A Theory of Machine Learning

Jinsook Kim[*]
Underwood International College
Yonsei University
Seoul, Korea 03722
jki76364@gmail.com

Jinho Kang
Emeritus, Seoul National University
Seoul, Korea 08826
jhkang@snu.ac.kr

Abstract

We critically review three major theories of machine learning and provide a new theory according to which machines learn a function when the machines successfully compute it. We show that this theory challenges common assumptions in the statistical and the computational learning theories, for it implies that learning true probabilities is equivalent neither to obtaining a correct calculation of the true probabilities nor to obtaining an almost-sure convergence to them. We also briefly discuss some case studies from natural language processing and macroeconomics from the perspective of the new theory.

1 Introduction

In this paper, we examine three major theories of machine learning. We will call them *the possible worlds theory*, *the recognition theory*, and *the operation theory*. Both the possible worlds theory and the recognition theory are based on what we will call the *epistemic* approach to machine learning, whereas the operation theory is based on what we will call the *behavioral* approach. We will prove that all three theories have important problems. We will then provide a new theory of machine learning according to which machines learn a function when machines *successfully* compute it. We will show that this theory challenges common assumptions in the statistical and the computational learning theories, for it implies that learning true probabilities is equivalent neither to obtaining a correct calculation of true probabilities nor to obtaining an almost-sure convergence to them. Lastly, we will discuss when machines can or cannot learn a probability function in the perspective of our new theory by considering two case studies, the first one from natural language processing and the second one from macroeconomics.

2 Two Theories of Machine Learning in the Epistemic Approach

The epistemic approach to machine learning emphasizes that learning is the phenomenon of *knowledge* acquisition. In order for machines to learn a function, they must acquire knowledge of it. Given that what machines do is essentially computational, we can say that machines *learn* when they acquire *knowledge* through a *computational* method. There are two influential theories of machine learning in the epistemic approach, the possible worlds theory and the recognition theory. Let us consider them in turn.

2.1 The Possible Worlds Theory

First, the possible worlds theory: two things are defined here, (1) *knowledge* and (2) the *process* through which the learning algorithm returns the final knowledge from the completed task. For

[*]Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Under review.

(1), knowledge is defined as *truth in all epistemically possible worlds*. Based on this definition of knowledge, process is defined as follows: extending some notions from Halpern et al. (1997, 2003), let us define a computer system or process by the set of possible runs. Here, a *run* is a description of the behavior of the system over time. Formally, a run is a function from time to state while each state encapsulates all the knowledge available to the system at that time. Now, this system may consist of either a single process or multiple processes. In the latter case, states can be further subdivided into local states and global states, and the system becomes a distributed system. It is worth noting here that time is discrete and does not need to be "real time." To distinguish time from"real time", we will call it a *step*. Now, except for the initial knowledge base given in the initial state s_1, all the pieces of knowledge afterwards are *internally* provided at each step while the algorithm is being executed. Thus, once any piece of knowledge is added internally to the state, an additional step is counted. Hence we can derive the following definition.

Definition 1 Machines **learn** when their *processes* are in the overall state $(s_1, \ldots, s_n)$, where s_n is the state at terminating step *n* and each state s_i encapsulates all pieces of *knowledge*, i.e. every truth in all epistemically possible worlds, that are available at each step i once their algorithms are executed.

The possible worlds theory provides an excellent formalism for reasoning about knowledge. However, it does not explicate what learning is because it ultimately fails to explicate what knowledge is. That is, considering DeRose (1991) and Stalnaker (2006), we derive the following theorem:

Theorem 1 The definition of knowledge in the possible worlds theory does not provide any substantial analysis of how to determine the truth in all *epistemically possible worlds* without relying on the very notion of knowledge.

If Theorem 1 holds, then the possible worlds theory cannot explicate what knowledge is in terms of truth in all epistemically possible worlds. Hence it cannot explicate what learning is either. Therefore, the possible worlds theory inevitably leads to the situation in which any knowledge-based programming must eventually rely on "knowledge" of the programming designers who design the program and set its state space.

2.2 The Recognition Theory

We can point out a similar problem in the second theory of machine leaning in the epistemic approach, namely the recognition theory, which was proposed in Valiant (1984). According to Valiant (1984), machine learning can be defined as follows.

Definition 2 Machines **learn** a concept if, given data, there exists a deduction procedure by which a correct *recognition algorithm* is derived for that concept.

Here, *algorithims* mean step-by-step instructions used in computing to correctly recognize the concept. So this theory is computational, which we will explain below. Just like knowledge, *recognition* here is another epistemic notion that needs to be explicated in order to explicate what learning is.

Rather than providing an explicit algorithm that would universally return correct recognition in every possible instance, the recognition theory in Valiant (1984, 2008) aims to control the error that machines fail to recognize a concept for given data. This is the point where the recognition theory is distinctive from the possible worlds theory: unlike the possible worlds theory, it does not aim to encode universal rules that can be applied to every possible case without fail. When errors are controlled arbitrarily small by increasing sample size at reasonable cost, machines are guaranteed to be robust with correct recognition. Note that the recognition algorithm is the Boolean circuit expressed by any concept function which is approximate to the true function, so that when the true concept function can be arbitrarily approximated by controlling error, the recognition algorithm can be deduced. What is "recognition," then? In the case of concept recognition, it is what machines obtain by approximating the true target concept function. What is the "true concept function," then? Machines do not need any explicit definition of the true concept function to recognize it because it would be enough if machines could approximate the function arbitrarily well. Technically, machines are guaranteed to control errors when errors come from the *i.i.d.* distribution. In the end, then, **recognition** is whatever springs up to machines under the *i.i.d.* assumption after executing some Boolean circuit by processing sufficient data.

The recognition theory is a great breakthrough given the problem of induction that machines otherwise must face to learn universal rules from finite samples. However, the *i.i.d.* assumption here cannot be innocent. It is one thing that machines need some *i.i.d.* assumption to learn, while it is another that machines indeed satisfy such an assumption so that they learn. We thus derive the following theorem.

Theorem 2 To satisfy the *i.i.d.* assumption, the deducing recognition algorithm must rely on *learning* a certain identical distribution.

Theorem 2, however, implies that the definition of learning in the recognition theory is ultimately based on the very notion of learning. Hence the definition is circular and cannot explicate what learning is.

3 A New Theory of Machine Learning and Its Implications

We have shown how formal treatments of epistemic notions such as "knowledge" in the possible worlds theory and "recognition" in the recognition theory fail to explicate what learning is. Thus, our strategy to define "learning" will be rather indirect, without committing ourselves to what knowledge or recognition is. Note that the phenomenon of learning must be at least computational in its essence when achieved by machines. Therefore, we adopt the notion of computation to define what learning is in Definition 3. Inspired by the ideas of Turing (1936) and Church (1936), we require that machines be able to *effectively calculate* or *compute* a target function when machines can learn the function. Thus, once machines *learn* a function, there must *exist* some definite and explicit *instructions* that machines must follow to return the function, which is to capture the role of "machine learning" as a computational notion.

Definition 3 Machines **learn** any target function when they *succeed* in effectively calculating/computing the function if any, after processing possibly infinite amounts of data.

It should be noted that we added the notion of success in Definition 3, which is to capture the role of "learning" as an *epistemic* notion. The epistemic notion of machine learning requires two important components: if machines learn, (*i*) they are *indeed correct* most of the time, and (*ii*) they are *self-assured to be correct* most of the time. In other words, machines successfully compute a target function if and only if they compute it in a *reliable* and *doxastic* manner. As a result, we derive the following success criterion:

The Success Criterion for Machines to Learn a target function

(1) If machines achieve computational success by learning, what they compute in the end must be at least true to our world most of the time, which should be assured to the machines themselves.

If what they compute turns out to be wrong or the machines admit errors repeatedly too often out of infinite opportunities to learn, then their computation cannot be considered successful. In case of learning the true probability, we prove what exactly is meant by "most of the time" in the proof of Theorem 3. Furthermore, Kim (2024) proves in Corollary 4.37 that Success Criterion (1) is a sufficient condition for learning in the case of computing the true probabilities. Thus, although the notions of "computation" and "success" are informal in Definition 3, we can provide under what formal conditions machines can achieve such computational success once machines are under the specific task of learning a particular function.

