured by logic $\overline{C}$ and universe u of the oc-circuit $\mathscr{C}$ (where $|u| = N_u > 0$), and the characteristic information (biological meaning) on each genome pattern $(^{\delta}\approx Y \xleftarrow{R} \mathscr{C})$ can be captured by semantics $\vec{m}$ of $\mathscr{C}$ (where $|m_i| = N_m > 0$).

Our complexity definition also covers the computational feature, which is characterized by the size of logic $\overline{C}$ of oc-circuit $\mathscr{C}$ in our definition.

5.3.3. Structured Organized Complexity. We can construct a circuit C using another basis of gates, for example, {NAND} and {AND, NOT}, in place of {AND, OR, NOT}. We express an oc-circuit using such a basis by $\mathscr{C}_{\{\text{NAND}\}}$ and $\mathscr{C}_{\{\text{AND,NOT}\}}$, respectively. We also express the organized complexity of X using such a circuit by $OC_{\{\text{NAND}\}}(X,\delta)$ and $OC_{\{\text{AND,NOT}\}}(X,\delta)$, respectively.

Based on such a different basis of gates, a natural variant of the proposed organized complexity, *structured organized complexity*, is given below.

Definition 3 (structured organized complexity). Let X be a distribution over $\{0,1\}^n$ for some $n \in \mathbb{N}$.

Let $\mathscr{C}^S := (\overline{C}^S, u, n, \overrightarrow{m})$ be a structured oc-circuit, where $\overline{C}^S = (\text{macros}, C_{\text{macros}}, N_u, N_s, N_m, N_r, L_y, s_1)$ and macros represents a set of macro gates (subroutine circuits) that are constructed from basis gates and that can be hierarchically constructed, where a level of macro gates are constructed from lower levels of macro gates. In addition, macros is notationally abused as the canonical description of macro gates in macros, which is specified in the same manner as that in the canonical description of a circuit. Term C_{macros} is (the canonical description of) a circuit constructed from basis gates B as well as macro gates in macros.

Structured organized complexity OC^S of distribution X at precision level δ $(0 \le \delta < 1)$ is

$$OC^S(X,\delta) := \min\left\{|\mathscr{C}^S| \;\middle|\; X^\delta \approx Y \xleftarrow{R} \mathscr{C}^S\right\}. \tag{15}$$

Theorem 3. *For any distribution X over $\{0,1\}^n$ $(n \in \mathbb{N})$ and any precision level $\delta > 0$, $OC^S(X,\delta)$ can be computed.*

Proof. Given distribution X, we can construct structured oc-circuit $\mathscr{C}^*$ in the same manner as that shown in the proof of Theorem 1. Here, note that any (basic) oc-circuit $\mathscr{C}$ can be expressed as structured oc-circuit $\mathscr{C}^S$ where macros $:= \lambda$, with slightly relaxing the format for structured oc-circuits, or to allow macros $:= \lambda$.

From the definition of OC^S, for any value of $0 < \delta < 1$, $OC^S(X,\delta) \le |\mathscr{C}^*|$.

We then, given δ, exhaustively check all values of Z with $|Z| < |\mathscr{C}^*|$ whether Z is a structured oc-circuit such that $X^\delta \approx Y \xleftarrow{R} Z$. Finally, we select the shortest one among the collection of Z (and $\mathscr{C}^*$) satisfying the condition. Clearly, the size of the shortest one is $OC^S(X,\delta)$. □

Remark 4. As described in Remark 2, the shortest structured oc-circuit with these parameters characterizes the properties of object X. It especially shows the optimized hierarchically structured circuit $\mathscr{C}^S$.

As mentioned above, we can construct a structured oc-circuit using another basis such as {NAND} and {AND, NOT}, for example, $\mathscr{C}^S_{\{\text{NAND}\}}$ and $\mathscr{C}^S_{\{\text{AND,NOT}\}}$. Then, macros in $\mathscr{C}^S_{\{\text{NAND}\}}$ can consist of macro gates of AND, OR, and NOT from NAND gates.