Now that we define what "machine learning" is, let us discuss when machines can learn the true probabilities. Before doing so, let us first derive a necessary condition for machines to learn a true probability function.

In Hintikka (1962), the knowledge of a person i refers to the knowledge of that person i on any proposition A. Likewise, machine's learning of the true objective probability P here refers to the knowledge acquired by any machine on the probabilistic proposition A_p. If machines learn the true probability as α, then the probabilistic proposition A_p amounts to that the true objective probability P, if any, is what the very machines calculate as α. Here, we convert the non-propositional learning into propositional learning. But it deserves to emphasize that machines do not lose any information in the representational content of learning while converting the non-propositional structure into propositional one. (e.g. Peacocke (2006))

Now, just as a person i's knowledge on proposition A must satisfy the necessary condition that the person i's belief in A be true, machine learning of the true probability P must also satisfy the necessary condition that the representation of A_p of the machines be true. Note here that such a representation of A_p is true when what has been computed by the machines is indeed equal to the true probability P. Now, this computed probability function by machines is nothing more than the subjective probability of the machines. Therefore, a necessary condition for machine learning of true probability P requires that the subjective probability of machines is indeed in congruence with the true objective probability P. In short, if machines learn the true objective probability P, then the subjective probability Π of the machines is actually equal to the true probability P.

A Necessary Condition for Machines to Learn the True Probability

(2) If machines *learn* the true objective probability $P(A_{t+1}|ß_t)$, then $\Pi(A_{t+1}|ß_t) = P(A_{t+1}|ß_t)$

where $\Pi(A_{t+1}|ß_t)$ denotes the subjective probability of the machines at time t.

We assume, without loss of generality, that the event A_{t+1} is an elementary event, for simplicity. Thus, the event A_{t+1} is a singleton, i.e. $\{\omega_{t+1}\}$ where ω_{t+1} is a state in the sample space of the true probability.

It is worth emphasizing that learning implies obtaining a true fact by satisfying the truth condition that $\Pi(A_{t+1}|ß_t) = P(A_{t+1}|ß_t)$, but that the converse does *not* hold, which will be demonstrated shortly by constructing a counterexample. This is why the above is only a necessary but not sufficient condition for learning the true probability. In particular, we will prove that learning is not equivalent to any of the following three mathematical conditions, although such equivalence is commonly assumed in large literature.

3.1. Learning Is Not a Correct Calculation of True Probabilities

First, machine learning of true probabilities is not equivalent to a correct calculation of them by machines. That is, it is *not* necessarily the case that machines learn even when they return a correct calculation that $\Pi(A_{t+1}|ß_t) = P(A_{t+1}|ß_t)$. If machines return correct calculations of true probabilities, then machines must have obtained true facts about some objective uncertainties of our world. But not every obtainment of a true fact is an acquisition of *knowledge*, for machines may have *happened* to compute the true probability only by *luck*. Ever since Plato (2014), it has been accepted that merely an accidental coincidence with the truth by luck is not enough to count as knowledge. Since knowledge is not acquired through obtaining truth by luck, machines do not necessarily learn even if they obtain a true fact.

However, there is large literature whose common assumption is that (acquired) knowledge is equivalent to a correct calculation of true probability. (e.g. Blume and Easley (2006), Sandroni (2000), Cogley and Sargent (2008)) In particular, this assumption justifies some game settings where certain winning strategies in games are equivalent to learning strategies in the theory of learning through games. (e.g. Foster and Vohra (1993), Nisan et al. (2007)) But such an assumption is challenged by the following Theorem 4.

Theorem 3 If machines learn the true probability $P(A_{t+1}|ß_t)$ as α, then $P\ (p_k \to \alpha) = 1$ with $k \to \infty$ where k is the number of days in the test set,

$$p_k = (\sum_{t=0}^{k-1} \xi_{t+1})^{-1} \cdot (\sum_{t=0}^{k-1} \xi_{t+1} \cdot 1_{\{A_{t+1}\}}) \text{ with } 1_{\{A_{t+1}\}} \text{ being the indicator of } A_{t+1} \text{ and}$$

$$\xi_{t+1} := \begin{cases} 1 & \text{if } \Pi(A_{t+1}|ß_t) = \alpha = P(A_{t+1}|ß_t) \\ 0 & \text{if } \Pi(A_{t+1}|ß_t) = \alpha \neq P(A_{t+1}|ß_t) \end{cases}$$

In general, a test set denotes a set whose elements are collected by certain selection criterion to test calibration property in Dawid (1982). Here, our selection criterion to construct a test set is a *correct* probability as α.

Theorem 4 If $P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n$ for some $n < \infty$ along the stochastic path, then $P(A_{t+1}|ß_t) = \alpha$ at some $t^* < n$ is not equivalent to *learning* the true probability at t^*.

For example, we can think of the situation in Theorem 4 as an analogue to the following: even a broken clock is correct twice a day. But machines cannot be said to learn what time it is from the broken clock even when they happen to return the correct time, say 2pm, by processing the data from this clock luckily at the right timing, namely at 2pm. Therefore, “learning” should be distinguished from “obtaining the true fact”.

3.2. Learning Is Not an Almost-Surely Correct Calculation of True probabilities

Second, machine learning of true probabilities is not equivalent to an almost-surely correct calculation of them by machines. That is, it is *not* necessarily the case that machines learn even when they return an almost-surely correct calculation.

Theorem 5 If $P(\ P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n\ ) > 0$ for some $n < \infty$ along the stochastic path of the test set, then $P(\ P(A_{t+1}|ß_t) = \alpha\ ) = 1$ at some $t^* < n$ is not equivalent to *learning* the true probability at t^*.

Remark 1 We provide more detailed explanation on Theorem 5 with an example in the Appendix B.

However, there is large literature in which the condition of the true probability $P-$one is regarded as the definition or the necessary and sufficient condition for learning/knowledge. For example, Aumann and Brandenburger (1995) stipulates that the player i is said to *know* an event at state s if he/she ascribes probability 1 to that event at s. It is not clear whether or not Aumann and Brandenburger (1995) meant by "probability" here the true objective probability. But since the condition of *subjective* probability being 1 is usually interpreted as *full confidence* but “knowing” is not necessarily the same as “being fully confident/self-assured”, we assume that the probability in this context should be the true objective one, not just the subjective one. Let us then discuss how Aumann and Brandenburger (1995) can be understood in our context.

Note that knowledge in the context of Aumann and Brandenburger (1995) is often knowledge about the player *himself / herself*. For example, Lemma 2.6 in Aumann and Brandenburger (1995) says that player i knows *that he/she attributes probability π to an event E* if and only if he/she indeed attributes probability π to E with the true probability $P-$ one. Aumann and Brandenburger (1995) establishes this Lemma 2.6, showing that if the player i indeed attributes this, then with the true probability $P-$ one, the player i does so. Now, note that whether the player attributes a certain (subjective) probability π to E or not is an *internal* event that occurs to the player himself / herself, but that whether such an internal event occurs or not is a matter of fact. In any case, the definition of knowledge in terms of obtaining a true fact in Aumann and Brandenburger (1995) conflicts with our general argument that not every obtainment of a true fact is knowledge. In other words, Lemma 2.6 conflicts with our general argument that satisfying the truth condition with true probability $P-$ one is just a necessary condition for learning/knowing, not a sufficient one. We thus conclude that the way Aumann and Brandenburger (1995) uses “knowledge” is erroneous *in general*.

However, we should point out that humans’ *self-knowledge* has some privileged status. (e.g. Gertler (2021, 2011)) That is, humans’ true beliefs about themselves are *guaranteed* to be true in many cases so that they do amount to knowledge. So if humans are guaranteed to be true about themselves with true probability $P-$ one, they are not only fully confident/self-assured about themselves but also know about themselves. Hence human beings come to know that an internal event occurs to themselves whenever what is internally occurring to them holds with true probability $P-$ one. But more discussions of the privileged status of human self-knowledge are beyond the scope of this paper, so we will not investigate this issue in Aumann and Brandenburger (1995) any further.

3.3. Learning Is Not an Almost-Sure Convergence to True Probabilities

Third, machine learning of true probabilities is not equivalent to obtaining almost-sure convergence to them by machines.

Theorem 6 Even if machines correctly calculate that $P(\lim_{t\to\infty} P(A_{t+1}|ß_t) = \alpha) = 1$, this is not equivalent to that machines learn true probabilities $P(A_{t+1}|ß_t)$ as α.

From the Success Criterion (1) and the necessary condition (2), we further derive that “machine learning” implies an almost sure convergence of the true probability point-wise, but not vice versa. As in the previous two sections, “learning” should be distinguished from obtaining “ the true fact” that $P(A_{t+1}|ß_t)$ converges to α with true probability P-one. Note that Kim (2024) explains by Lemma 4.31 why machines cannot learn even if they satisfy the condition of the almost-sure convergence: if machines are not self-assured of whether the true probabilities remain observable most of the time even when the true probabilities do remain so, machines cannot learn them. In other words, Corollary 4.37 in Kim (2024) shows that even if machines satisfy the truth condition by calculating the true probability most of the time, they cannot be said to learn when they do not satisfy the second part of the Success Criterion that they must be self-assured of being correct most of the time when they indeed are correct that often.

Now, note that while constructing a counter-example in the proof of Theorem 6, we show the following: consider an identically distributed true process $\mathbb{X}^*$, a sequence of random variables $\{X_t^*\}_{t=k}^{\infty}$ whose underlying common probability is P. Then, the data exclusively from this process $\mathbb{X}^*$ cannot be *available* to machines point-wise, so that the machines cannot learn the common true probability distribution of the process $\mathbb{X}^*$. Machines cannot learn it because machines cannot determine the selection criterion ξ_{t_k} of collecting data exclusively from X_t^*'s unless machines have been self-assured of what the true underlying probabilities are. As Theorem 4.36 and Corollary 4.37 in Kim (2024) show, machines can be self-assured of what the underlying true probabilities are only when the true probabilities are directly observable because machines then can collect data from a *given* population where an identical distribution is already endowed.

However, it has been widely accepted (e.g. Vapnik (2000)) that almost sure convergence condition (e.g. Glivenko-Cantelli theorem) constitutes the definition of "learning." But at least with regard to a true probability function, it is one thing that the functions of a sequence of random variables *mathematically converge* to a certain limit almost surely, while it is another thing that machines can *learn* the true function as such limit by processing the *relevant* data.

4 The Operation Theory of Machine Learning in the Behavioral Approach

Let us now consider the operation theory of machine learning, which is based on the behavioral approach, while discussing some fundamental limits on deriving practical implications from the theoretical results on learning true probabilities. Note that one way to prove a practical significance of any theoretical result on learning is to show that such a theoretical result implies some observational significance which can be quantitatively measured by some performance of real learning machines. According to Mitchell (1997), this idea succinctly constitutes the very definition of machine learning.

Definition 4 A computer program **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by performance P, *improves* with experience E. (Bengio et al. (2016))

Unfortunately, however, this definition cannot hold for the case of learning *the true probability*, as long as the tasks of machines are in the form of standard models of optimal decision under uncertainty. For there exists an *observational equivalence* in machine behaviors under various probability measures.

Theorem 7 Suppose that machines accomplish their tasks in the form of solving the standard optimization problems under uncertainty as follows:

$$\max \int f(x)\, d\nu = \int f(x) h\, d\mu \qquad \text{subject to } x \in \partial.$$

where f is any gain function and h is the *Radon-Nikodym* derivative of ν with respect to μ.

Then, there exists observational equivalence in the optimal behaviors of machines under various probability measures, μ and ν. Furthermore, suppose that the true probability is unique, that is, it is ν but not μ. Then machines cannot learn the true probability under Definition 4.

Therefore, once we try to prove the practical significance of a theoretical result, we return to where we prove the impossibility of learning the true probability.

We have called the theory in Mitchell (1997) "the operation theory" in the sense that learning is defined by the operation of the subject to whom the learning occurs. The general idea behind the operation theory is that we can define some non-observational concepts such as "learning", a psychological concept, in the context of science only when we have a method of measurement for those concepts. This measurement is usually obtained by the outcomes of some specified manipulation through which the target operation is obtained in each case. The operation theory entitles scientists the right to do some empirical tests for the hypotheses formulated by non-observational terms. However, if there are always observational equivalences among the outcomes of multiple operations no matter what kind of manipulations are adopted, we cannot properly measure the outcomes of the target operation and, accordingly, cannot define those non-observational terms through any operation. Hence we conclude that the operation theory in general fails to explicate what learning true probabilities is, just as the possible worlds theory and the recognition theory fail to do so.

5 Some Practical Implications for Machine Learning Algorithms

However, once we abandon the idea of directly estimating the true probabilities by measuring machine performances which are quantitatively distinguished by learning different probabilities but let machines aim to directly estimate the true probabilities, there can be some possibilities to discuss practical implications from learning the true probabilities.

Theorem 8 Let the machines aim at directly estimating the true probability itself. Then, there exists a parametric deep learning model which does not suffer from the problem of observational equivalence in Theorem 7.

Theorem 4.36 in Kim (2024) proves under Definition 3 that ideal machines can learn the true probabilities if and only if the probabilities are *directly observable* by the machines. Based on this Theorem 4.36, we will discuss two case studies, the first being the one in which real machines can learn true probabilities while the second being the one in which no real machines can do so. From these case studies, we can discuss what kind of practical implications for real machine learning algorithm we can derive from the theoretical results on ideal machine learning algorithm in Kim (2024).

Case 1. A Learnable Case from Natural Language Processing

Deep learning has produced some astonishing results in speech recognition. (*e.g.* Hinton et al. (2012)) This remarkable phenomenon can be chiefly explained by the fact that the true probability of obtaining any given sequence of words is indeed learnable by machines. Thus, as a simple example, let us consider the case of machine learning algorithms for a language model such as the N-gram model in the natural language processing. (*e.g.* Jurafsky and Martin (2008)) In particular, let us focus on simple (unsmoothed) N-grams as a baseline to understand the direct observation on the true probabilities in principle. For simplicity, we assume that the language we consider is English.

According to Jurafsky and Martin (2008), N-grams are essential in any task in which machines must identify words in noisy, ambiguous *inputs*. Here, what deserves to note is the expression "input." For example, in speech recognition tasks, machines are given the inputs in the form of ambiguous speech to be recognized. Then, once those inputs were ever used for actual speech and provided as what is needed to be further recognized by machines, there must exist some sort of *true population* which contains whatever is the unambiguous form of the given input. This true population is what we call an *ideal corpus* whose definition is provided below.

Definition 5 At any given time t_0, let us consider the population W^*, the set of all words and sequences of words that have been used by any subject, such as real persons, for certain periods up to that time t_0 to *effectively* communicate each other in our world. We define the set W^* of all such words and sequences of words by the *ideal corpus*.

For instance, Brown and Switchboard are some well-known corpora in practice. However, the corpus in Definition 5 is ideal in the sense that it includes all the relevant words and sequences of words that have ever been circulated within any community in this world for certain periods up to any given time. It is beyond any simple online collection of words and sentences. Here, words may also represent various ranges of things such as numbers, punctuation marks, fragments, filled pauses, tabs or spaces, etc.

Thus, for any *given* sequence of words S, what is the probability of obtaining S? This is the question that we would like to address here. This question, however, is different from what we would call the Markov question by following Markov (1913): What is the probability of obtaining any *arbitrary* sequence of words? In the Markov question, we must consider a potentially infinite sequence of strings, and thus there does not exist any maximum size such as the maximum of word lengths or the maximum of size of the sequence of words in $W.^*$ It should be noted that the distinction between these two questions is particularly important because it implies that the way machines learn how to recognize sentences in a natural language is quite different from the way humans learn it, given that the ideal corpus W^* in the simple N-gram model contains only a finite number of sentences whereas humans are supposedly able to learn how to recognize potentially infinite number of sentences in the natural language.

Remark 2 We provide more detailed explanation on the implications of the distinction between these two questions in the Appendix B.