Theorem 4

$$\left|OC^S_{\{\text{NAND}\}}(X,\delta) - OC^S(X,\delta)\right| \le O(1), \tag{16}$$

where X is a distribution over $\{0,1\}^n$ and $O(1)$ is a constant in n.

Proof. Let $\mathscr{C}^S$ be a structured oc-circuit with macros that consists of the NAND gate macro from {AND, OR, NOT} as well as macros from {NAND} (i.e., macros is macros from {AND, OR, NOT}), and C_{macros} be constructed only by NAND and macros. Then $OC^S(X,\delta)$ is upper bounded by $|\mathscr{C}^S|$ for (X,δ), whose minimum value is $OC^S_{\{\text{NAND}\}}(X,\delta) + O(1)$ since the size of the NAND gate macro from {AND, OR, NOT} is a constant in n. Therefore,

$$OC^S(X,\delta) \le OC^S_{\{\text{NAND}\}}(X,\delta) + O(1). \tag{17}$$

Let $\mathscr{C}^S_{\{\text{NAND}\}}$ be a structured oc-circuit with macros that consists of the AND, OR, and NOT gate macros from {NAND} as well as macros from {AND, OR, NOT} (i.e., macros is macros from {NAND}) and C_{macros} be constructed only by {AND, OR, NOT} and macros. Then, $OC^S_{\{\text{NAND}\}}(X,\delta)$ is upper bounded by $\mathscr{C}^S_{\{\text{NAND}\}}$ for (X,δ), whose minimum value is $OC^S(X,\delta) + O(1)$, since the size of the AND, OR, and NOT gate macros from {NAND} is a constant in n. Hence,

$$OC^S_{\{\text{NAND}\}}(X,\delta) \le OC^S(X,\delta) + O(1). \tag{18}$$

□

5.4. Variations. We have more variations of the organized complexity.

(1) Computational Distance: In Definitions 2 and 3, statistical distance is used for defining the closeness $\overset{\delta}{\approx}$. We can replace this with the "computational" closeness $\overset{(\mathscr{D},\delta)}{\approx}$. Here, for two distributions, X and Y, over $\{0,1\}^n$, the computational closeness of X and Y is defined by

$$X \overset{(\mathscr{D},\delta)}{\approx} Y \text{ iff } \forall D \in \mathscr{D},\ 1/2 \cdot |\Pr[1 \xleftarrow{R} D(\alpha) | \alpha \xleftarrow{R} X] - \Pr[1 \xleftarrow{R} D(\alpha) | \alpha \xleftarrow{R} Y]| < \delta,$$

where $\mathscr{D}$ is a class of machines D: $\{0,1\}^n \longrightarrow \{0,1\}$. Intuitively, the computational closeness means that X and Y are indistinguishable at precision level δ by any machine in class $\mathscr{D}$ (the computational closeness/indistinguishability is widely used in cryptography [43]; for example, $\mathscr{D}$ is the class of probabilistic polynomial-time Turing machines and δ is a polynomially negligible function in the security parameter).

(2) Quantum Circuits: Circuit C in Definitions 2 and 3 can be replaced by a "quantum" circuit [44]. In this variation, we assume that the source of observed data is principally given by a quantum phenomenon.

There are several variations of the quantum complexity definition, typically: (1) all inputs and outputs of C are classical strings, (2) only state s_i is a quantum string, and the others are classical, (3) all inputs and outputs of C except output y_i are quantum strings, and (4) all inputs and outputs of C are quantum strings.

(3) Interactions: If an observation object actively reacts similar to a living thing, we often observe it in an interactive manner.

So far in this paper, we have assumed that an object is a distribution that we perceive passively. We can extend the object from such a passive one to an active one with interactive observation.

Suppose that the observation process is interactive between observer A and observation object B. For example, A first sends z_1 to B, which replies x_1 to A, and we continue the interactive process, z_2, x_2, ..., z_J, x_J.

Let $Z \coloneqq (z_1, .., z_J)$ and $X \coloneqq (x_1, .., x_J)$ be distributions. We then define the conditional organized complexity of X under Z with precision level δ, $OC(X\colon Z, \delta)$, which can be defined as the shortest (finite version of) conditional oc-circuit to simulate X (with precision level δ) under Z.

The extended notion of the organized complexity of interactive object (X, Z) can be defined by $OC(X\colon Z, \delta)$.

(4) Generalization of the Size of Circuit: Some applications may employ a metric (e.g., depth) of a circuit in place of the size of a circuit [45].

We can introduce a variant of the proposed definition using another metric f of an oc-circuit in place of the oc-circuit size $|\mathscr{C}|$ such that

$$OC_f(X, \delta) \coloneqq \min\left\{ f(\mathscr{C}) \;\middle|\; X^\delta \approx Y \xleftarrow{R} \mathscr{C} \right\}. \tag{19}$$

5.5. Evaluation in terms of the Criteria. In terms of the criteria described in Section 3, the proposed definition has the following properties (summarized in Tables 1 and 2).

(1) Object: Prob. & Det.

The proposed definition can treat probability distributions and deterministic strings (as special cases of distributions) in a unified manner as the objects (good).

For example, an oc-circuit of pointer (3) in this section is a case for simulating a probability distribution (simply random), and an oc-circuit of pointer (2) in this section is a case for simulating a deterministic string (simply regular).

More generally, for any deterministic string (as a special case of distribution), there exists an oc-circuit with circuit C to simulate the deterministic string by $Y \coloneqq (y_1, \ldots, y_K)_n \in \{0, 1\}^n$, such that

$$(s_{i+1}, y_i) \coloneqq C(u, s_i, m_i, \lambda), \quad i = 1, 2, \ldots, K, \tag{20}$$

where $N_r = 0$, that is, $r_i \coloneqq \lambda$ for $i = 1, \ldots, K$.

For example, any deterministic form of network as an object can be simulated (produced) by this oc-circuit with specifying C for the network (see Section 6.2 for an example of an oc-circuit to produce network expressions).

(2) Simply Regular

Simple (or very regular) objects have low complexity (good).

For example, in a very regular case ("$11\cdots1$" $\in\{0, 1\}^n$), the object can be simulated by an oc-circuit $\mathscr{C} \coloneqq (\overline{C}, 1, n, \lambda)$ with $\overline{C} \coloneqq (C, 1, 1, 0, 0, 1, 0)$, such that

$$(s_{i+1}, 1) \coloneqq C(1, s_i, \lambda, \lambda), \quad i = 1, 2, \ldots, n, \tag{21}$$

where $N_u = 1\,(u = 1)$, $s_i \coloneqq 0$ for $i = 1, .., n + 1$ (i.e., $N_s = 1$), $N_m = N_r = 0$ (i.e., $m_i = r_i = \lambda$), and $L_y = 1$. That is, input size $N = 2$ and output size $L = 2$, that is, C has two gates that are input gates labeled by $(1, 2)$ and that are also output gates. $C \coloneqq ((w_{ij}) \coloneqq I_2, (\ell_1, \ell_2) \coloneqq (1, 2), (o_1, o_2) \coloneqq (2, 1))$, where I_2 is the two-dimensional identity matrix. Hence, $\mathscr{C} \coloneqq ((((1, 0, 0, 1), (1, 2), (2, 1)), 1, 1, 0, 0, 1, 0), 1, n, \lambda)$, and OC ("$11\ldots1$") $\leq c + \log_2 n$, where c is a small constant.

(3) Simply Random

Simply random objects have low complexity (good).