Now, for any arbitrary finite number N $\in \mathbb{N}$, the unsmoothed N-gram models forecast the probability of obtaining the next word following the previous N-1 words for any *given* sentence. This probability is in the form of conditional probability, which is connected to the joint probability of a sequence of words under the chain rule. The conditional probability then is defined to be measured by counting the frequencies in the ideal corpus in the following way:

$P(w_n|w_1^{n-1}) = \frac{C(w_n, w_{n-1}, \ldots, w_1)}{C(w_{n-1}, \ldots, w_1)}$ where w_1^{n-1} represents the sequence of words $\{w_1, \ldots, w_{n-1}\}$ and $C(w_1^n)$ counts the number of frequencies that w_1^n occurs in the *ideal corpus*.

Then, by chain rule, $P(S) = P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1})$ with $P(w_1^0) = 1$ for any given S whose length is n.

Now, following Kim (2024), let us define when a true probability is directly observable and then prove that $P(S)$ is directly observable and thus learnable by machines.

Definition 6 Let us consider a set W that consists of the sequence of events A_{t+1}'s, $\{A_{t+1}\}_{t=0}^{k-1}$ with k potentially infinite. The set W is then defined to be a **population** with k number of elements, when this set W is assumed to have a certain attribute of interest, and so an indicator variable $1_{\{A_{t+1}\}}$ is assigned to each event A_{t+1} where $1_{\{A_{t+1}\}}$ has a value 1 or 0 depending on whether the event A_{t+1} satisfies such an attribute or not, once the set W is collected. The empirical distribution of the population W with respect to the given attribute is $\frac{1}{k}\sum_{t=0}^{k-1} 1_{\{A_{t+1}\}}$.

Definition 7 Machines **directly observe** $P(A_{t+1}|ß_t)$ from the *true population* W at t^* if the following two conditions are satisfied: (*i*) a true population W is in principle *available* to the machines. (*ii*) machines effectively calculate the *empirical distribution* of the population W with respect to the given attribute, which is the true probability distribution of the event A_{t+1}.

Now, in case where the sequence $\{A_{t+1}\}_{t=0}^{k-1}$ is a time-series, Definition 7 means that $\Pi(A_{t^*+1}|ß_{t^*}) = \frac{1}{k}\sum_{t=0}^{k-1} 1_{\{A_{t+1}\}} = P(A_{t^*+1}|ß_{t^*})$ with $k = t^*$. Thus, if t^* goes to infinity, then the directly observable true probability becomes the limiting relative frequency, the representative objective true probability.

Remark 3 We provide more detailed explanation on Definition 7 with an example in the Appendix B.

Theorem 9 At any given time t_0, there exists the ideal corpus W^* with a finite number of words and finite number of sequence of words that have been used by humans up to t_0. Then, for any given sentence S, the true probability of obtaining S is effectively calculable from this corpus W^* by machines. Furthermore, this true probability $P(S)$ is learnable by machines.

In practice, facing various technological limits, real machines accomplish this task of learning the true probability by approximating through various methods such as maximum likelihood or some smoothing methods. Perhaps, real machines may approximate it by some other clustering algorithms to generalize the word N-gram to the class N-gram model while avoiding its distance dependency. No matter what approximation methods machines use, however, it is pertinent to show here that such a practical approximation can work well in principle with this case of directly observable probability, while it cannot work well in the other cases of not directly observable probability. We leave this issue of learning and approximation as a future research topic.

Case 2. An Unlearnable Case from Macroeconomics and Finance

Unlike what we have seen in Case 1, probability is not directly observable and so unlearnable in many cases. Recall by Definition 7 that machines can directly observe the true probability at least when there exists a true population available to machines. However, there is no population available to machines for r-star in macroeconomics and finance because r-star is a model-based estimate, not actual data, as large body of economics literature pointed out. (e.g. Krugman (2023)) Therefore, in this unlearnable case, machines usually try to learn the true probability *indirectly*, say by inferring it from other relevant observable variables while further assuming some system of the observation and the state equations that include unobservable variables with possibly known state-space. (e.g. Kalman Filter in Laubach and Williams (2003))

Now, we claim that no machines can estimate r-star because machines cannot learn the true probability of r-star nor approximate its true probability by learning the error/distance function of r-star. First, let us discuss some basic notations to define what the distance function is: let $(\Xi, \mathbf{ß})$ be a measurable space, where **ß** is a $\sigma-$algebra generated by random vector X in Ξ. Now, call Ξ the *sample* space of X and then fix a space Ψ, called the *value* space of Y. If there indeed exists a true function f^* whose learning is a given task to machines, then a learning rule is any sequence of measurable functions $f_{n,m} : \Xi^m \times \Psi^n \times \Xi \to \Psi$ for $n, m \in \mathbb{N} \cup \{0\}$ where the domain of $f_{n,m}$ is the set of the following three things, (i) the initial training samples of labeled data $(X_1, Y_1), \ldots, (X_n, Y_n)$ and (ii) the unlabeled data $X_{n+1}, \ldots, X_m$, and (iii) the random variable X_{m+1} for prediction, given that $m > n$. Then, the output of $f_{n,m}$ is the prediction Y of X_{m+1} based on the given data. Thus, for example, if $m = n$ with $n > 0$, then this learning rule is tied to supervised learning, while it is tied to semi-supervised learning with $m > n$ and $n > 0$ and tied to unsupervised learning with $m > n$ but $n = 0$.

Now, let us define a function $\ell : \Psi^2 \to [0, \infty)$, called the *distance function* such that ℓ ($f_{n,m}(\{X_i\}_{i=1}^m,\, f^*(\{X_i\}_{i=1}^n),\, \{X_i\}_{i=m+1}^\infty),\, f^*(\{X_i\}_{i=1}^\infty))$. In statistical learning theory, this distance function is usually called a loss function with $\ell\,(y_1, y_2) = \ell\,(y_2, y_1)$, which is connected to the notion of risk when the data come from **i.i.d.**. Here, we do not require the distance function to be symmetric, because some distance function which measures the "distance" between the *probability* functions is not a metric. For example, the function ℓ here can include the Kullback-Leibler divergence. In general, this function ℓ is supposed to measure the distance between any two functions from $i)$ the predicted function $f_{n,m}$ for the sequence of future random vector $\{X_i\}_{i=m+1}^\infty$ which is obtained from the learning rule with the initial training samples and from $ii)$ its true function f^* for $\{X_i\}_{i=m+1}^\infty$.

Theorem 10 Suppose that machines cannot learn the true probability function P as f^*_∞ even after processing infinitely many training samples $\{X_i\}_{i=1}^\infty$. Then, the machines cannot evaluate any $f_{n,m}$, which is a learning rule for any $n, m < \infty$, as a good (or bad) approximation to the true probability f^*_∞.

Now that machines cannot obtain any good or bad approximation to the true probability of r-star because machines cannot learn its true probability, we conclude that machines cannot estimate r-star. We leave as future research topic what would be the alternative method of forecasting the evolution of macro-economy as time passes, when it is not possible to learn or approximate the true probability of such an important variable in the model.

6 Conclusion

We close our paper with a few more remarks on how various theories of machine learning fare in the learnable case of Section 5 and the significance of the epistemic approach to machine learning.

First, we pointed out in Section 2 that the possible worlds theory of machine learning must rely on the knowledge of the program designer while setting the state space, and that the recognition theory bears the burden of proof that the distributional assumption is indeed satisfied. It seems hard to incorporate knowledge of the program designer into the system of the possible worlds theory unless it is first defined what knowledge is. Also, it seems difficult to show that the identical distribution assumption is satisfied in the system of the recognition theory unless it is first defined what learning is. In the cases of the directly observable probability, however, the true probability can be *a priori* defined by the empirical distribution of the *true population.* Since the target attribute of this population sets the state space of the true probability, it resolves the problem of incorporating knowledge of the program designer in the possible worlds theory. Also, the true population shares the common identical distribution, so it satisfies the assumption of identical distribution in the recognition theory.

Second, we emphasize that, in this paper, we have focused on the notion of "machine learning" which is not just computational but also epistemic, a counterpart to "human learning." We have focused on this epistemic notion of machine learning because we particularly mean by "machines" some artifacts which perform *human-level intelligent* behaviors. As long as we aim to contribute to the field of Artificial Intelligence through machine learning algorithms, machines must be such intelligent "learners."