For example, in the case of uniformly random distribution X over $\{0, 1\}^n$, the object can be simulated by an oc circuit $\mathscr{C} \coloneqq (\overline{C}, \lambda, n, \lambda)$ with $\overline{C} \coloneqq (C, 0, 1, 0, 1, 1, 0)$, such that

$$(s_{i+1}, r_i) \coloneqq C(\lambda, s_i, \lambda, r_i), \quad i = 1, 2, \ldots, n, \tag{22}$$

where $N_u = 0$, $s_i \coloneqq 0$ for $i = 1, .., n$ (i.e., $N_s = 1$), $N_m = 0$ (i.e., $u = m_i = \lambda$), $N_r = 1$, $r_i \xleftarrow{U} \{0, 1\}$, and $L_y = 1$. That is, input size $N = 2$ and output size $L = 2$, that is, C has two gates that are input gates labeled by $(1, 2)$ and that are also output gates. $C \coloneqq ((w_{ij}) \coloneqq I_2, (\ell_1, \ell_2) \coloneqq (1, 2), (o_1, o_2) \coloneqq (1, 2))$. Hence, $\mathscr{C} \coloneqq ((((1, 0, 0, 1), (1, 2), (1, 2)), 0, 1, 0, 1, 1, 0), \lambda, n, \lambda)$, and $OC(X) \leq c + \log_2 n$, where c is a small constant.

(4) Key features of Organized Objects

As described in Section 5.1 and Remarks 2 and 3, the proposed definition simultaneously captures the distributional, computational, and descriptive features of organized complexity (good).

Hence, our definition does not have the drawbacks of the existing definitions described in Section 4. That is, the proposed definition correctly estimates the complexity of highly organized objects for

which an existing definition miss-estimates low. In addition, an object estimated by an existing definition high for a feature is also estimated high in our definition.

(5) Computability

The proposed complexity of any object (distributions/strings) is computable as shown in Theorems 1 and 3 (good).

6. Applications of the Proposed Complexity Definition

This section will show several possible applications of the proposed complexity definition. Exploring the applications is one of the most important issues regarding this notion.

6.1. Applications and Approximations. The Kolmogorov complexity is a quantitative definition for *disorganized* complexity (or randomness) of deterministic strings, and it is incomputable (Section 4.1). Nonetheless, many applications have been known [21], for example, theoretical background for inductive reasoning and compression theory, lower bound and average-case analysis of algorithms and computational and communication complexity, and information theory.

Several methods of approximating the Kolmogorov complexity have been developed for more practical applications [11, 21, 39, 46–48]. The methods of approximating the Kolmogorov complexity of an object (deterministic string) are closely related to the compression of the object since the Kolmogorov complexity is the size of an optimally compressed (shortest) program (on a Turing machine) to produce the object and its approximation is considered the size of a nearly optimal (program) compression to produce the object. For example, approximation methods based on data compression such as MDL (minimum description length) [3, 25, 26, 49, 50] were proposed [39, 46]. Approximation methods based on the coding theorem [21], which implies a compression by random sampling, were explored [11, 47, 48] (which were criticized by [51]). However, Faloutsos et al. [39] emphasized the importance of selecting good models as an art rather than the automated methods on the data compression and the coding theorem.

In contrast to the Kolmogorov complexity, the proposed complexity definition (Section 5) is a quantitative definition for *organized* complexity of distributions (including deterministic strings as a special case), and it is computable. Therefore, it should have more advanced applications than the Kolmogorov complexity, for example, (1) theoretical background for machine learning and artificial intelligence; (2) lower bound and average-case analysis of algorithms, networks, and computational and communication complexity regarding organized matters; and (3) semantic information theory.

Although the proposed definition is computable, computing it for an object generally requires at least an exponential time in the object size. Hence, for practical applications, it should be beneficial to explore efficient methods for approximating it.

Similarly to the Kolmogorov complexity, the methods of approximating the proposed complexity definition of an object (distribution) should be closely related to the (oc-circuit form) compression of the object, since the proposed complexity definition is the size of the optimally compressed (shortest) oc-circuit to simulate the object and its approximation is considered the size of a nearly optimal (oc-circuit form) compression to simulate the object.

Here, the approximation should be achieved by more sophisticated compression methods (considering the organized structure of the object) than just a randomness compression for the Kolmogorov complexity approximation. Since the oc-circuit captures the three key features of organized complexity, descriptive, computational, and distributional features, such sophisticated compression methods should utilize these three key features. As emphasized by [39], selecting good models and frameworks as an art that captures the object's key features should be important to seek effective approximation methods.

6.2. Applications to Network Expressions. We now suppose that objects are expressed by network forms, which have been widely used for the natural complexity [9, 11, 27, 33, 34, 36, 37]. Note that the proposed complexity definition can treat (a deterministic form of) network expressions as a particular case of distributions (see pointer (1) in Section 5.5). This section will show how our complexity definition can be applied to network expressions.

If a network (object) is less structured/featured, it should have more compressed (shorter) oc-circuit to produce the network, that is, it has (approximate) lower complexity. Conversely, it should have less compressed oc-circuit if it is more structured/featured, that is, it has (approximate) higher complexity.

For example, we suppose an undirected graph expresses a network with n nodes, represented by an adjacent matrix ($n \times n$ upper-triangular binary matrix) and expressed by an $n(n-1)/2$ bit string. Then, if a network (object) is randomly generated (i.e., almost the least structured/featured), the object should be the uniform distribution of possible networks (not a sampled network), that is, the uniform distribution of $n(n-1)/2$ bit strings. Then, the oc-circuit shown in pointer (3) of Section 5.5 (over $\{0,1\}^{n(n-1)/2}$) produces the uniform distribution of the graphs (networks) and should be nearly the shortest one to produce it. The approximate complexity of this object (the randomly generated network) is the size of the oc-circuit. If a network is very regular (e.g., complete graph; i.e., nearly the least structured/featured), the oc-circuit shown in pointer (2) of Section 5.5 produces the complete graph and should be nearly the shortest one to produce it. The approximate complexity of this object (the complete graph network) is the size of the oc-circuit.

If a network is general, its properties/features such as network structure and topology, which have been extensively studied in the framework and analysis of the natural complexity [9, 11, 27, 33, 34, 36, 37], should be beneficial to

obtain a compressed (short) oc-circuit to produce the network. The size of such an oc-circuit is the approximate complexity. For example, [33, 35] presented frameworks to characterize and analyze social networks and highlight many technical disciplines. To evaluate the (approximate) proposed complexity of an object (social network), we have to find a compressed representation of the object or a short oc-circuit to produce the network, where the characterization and analysis suggested by [33, 35] could be employed.

A network is usually considered deterministic in the natural complexity [9]. However, as shown above, in the random network case, it may be appropriate to consider a network expression as a distribution of networks since some parts of a network are often randomly distributed (e.g., some parts of genome patterns described in Section 2). As shown in Section 3, it may also be helpful to capture the computational feature of networks.

6.3. Remark on the Identification of Objects. This paper focuses on the complexity definition for a given object, and it is outside the scope of this paper how to find or identify the object (Section 2). However, if we analyze or compute the complexity of an object in practical applications, we may have to identify the object before computing the (approximate) complexity of the object. When the object is a distribution, the model selection theory such as MDL [25, 26] and AIC [24] would be used to identify the distribution from sampled data. For a variety of complex systems, we may need a method more specific to each system. For example, when the object is a complex system like biochemical systems, a graphical approach with the observability theory has been explored to identify the system [37].

6.4. Summary and Challenge. Here, we summarize several issues about the applications of the proposed definitions.

(1) Theoretical Applications

As described above, possible theoretical applications of the proposed complexity definition are: (1) theoretical background for machine learning and artificial intelligence; (2) lower bound and average-case analysis of algorithms, networks, and computational and communication complexity regarding organized matters (living things, ecosystems, economic and social systems, artificial things, etc.); and (3)semantic information theory.

(2) Approximation Methods

For practical applications, we need an efficient approximation method of the proposed complexity definition. As mentioned above, such an approximation method is closely related to a compressed (short) oc-circuit to simulate the object.

In the computational complexity theory, approximation algorithms for optimization problems have been extensively studied for many years [52]. Here, an approximation ratio, the multiplicative factor on how close the approximate solution is to the optimization solution, is a key factor to evaluate an approximation algorithm.