References

Aumann, R. and A. Brandenburger (1995). Epistemic conditions for nash equilibrium. *Econometrica 63*(5), 1161–1180.

Bengio, Y., I. Goodfellow, and A. Courville (2016). *Deep Learning*. The MIT Press.

Blume, L. and D. Easley (2006). If you're so smart, why aren't you rich? belief selection in complete and incomplete markets. *Econometrica Vol. 74*, 929–966.

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics Vol. 58*(2), 345–363.

Cogley, T. and T. Sargent (2008). The market price of risk and the equity premium: A legacy of the great depression. *Journal of Monetary Economics Vol. 55*, 454–476.

Dawid, P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association 77*(379), 604–613.

DeRose, K. (1991). Epistemic possibilities. *The Philosophical Review 100*(4), 581–605.

Durrett, R. (2019). *Probability: Theory and Examples*. Cambridge: Cambridge University Press.

Foster, D. and R. Vohra (1993). Asymptotic calibration. *Biometrika 85*(2), 379–390.

Gertler, B. (2011). *Self-Knowledge*. London: Routledge.

Gertler, B. (2021). Self-knowledge. *Stanford Encyclopedia of Philosophy by Edward N. Zalta (ed.)*.

Halpern, J. and R. Fagin (1994). Reasoning about knowledge and probability. *Journal of the Association for Computing Machinery Vol 41*(2), 340–367.

Halpern, J., R. Fagin, and Y. Moses (1997). Knowledge-based programs. *Distributed Computing 10*(4), 199–225.

Halpern, J., R. Fagin, Y. Moses, and M. Vardi (2003). *Reasoning About Knowledge*. Cambridge: MIT Press.

Hintikka, J. (1962). *Knowledge and Belief.* Ithaca: Cornell University Press.

Hinton, G., , L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine 29*, 82–97.

Jurafsky, D. and J. H. Martin (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

Kim, J. (2024). Can machines learn true probabilities? *International Conference on Machine Learning PMLR 235*.

Krugman, P. (2023). Wonking out: Are high interest rates the new normal? *New York Times*.

Laubach, T. and J. Williams (2003). Measuring the natural rate of interest. *Review of Economics and Statistics 85*(4), 1063â€“1070.

Markov, A. (1913). Example of a statistical investigation of the text of â€œEugene Oneginâ€ illustrating the dependence between samples in chain. *Izvistia Imperatorskoi Akademii Nauk English Translation by Morris Halle*, 153–162.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Nisan, N., T. Roughgarden, E. Tardos, and V. V. Vazirani (2007). *Algorithmic Game Theory*. New York: Cambridge University Press.

Peacocke, C. (2006). *The Realm of Reason*. Clarendon Press.

Plato (2014). *Theaetetus*. Translated by McDowell. Oxford: Oxford University Press.

Sandroni, A. (2000). Do markets favor agents able to make accurate predictions? *Econometrica Vol. 68*(6), 1303–1341.

Stalnaker, R. (2006). On logics of knowledge and belief. *Philosophical Studies 128*, 169–199.

Turing, A. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society 42*, 230–265.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM 27, Nov.*, 1134–1142.

Valiant, L. (2008). Knowledge infusion: In pursuit of robustness in artificial intelligence. *Proc 28th Conference on Foundations of Software Technology and Theoretical Computer Science*, 415–422.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer.

Appendix

A Technical Proofs for Theorems

Proof of Theorem 1 Let an interpreted system $\mathcal{I}$ consists of a pair $(\mathcal{R}, \pi)$ where $\mathcal{R}$ is an system over a set $\mathcal{G}$ of states and π is an interpretation over $\mathcal{G}$, which assigns truth values. Now that knowledge is defined as truth in all epistemically possible worlds, agent i knows φ if φ is true at all points that agent i considers epistemically possible. Thus, we define

(1) $(\mathcal{I}, r, m) \models \mathcal{K}_i\varphi$ if and only if $(\mathcal{I}, r', m') \models \varphi$ for all (r', m') such that $(r, m) \sim_i (r', m')$
where $\sim_i$ represents an epistemically possible relation to agent i.

However, due to DeRose (1991),

(2) $(\mathcal{I}, r', m') \models \varphi$ if and only if $(\mathcal{I}, r, m) \not\models \mathcal{K}_i\neg\varphi$

Therefore, we conclude that $(\mathcal{I}, r, m) \models \mathcal{K}_i\varphi$ in (1) cannot be determined without relying on whether $(\mathcal{I}, r, m) \not\models \mathcal{K}_i\neg\varphi$ in (2), which shows that the notion of knowledge $\mathcal{K}_i$ in (1) by the agent i relies on the very notion of knowledge $\mathcal{K}_i$ in (2) by the very agent i.

In the possible worlds theory, *knowledge* is standardly represented in terms of the possible worlds framework so that it is analyzed as *truth in all epistemically possible worlds*. (e.g. Hintikka (1962), Halpern and Fagin (1994), Stalnaker (2006)). Here *epistemically* possible worlds must be distinguished from *counterfactually* possible worlds. An epistemically possible world for an agent is the one that is compatible with the knowledge possessed by the agent, whereas a counterfactually possible world is any possible world that is compatible with all logical and mathematical truths.

For example, let us suppose that Jane knows that it is Mount Everest that is the highest mountain in the world. Then there is no epistemically possible world for Jane, in which Mount Kilimanjaro is the highest mountain, because such a world is not compatible with what she knows. On the other hand, there are counterfactually possible worlds in which Kilimanjaro is the highest mountain, for Kilimanjaro's being so is clearly compatible with all logical and mathematical truths. To put it another way, it is certainly the case that Kilimanjaro could have been the highest mountain so that we can easily imagine it without any contradiction. Sentences of the form 'It is possible that P', in which P is in the indicative mood, typically express epistemic possibilities, whereas sentences of the same form in which P is in the subjunctive mood typically express counterfactual possibilities. (e.g. DeRose (1991)) Thus the sentence 'It is possible that Kilimanjaro is the highest mountain' typically expresses an epistemic possibility, whereas the sentence 'It is possible that Kilimanjaro should have been the highest mountain" typically expresses a counterfactual possibility.

The above distinction between epistemically and counterfactually possible worlds shows that the analysis of knowledge in the possible worlds approach as truth in all epistemically possible worlds cannot be a reductive analysis, for we must use the very notion of knowledge in order to characterize the notion of epistemically possible worlds. $Q.E.D$

Proof of Theorem 2 Let X be a set which is called the *instance space*. Then, a concept over X is a subset $c \subseteq X$. Formally, a concept c over X is a boolean mapping $c : X \to \{0, 1\}$, with $c(x) = 1$ if $x \in c$ and $c(x) = 0$ otherwise. Now, let $\mathcal{D}$ be any fixed probability distribution over X. Then, provided that h is any hypothesis concept over X, the error between h and the target concept c can be defined as:

$$error(h) = Pr_{x \in \mathcal{D}}[h(x) \neq c(x)]$$

Then, a **recognition** algorithm L is deduced for the concept c when, with probability at least $1-\delta$, the algorithm L returns a hypothesis concept h that satisfies $error(h) < \epsilon$ for every sufficiently small ϵ and δ and every probability distribution $\mathcal{D}$. Thus, no matter what distribution x is drawn from, x must be sampled from an *identical* distribution. However, as we show by the counterexample in the proof of Theorem 6, collecting data from a fixed identical distribution can amount to "learning" the distribution itself. Therefore, in this case, deducing recognition algorithm must rely on the very notion of "learning." $Q.E.D.$

Proof of Theorem 3 For infinitely many t's when $P(A_{t+1}|ß_t)$ stays the same as α by Lemma 4.5 in Kim (2024), suppose that machines learn this $P(A_{t+1}|ß_t)$ as α at some time t_0. Then, by the Success Criterion (1), $\Pi(A_{t_k+1}|ß_{t_k}) = \alpha = P(A_{t_k+1}|ß_{t_k})$ at least infinitely often out of infinite opportunities at t's to learn. (We prove in the below what we mean exactly by "most of the time" in the Success Criterion (1). Here we tentatively mean "at least $i.o.$" by it because machines are otherwise wrong too often to learn given the Success Criterion (1).)