Therefore, such an approximation ratio (of the approximate complexity value over the precise complexity value) is a key factor for an approximation method of the proposed complexity definition. Since the proposed definition generally requires at least an exponential-time computation, a natural question is which approximation ratio is achieved if an approximation method is a polynomial-time algorithm. Similarly, how is the approximation ratio for a sub-exponential-time or exponential-time algorithm? We need both theoretical and practical analyses for the problem.

(3) Applications to Network Expressions

The proposed complexity definition can be applied to any form of network expression. Here, the framework and analysis studied in the natural complexity [9, 11, 27, 33, 34, 36, 37] should be beneficial for our complexity definition approach to network expressions.

(4) Evaluation Process

An advantage of our complexity definition for practical applications is that we can clarify the evaluation process, especially separating the phase for identifying the object, the phase for finding a compressed (short) oc-circuit to simulate the object, and the phase for obtaining the (approximate) complexity.

The evaluation process of our (approximate) complexity is as follows:

(a) Identifying an object
We can skip this phase if an object is given in a proper form. Otherwise, we must identify the object or describe it precisely in some model (see Section 6.3). Here, note that an (identified) object can be any form of expression, for example, network expressions or others, deterministic or probabilistic, and so on.

(b) Finding a short oc-circuit
Given an object, this phase finds a short oc-circuit to simulate the object as shorter as possible (see Sections 6.1 and 6.2).

(c) Obtaining the (approximate) complexity
The size of the oc-circuit obtained in the previous phase is the (approximate) value of the proposed complexity definition of the object. If it is an approximate value, the approximation ratio (pointer (2) in this section) may be evaluated heuristically or theoretically.

(5) Coding Theorem

Let X be a distribution over n bit strings, δ be a precision level $(0 \leq \delta < 1)$, $\mathscr{C}$ be an oc-circuit, and $OC(X, \delta)$ be the proposed organized complexity of X at δ (Section 5.2).

Let $m(X, \delta) := \sum_{X \overset{\delta}{\approx} Y \overset{R}{\longleftarrow} \mathscr{C}} 1/2^{|\mathscr{C}|}$.

Note that the binary representation of $\mathscr{C}$ is inherently prefix-free (i.e., the Kraft inequality [3] holds, $m(X,\delta) \leq 1$.)

We then conjecture the following inequality, which corresponds to the coding theorem of the Kolmogorov complexity.

$$\left|OC(X,\delta) + \log_2 m(X,\delta)\right| < O(1). \tag{23}$$

Note that the difference with the coding theorem of the Kolmogorov complexity $K(x)$ (Section 4.1 (1)) is as follows: (1) $OC(X,\delta)$ is computable, while $K(x)$ is incomputable; (2) $OC(X,\delta)$ includes the computational complexity for $\mathscr{C}$ to produce (simulate) the object X, while $K(x)$ does not include the computational complexity for a prefix universal Turing machine U to produce the object x; (3) the object X of $OC(X,\delta)$ is a distribution, while the object x of $K(x)$ is a deterministic string; and (4) an oc-circuit $\mathscr{C}$ with $X^{\delta} \approx Y \xleftarrow{R} \mathscr{C}$ is used to define $m(X,\delta)$ and $OC(X,\delta)$, while a program p with $x = U(p)$ is used for $m(x)$ and $K(x)$.

(6) A Problem in the Economy

An attractive but challenging application of our definition is to give the complexity of a country's economy in a single number [12]. If we can express a country's economy with an appropriate (probabilistic or deterministic) description of the economic activities, it would be possible to evaluate our approximate complexity definition of the description, which could be a single number for the *organized* complexity of the country's economy.

7. Conclusion

This paper quantitatively defined the organized complexity. The proposed definition for the first time simultaneously captured the three key features of organized complexity: descriptive, computational, and distributional features. It is computable, rigorously specified, and treated both probabilistic and deterministic forms of objects in a unified manner or seamlessly. As a result, it satisfied all of the criteria for organized complexity definitions introduced in this paper.

Organized complexity is an interdisciplinary concept straddling physics, cosmology, chemistry, biology, ecology, sociology, economy, informatics, and so forth. Thus, the proposed organized complexity definition would be a core notion in such interdisciplinary areas and offer some basis for tackling the problems posed in [2, 20] and Section 6.

Data Availability

No data sets were generated or analyzed during the study.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] W. Weaver, "Science and complexity," *American Scientist*, vol. 36, no. 4, pp. 536–544, 1948.

[2] C. H. Lineweaver, P. C. W. Davies, and M. Ruse, Eds., *Complexity and the Arrow of Time*, Cambridge University Press, Chennai, India, 2013.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Hoboken, NJ, USA, 1991.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[5] C. H. Bennett, "Logical depth and physical complexity," in *The Universal Turing Machine, A Half-Century Survey*, R Herken, Ed., Oxford University Press, Oxford, UK, 1988.

[6] M. Gell-Mann, "What is complexity," *Complexity*, vol. 1, no. 1, Article ID 6130010105, 1995.

[7] M. Gell-Mann and S. Lloyd, "Information measures, effective complexity, and total information," *Complexity*, vol. 2, no. 1, pp. 44–52, 1996.

[8] M. Gell-Mann and S. Lloyd, "Effective complexity," in *Nonextensive Entropy Interdisciplinary Applications*, M. Gell-Mann and C. Tsallis, Eds., The Santa Fe Institute, OUP USA, 2004.

[9] P. L. Gentili, "Why is Complexity Science valuable for reaching the goals of the UN 2030 Agenda?" *Rendiconti Lincei. Scienze Fisiche e Naturali*, vol. 32, no. 1, pp. 117–134, 2021.

[10] R. M. Hazen, P. L. Griffin, J. M. Carothers, and J. W. Szostak, "Functional information and the emergence of biocomplexity," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 8574–8581, 2007.

[11] M. Morzy, T. Kajdanowicz, and P. Kazienzko, "On Measuring the complexity of networks: Kolmogorov complexity versus entropy," *Complexity*, vol. 2017, Article ID 3250301, 12 pages, 2017.

[12] C. Sciarra, G. Chiarotti, L. Ridolfi, and F. Laio, "Reconciling contrasting views on economic complexity," *Nature Communications*, vol. 11, no. 1, p. 3352, 2020.

[13] S. Lloyd and H. Pagels, "Complexity as thermodynamic depth," *Annals of Physics*, vol. 188, no. 1, pp. 186–213, 1988.

[14] P. Grassberger, "Problems in quantifying self-generated complexity," *Helvetica Physica Acta*, vol. 62, pp. 489–511, 1989.

[15] J. P. Crutchfield and K. Young, "Inferring statistical complexity," *Physical Review Letters*, vol. 63, no. 2, pp. 105–108, 1989.

[16] J. P. Crutchfield, "The calculi of emergence: computation, dynamics and induction," *Physica D: Nonlinear Phenomena*, vol. 75, no. 1–3, pp. 11–54, 1994.

[17] J. P. Crutchfield and C. R. Shalizi, "Thermodynamic depth of causal states: objective complexity via minimal representations," *Physical Review*, vol. 59, no. 1, pp. 275–283, 1999.

[18] C. R. Shalizi and J. P. Crutchfield, "Computational mechanics: pattern and prediction, structure and simplicity," *Journal of Statistical Physics*, vol. 104, no. 3, pp. 817–879, 2001.

[19] J. Ladyman, J. Lambert, and K. Wiesner, "What is a complex system?" *European Journal for Philosophy of Science*, vol. 3, no. 1, pp. 33–67, 2013.

[20] J. Ladyman and K. Wiesner, *What Is a Complex System?*, Yale University Press, London, UK, 2020.

[21] M. Li and P. M. B. Vitanyi, "An introduction to Kolmogorov complexity and its applications," *Graduate Texts in Computer Science*, Springer-Verlag, New York, NY, USA, 2nd 4th edition, 2019.