Thus we can construct a test set which consists of the subsequence of $\Pi(A_{t_k+1}|ß_{t_k})$ which is equal to $P(A_{t_k+1}|ß_{t_k})$ for those infinitely many t_k's. Let $\xi_{t_k+1} = 1$ if and only if $\Pi(A_{t_k+1}|ß_{t_k}) = P(A_{t_k+1}|ß_{t_k}) = \alpha$. Note that ξ_{t_k+1} is $ß_{t_k}-$measurable, because machine forecasting α occurs at time t_k. Then, by Theorem 4.1 in Kim (2024) while replacing Π by P, with true probability $P-$one, $p_k - \alpha = (\sum_{j=0}^{k-1} \xi_{t_j+1})^{-1} \cdot \sum_{j=0}^{k-1} \xi_{t_j+1}(Y_{t_j+1} - \alpha) \to 0$, as $k \to \infty$ where P is defined over $ß_\infty = \bigvee_{k=0}^{\infty} ß_{t_k}$ and $ß_{t_k}$ is denoted by the totality of *true facts* up to day t_k. Thus if machines learn the true probability $P(A_{t+1}|ß_t)$, then machines satisfy the calibration property by Dawid (1982) along the stochastic path of the test set where machines are correct at least $i.o.$ out of infinite opportunities to learn.

Now, by Lemma 4.10 and Theorem 4.17 in Kim (2024), machines cannot satisfy the calibration property when the test set is constructed by the selection criterion of an assessed probability α if $P(P(A_{t+1}|ß_t) \neq \alpha$ at least $i.o.) > 0$. Therefore, in order to learn, the machines must return the correct calculations except a finite number of times out of infinite opportunities to learn. Thus, "most of the time" in the Success Criterion (1) should be "all but finitely often out of infinite opportunities to learn," which means that machines must be correct not just infinitely often while being wrong that often. $Q.E.D.$

Proof of Theorem 4 For any machine forecast $\alpha \in \mathbb{R}[0, 1]$, suppose that $P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n$ for some $n < \infty$ along the stochastic path but that $P(A_{t+1}|ß_t) = \alpha$ at some $t^* < n$. Then, for $\alpha = 1$, $P(P(A_{t+1}|ß_t) \neq \alpha$ at least $i.o.) > 0$ for some event A_{t+1} with $t \geq n$ for some $n < \infty$ by Theorem 4.18 in Kim (2024). But $\{\omega \in ß_\infty = \bigvee_{t=0}^{\infty} ß_t : 1_{\{\omega\}} = 1$ when $P(A_{t+1}|ß_t) \neq 1$ for all $t \geq n$ for some $n < \infty\} \subset \{\omega \in ß_\infty = \bigvee_{t=0}^{\infty} ß_t : 1_{\{\omega\}} = 1$ when

$P(A_{t+1}|ß_t) \neq \alpha$ for all $\alpha \in \mathbb{R}[0,1]$ and for all $t \geq n$ for some $n < \infty\}$. Therefore, for any $\alpha \in \mathbb{R}[0,1]$, $P(P(A_{t+1}|ß_t) \neq \alpha$ at least $i.o.) > 0$ for some event A_{t+1} with $t \geq n$ for some $n < \infty$. Then, by (Case 3) of Theorem 4.16 in Kim (2024) and Theorem 3, the machines cannot learn the true probability $P(A_{t+1}|ß_t$ as α. $Q.E.D.$

Proof of Theorem 5 Suppose that $P(\ P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n$ $) > 0$ for some $n < \infty$ along the stochastic path of the test set. Then, by the Case 3 of Theorem 4.16 in Kim (2024), $P(p_k \to \alpha) \neq 1$ where p_k denotes the limiting relative frequency along the path of the test set. Thus, by Theorem 3, machines cannot learn the true probability $P(A_{t+1}|ß\ _t)$ as α.

Now, as $P(\ P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n\ ) > 0$ for some $n < \infty$ by assumption, consider a special case where $P(\ P(A_{t+1}|ß\ _t) \neq \alpha$ at least for infinitely many t's with $t \geq n\ ) = 1$. Then, even if $P(\ P(A_{t+1}|ß_t) = \alpha\ ) = 1$ for some $t^* < n$, machines cannot learn the true probability $P(A_{t+1}|ß\ _t)$ as α. $Q.E.D.$

Proof of Theorem 6 We prove this theorem by constructing a following counterexample where the almost-sure convergence to true probability function is satisfied but learning is not possible.

Counterexample Let us consider a simple process $\mathbb{X}$ whose realized values consist of the set D = $\{x_1, \ldots, x_k\}$ for each random variable X_t with a representative $x \in D$. Here, the set D is countably many, with k potentially infinite, because the number of functional values that machines can calculate is countable, and so without loss of generality we assume that the random variable X_t is discrete.

Now, let us assume that the process $\mathbb{X}$ has the following underlying true probabilities along the stochastic path: for any $n \in \mathbb{N}$, on the $[\frac{n}{2}]$ number of the periods t's where $\{X_t = x\}$ occurs at each t and $[\frac{n}{2}]$ is the nearest integer to $\frac{n}{2}$, $P(X_t = x|ß_{t-1}) = \alpha$, while $P(X_t = x|ß_{t-1}) = \beta$ on the rest of the periods for any $x \in \{x_1, \ldots, x_k\}$ with $\alpha \neq \beta$ along the stochastic path. Then, provided that $\lim\limits_{n\to\infty} \frac{1}{n} \sum\limits_{t=1}^{n} 1_{\{X_t=x\}}$ exists with true probability $P-$one,

(1) $P\ (\lim\limits_{n\to\infty} \frac{1}{n} \sum\limits_{t=1}^{n} P(X_t = x|ß_{t-1}) = \frac{\alpha+\beta}{2}) = 1$ if and only if $P\ (\lim\limits_{n\to\infty} \frac{1}{n} \sum\limits_{t=1}^{n} \mathbf{1}_{\{X_t=x\}} = \frac{\alpha+\beta}{2}) = 1$

by Lemma 4.10 and Lemma 4.11 in Kim (2024) (a similar result by Theorem 2.3.9 in Durrett (2019).)

However, note that given $\alpha \neq \beta$, $\lim\limits_{n\to\infty} \frac{1}{n} \sum\limits_{t=1}^{n} \mathbf{1}_{\{X_t=x\}}$ returns neither of the two true probabilities α nor β. Now, $\frac{1}{n} \sum\limits_{t=1}^{n} 1_{\{X_t=x\}}$ converges almost surely and $\frac{1}{n} \sum\limits_{t=1}^{n} P(X_t = x|ß_{t-1})$ does so as well by (1). Also, any arithmetic average of probabilities is also probability, because it satisfies three Kolmogorov axioms. Let us denote this new probability by P_n which is $\frac{1}{n} \sum\limits_{t=1}^{n} P(X_t = x|ß_{t-1})$. Thus, the condition of almost-sure convergence is satisfied by the sequence of probability functions $\{P_n\}_{n=1}^{\infty}$ but machines cannot "learn" any true probability from this convergence condition. The machines were able to effectively calculate the limit value of the convergent sequence of P_n's, but could not succeed in obtaining any true value, α or β.

Now, to learn the true probability α or β from data along the stochastic path, machines must classify the process $\mathbb{X}$ according to its underlying true probabilities so that $\lim\limits_{n\to\infty} (\sum\limits_{t=1}^{n} \xi_t)^{-1} \cdot \sum\limits_{t=1}^{n} (\xi_t \cdot 1_{\{X_t=x\}}) = \alpha$ with $\xi_t = 1$ when t comes from the periods when $P(X_t = x|ß_{t-1}) = \alpha$ and $\xi_t = 0$ otherwise, and so on. Then, we obtain the following from (1),

(2) $P(\lim\limits_{n\to\infty} (\sum\limits_{t=1}^{n} \xi_t)^{-1} \cdot \sum\limits_{t=1}^{n} (\xi_t \cdot 1_{\{X_t=x\}}) = \alpha) = 1$ if and only if $P\ (\lim\limits_{n\to\infty} \frac{1}{[\frac{n}{2}]} \sum\limits_{k=1}^{[\frac{n}{2}]} P(X_{t_k} = x|ß_{t-1}) = \alpha) = 1$

Now, without loss of generality, let $\mathbb{X}^*$ consist of the subsequence $\{X_{t_k} = x\}_{k=1}^{\infty}$ out of $\mathbb{X}$ where t_k's are chosen when $\xi_{t_k} = 1$. Therefore, to learn the underlying true probability of $\mathbb{X}^*$ as α, the machines must determine whether the selection criterion of $\xi_{t_k} = 1$ is satisfied or not at each t along the stochastic path. However, to determine whether $\xi_{t_k} = 1$ or not, machines must have learned whether $P(X_t = x|ß_{t-1}) = \alpha$ or not at each t. Then, this ends up with the result that machines must have learned the true underlying probability, in order to learn the true underlying probability. Clearly, learning is impossible in such a viciously circular way, no matter what definition of "learning" is adopted, including Definition 3 in this paper.

Therefore, we conclude that although the almost sure convergence condition is satisfied by this process $\mathbb{X}^*$, learning the true probability of this process $\mathbb{X}^*$ is not possible for machines. We prove by this counterexample that learning is not equivalent to the condition of almost-sure convergence. $Q.E.D$

Proof of Theorem 7 Suppose that machines solve a simple optimization problem in $\Re^n$ where the expected values of a given function $f : \Re^n \rightarrow \Re$ are to be maximized over a given set $\Game \subset \Re^n$. The function $E\left[f(x)\right]$ is called the objective function, say the expected gain function in the standard optimization model, and the set $\Game$ the constraint set. Now, let us suppose that μ and ν are two different probability measures such that ν is absolutely continuous with respect to μ. Then, there exists the *Radon-Nikodym* derivative h of ν with respect to μ so that the optimization problems under these two measures can be expressed as follows:

max $\int f(x)\; d\nu = \int f(x) h\; d\mu$ subject to $x \in \Game$.

Therefore, unless h is an identity function, the machines solve two different problems so that they optimize two different gain functions, $f(x)$ and $f(x)h$ under two different probability measures ν and μ. Notwithstanding, since the first-order conditions of the optimization problem are the same as $\frac{d}{dx}\int f(x)\; d\nu = \frac{d}{dx}\int f(x)h\; du = \eta \cdot \frac{d}{dx}g(x)$ where $\Game = \{x|\; g(x) = 0\}$ and η denotes a Lagrangian multiplier, we will obtain the same maximizer x^* as a solution, if any. Thus, we obtain the observational equivalence in the machine behaviors which manifest themselves as the same maximizer under different probability measures.

Now, it is equally possible that observationally equivalent behaviors, i.e., what are manifested as the same maximizer x^*, are derived from learning the true measure ν or some other measure μ. Therefore, under observational equivalence, we cannot exclusively measure machine performance by learning the true probability ν rather than some other probability μ, because:

provided that h is not an identity function and two measures, ν and μ, are different but absolutely continuous each other, $\int f(x^*)\; d\nu = \int f(x^*)h\; d\mu$.

Then, since there is no way to measure any improvement or worsening in machine performance exclusively from learning the true probability ν rather than learning the probability μ, we cannot show the practical significance of learning the true probability ν under this framework, which again implies that machines cannot learn the true probability by the very definition of Mitchell (1997), Definition 4. $Q.E.D.$

Proof of Theorem 8 For random vectors $Y_t \in \mathbb{R}$ and $X_t \in \mathbb{R}^I$, let y and x denote generic realizations and y_t and x_t denote sample realizations at t. Now, for some $s < t$, let $f(Y_t|\{X_i\}_{i=s}^t)$ be the conditional probability $P(Y_t = y|\{X_i = x\}_{i=s}^t)$ from the true data-generating distribution and let $f_m(Y_t|\{X_i\}_{i=s}^t)$ be our parametric model P_m for the true probability P. Then, given that Y and X represent some data sequences of $\{Y_t = y_t\}_{t=1}^n$ and $\{X_t = x_t\}_{t=1}^n$, we can rewrite the optimization equation in Theorem 7 by the joint conditional probabilities as follows:

(1) $$\min \int \frac{f(Y|X)}{f_m(Y|X)} = \frac{P(Y|X)}{P_m(Y|X)}\; dP$$

Here, note that this is the problem of minimizing the Kullback-Leibler divergence. Then, under the assumption that our probability model is the "closest" to the true data-generating function, machines can estimate the parameters of our model by solving this minimization problem. Now that the minimizing solution, if any, will be the same as in the following problem, we can rewrite (2) as follows;

$$(2) \qquad \max \int log\; P_m(Y|X)\; dP$$

Then, for example, under the assumption that the logarithmic representation of our model likelihood function is concave with respect to the parameters, machines can estimate the parameters from the given data X and Y using the deep learning method. Therefore, given I number of training samples X_{ti}'s at time t and J_k number of latent variables Z_{tj}'s in each k-th layer with K number of layers, machines now calculate the following equation:

$$(3) \qquad P_m(Y_t \mid X_t; \widehat{\Theta}) = P_m(Y_t; g(Z^K_{t1}, \ldots, Z^K_{tJ_K}; \widehat{\Theta}_K))$$

where $Z^K_{tj_K} = h^{(K)}(Z^{K-1}_{t1}, \ldots, Z^{K-1}_{tJ_{K-1}}; \widehat{\Theta}_{K-1})$ for $j_k = 1, \ldots, J_K$ and $Z^1_{tj_1} = h^{(1)}(X_{t1}, \ldots, X_{tI}; \widehat{\Theta}_1)$ for $j_1 = 1, \ldots, J_1$ and $\widehat{\Theta}_i$ denotes the set of estimated parameters at each layer $i \in \{1, ..., K\}$.

Note that we can obtain the optimization problem (2) from the setting of Theorem 7 once we replace the objective function f in Theorem 7 by the log of some *given* parametric function $P_m(Y|X)$. Then, the problem of observational equivalence in disappears now, because $f(x)$=$f(x)h$ in Theorem 7 is uniquely fixed as the log of $P_m(Y|X)$ with respect to the true measure P and therefore h must be an identity function. $Q.E.D$

Proof of Theorem 9 Let each word w with a length n_w be a sequence of the letters of the alphabet $\{a_1, \ldots, a_{n_w}\}$ and each sentence S with a length n_s be a sequence of words $\{w_1, \ldots, w_{n_s}\}$ that humans have circulated to effectively communicate up to a certain time t_0. Here, sentence is distinguished from arbitrary combination of ungrammatical sequence of words that cannot be used for human communication. We can extend the set of the letters of the alphabet to include all the punctuation marks, numerals, etc., if necessary. Note that for any $w \in W^*$, n_w must be finite, because no word can be included in the ideal corpus if its sequence goes to infinity. This is so because any infinite sequence cannot actually be used to effectively communicate at any given time. For the same reason, for any $S \in W^*$, n_s must be finite as well, except for the cases where some words are contained in sentences repeatedly infinitely often. In such exceptional cases, using the expression "..." which can be treated as the 27th alphabet, we let those sentences contain the word which consists of unitary sequence of this 27th alphabet. Then, for any t_0, there must exist some fixed number n^*_w and $n^*_s < \infty$ such that n^*_w denotes the the maximum size of the sequences of alphabets among all the words in the ideal corpus W^* while n^*_s denotes the maximum size of the sequences of words among all the sentences in W^*.

Now, note that the size of the population W^* must be less than $\lambda\times$ (27 to the power of $n^*_w \times n^*_s$) for some $\lambda \in \mathbb{N}$ as follows: here, λ is fairly large but countable, because the number of frequencies for actual usages of any given natural language is at most countably many, given that there are at most countable number of people who have used those languages to effectively communicate throughout history up to a given time $t_0 < \infty$ in our world. Now, let us denote the population with size $\lambda\times$ (27 to the power of $n^*_w \times n^*_s$) by $W_{\max}$. Then, at any given time t_0, there always exists the true population $W^* \subseteq W_{\max}$ which contains the accumulated history of the words or sentences that had been circulated up to t_0. The size of W^* is smaller than that of $W_{\max}$, because not all possible sequences of words in $W_{\max}$ are grammatical, so they may not be circulated for actual usage to effectively communicate. Now, if W^* does not exist within $W_{\max}$, it implies that there must exist some sentence S^* with the size greater than n^*_s such that $S^* \in W^*$, which is contradictory, because n^*_s is the maximum size of all the sentences in W^*.