[22] M. Sipser, *Introduction to the Theory of Computation*, Cengage Learning, San Francisco, CA, USA, 3rd edition, 2013.
[23] H. Vollmer, *Introduction to Circuit Complexity: A Uniform Approach*, Springer-Verlag, New York, NY, USA, 1999.
[24] H. Akaike, *Information Theory and an Extension of the Maximum Likelihood Principle, Proceedings of the 2nd International Symposium on Information Theory*, B. N. Petrov and F. Caski, Eds., Akadimiai Kiado, Budapest, Hungary, 1973.
[25] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 629–636, 1984.
[26] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.
[27] J. Cong and H. Liu, "Approaching human language with complex networks," *Physics of Life Reviews*, vol. 11, no. 4, pp. 598–618, 2014.
[28] G. J. Chaitin, "On the length of programs for computing finite binary sequences," *Journal of the ACM*, vol. 16, no. 1, pp. 145–159, 1969.
[29] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.
[30] R. J. Solomonoff, "A formal theory of inductive inference, part 1 and part 2," *Information and Control*, vol. 7, no. 1–22, pp. 224–254, 1964.
[31] L. Antunes, L. Fortnow, D. van Melkebeek, and N. V. Vinodchandran, "Computational depth: concept and applications," *Theoretical Computer Science*, vol. 354, no. 3, pp. 391–404, 2006.
[32] L. Levin, "Universal search problems," *Problems of Information Transmission*, vol. 9, pp. 265-266, 1973.
[33] D. R. Farine and H. Whitehead, "Constructing, conducting and interpreting animal social network analysis," *Journal of Animal Ecology*, vol. 84, no. 5, pp. 1144–1163, 2015.
[34] P. L. Gentili, *Untangling Complex Systems: A Grand challenge for Science*, CRC Press, Boca Raton, FL, USA, 2018.
[35] P. M. Kappeler, "A framework for studying social complexity," *Behavioral Ecology and Sociobiology*, vol. 73, no. 1, p. 13, 2019.
[36] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.
[37] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Observability of complex systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 7, pp. 2460–2465, 2013.
[38] C. Adami, C. Ofria, and T. C. Collier, "Evolution of biological complexity," *Proceedings of the National Academy of Sciences*, vol. 97, no. 9, pp. 4463–4468, 2000.
[39] C. Faloutsos and V. Megalooikonomou, "On data mining, compression, and Kolmogorov complexity," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 3–20, 2007.
[40] D. W. McShea, "Perspective metazoan complexity and evolution: is there a trend?" *Evolution*, vol. 50, no. 2, pp. 477–492, 1996.
[41] J. W. Szostak, "Functional information: molecular messages," *Nature*, vol. 423, no. 6941, 2003.
[42] H. B. Enderton, *A Mathematical Introduction to Logic S*, Academic Press, Cambridge, MA, USA, 2nd ed. edition, 2011.
[43] O. Goldreich, *The Foundations of Cryptography*, Cambridge University Press, Chennai, India, 2001.
[44] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Chennai, India, 2000.
[45] J. Machta, "Natural complexity, computational complexity and depth," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 21, no. 3, Article ID 037111, 2011.
[46] A. Kulkarni and S. Bush, "Detecting distributed denial-of-service attacks using Kolmogorov complexity metrics," *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 69–80, 2006.
[47] J. Schmidhuber, "Discovering neural nets with low Kolmogorov complexity and high generalization capability," *Neural Networks*, vol. 10, no. 5, pp. 857–873, 1997.
[48] F. Soler-Toscano, H. Zenil, J. P. Delahaye, and N. Gauvrit, "Calculating Kolmogorov complexity from the output frequency distributions of small turing machines," *PLoS One*, 2014.
[49] A. Lempel and J. Ziv, "Compression of two-dimensional data," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 2–8, 1986.
[50] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.
[51] P. M. B. Vitányi, "How incomputable is Kolmogorov complexity?" *Entropy*, vol. 22, no. 4, p. 408, 2020.
[52] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, Chennai, India, 2010.