Now, let us denote a function f_c from the population W^* to the set of natural numbers by a counting function. This counting function f_c counts the frequencies of each given sentence S^* out of the entire true population W^*. Since the set of languages is lexicographically Turing-enumerable, the counting function f_c is Turing machine calculable. Now that W^* exists and that the counting function f_c is Turing-machine calculable, machines can effectively calculate the relevant empirical distribution out of the population W^*, which is the true probability distribution because the counting function f_c provides $P(w^k|w_1^{k-1})$ for each k with $1 \leq k \leq n_s$ and thus provides $P(S)$ by chain rule to machines for any given n_s.

Now, the true population W^* is available in principle to machines at any given time t_0 because the size of W^* is finite while the machines are ideal ones in the sense that they can disregard any practical limits to process data. Also, the machines can effectively calculate the empirical distribution of this W^*, which returns the true probability $P(S)$ because $P(S)$ is *defined* to be $P(w_1^n) = \prod_{k=1}^{n} \frac{C(w_k, w_{k-1}, \ldots, w_1)}{C(w_{k-1}, \ldots, w_1)}$ where $C(w_0)$ denotes the number of all the words in the ideal corpus. Therefore, $P(S)$ is directly observable to machines by Definition 7. Finally, by Theorem 4.36 in Kim (2024), $P(S)$ is learnable by the machines. $Q.E.D.$

Proof of Theorem 10 Let us suppose f^*_∞ exists and further that $f^*_\infty = P$ where $f^*_\infty = f^*(\{X_i\}_{i=1}^{\infty})$ and P is the true probability of r-star. Then, by Definition 7, the true probability of r-star is not directly observable because r-star is a model-based estimate, not actual data. Thus, machines cannot learn the true probability of r-star by Theorem 4.36 in Kim (2024). If f^*_∞ does not exist, then machines trivially cannot learn the true probability of r-star becasue there exists nothing to learn. Now that machines cannot succeed in effectively calculating f^* by Definition 3, ℓ is not well defined for machines because one of the arguments of the distance function ℓ is not calculable by machines. Therefore, the machines cannot evaluate whether any learning rule $f_{n,m}(\{X_i\}_{i=1}^{m}, f^*(\{X_i\}_{i=1}^{n}), X_j)$ is within any bound from $f^*(X_j)$ for any $j \geq m+1$, for any $n, m < \infty$. $Q.E.D.$

B Further Detailed Remarks

Remark 1 For example, if the following equation (2) holds, then $P(\ P(A_{t+1}|ß_t) = \alpha) = 1$ is not equivalent to that machines learn true probability $P(A_{t+1}|ß_t)$ as α at t.

(2) $\quad P(\ P(A_2|ß_1) = \alpha, \ldots, P(A_n|ß_{n-1}) = \alpha, P(A_{n+1}|ß_n) \neq \alpha, P(A_{n+2}|ß_{n+1}) \neq \alpha, \ldots) = 1.$

At first glance, this result may look puzzling, because machines are not said to learn even if they correctly calculate the true probability for the first finite $n-1$ number of times with true probability P- one. However, recall the discussion on the necessary condition (1) that obtaining the true probability *by luck* cannot be considered as learning. In fact, according to equation (2), here it is not only the case that $P(P(A_2|ß\ _1) = \alpha, \ldots, P(A_n|ß\ _{n-1}) = \alpha) = 1$, but also the case that $P(\ P(A_{n+1}|ß\ _n) \neq \alpha, P(A_{n+2}|ß\ _{n+1}) \neq \alpha, \ldots) = 1$. Now, note that if the machines had not been calculating the true probability correctly only by luck, they could not have been correct only for the *finitely* many times and wrong for the rest of the *infinite* times, with true probability P- one. For this reason, learning is defined as a *success* in effective calculation in Definition 3. Calculating the correct values only for some finite times by luck while wandering around cannot be considered as a genuine success. If such computations had indeed been scientifically successful, they should not have been wrong that many times with the true probability P- one.

Remark 2 We note that these two questions are different, especially because languages are creative. Note that any *new* sentences can be created for the purpose of actual usage to effectively communicate with, even if they have never been circulated before. Thus, any of the potentially new sentences, which have not yet circulated in our world, should have zero frequency in the given ideal corpus, no matter how large we set the ideal corpus. Then, any of those languages should have a potentially infinite sequence of strings. Thus, the probability of obtaining any arbitrary new sentences simply becomes zero when we calculate its probability from the empirical distribution in the given ideal corpus. But it is obvious that the probability of generating any new sentence cannot be zero if we admit that languages are indeed creative. Thus, the probability of obtaining an arbitrary new language must be defined in a different way from the way of determining the probability of obtaining any given sentence in the ideal corpus. In any case, however, what we emphasize here is that this zero-frequency problem of creative language is not relevant to the case of a simple N-gram model where inputs are given.

Recall that the unambiguous form of any inputs which have ever been given to machines in the tasks must be always included in the ideal corpus. For example, in speech recognition tasks, the unambiguous form of any given input must actually have been used for speech to effectively communicate before being given to machines, and therefore must be included in the ideal corpus at any given time t. Then, the problem of zero-frequency of newly created sentences in the ideal corpus does not apply here. Even in the extreme case where any input was just created at the very moment when the given speeches were made, the true probability of obtaining such sentences would not be definitely zero.

In any case, any further detailed discussions on the creative language will be beyond the scope of this paper. Instead, we only emphasize here that what we try to show with Theorem 9 below is the following: If the true probability of obtaining any *given* sentence can be conceptually defined by the empirical frequencies of the ideal corpus in Definition 5, the true probability in this simple N-gram model is an example of directly observable probability by Definition 7 and thus it is learnable.

Remark 3 Due to the incredible technological development such as in the deep learning and big data algorithms, real machines may be able to obtain the empirical distributions of the entire population efficiently enough in the near future even if the size of the population may be astronomical. In these cases, the machines do not have to estimate the true distribution of the *population* from some small samples. Instead, they only have to *directly observe* the true probability from the empirical distribution of the entire population data.

For example, let us consider an imaginary case where the Wonka Factory announces a contest in which a Golden Ticket is included in each of the five random Wonka chocolate bars worldwide and the winners will receive the full package of prizes along with the lifetime supply of Wonka bars. Now, let us further assume that to gratify his daughter's desire to win one ticket, the father of Veruca Salt hires workers. Among them are included the data scientists who use machine learning algorithms to collect and then process all the necessary data about the entire number of the Wonka bars being circulated in the global market and the number of the tickets already claimed worldwide in order to calculate the true probability of winning a ticket at every instant. Now, it is clear in this case that machines used by the data scientists *can* be thought to *learn* the true objective probability of Veruca Salt's winning a Golden Ticket.

Just like in the Wonka Bars example, when machines *directly observe* the empirical distribution of the *true population* while using deep learning and big data algorithms for instance, (i) machines come to calculate it by the following *definite and explicit instructions* on how to *effectively calculate* the true probability: collect all the necessary data from the *given* whole population and then do the relevant calculation with the data, in order to obtain the relative frequencies of the target attributes in the entire population. (ii) Now, this effective calculation must be *successful*, because the true probability *is defined* to come from such an empirical distribution by Definition 7. Then, by Definition 3, machines come to *learn* the true probability, because machines achieve computational success by (i) and (ii). Therefore, we argue that at least in such cases machines, in principle, can learn the true objective probability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We did our best to reflect the paper's contributions and scope accurately in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work at several points in the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete proof either in the main body of our paper or in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: Our paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: Our paper does not include experiments requiring code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: Our paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Our paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: Our paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and tried our best to conform with it while doing the research conducted in this paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper discusses very abstract and foundational questions regarding the nature of machine learning. It is unlikely that work on these questions will have societal impact, whether positive or negative.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not contain datasets or models, and accordingly does not pose any such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Our paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.