# A Preliminary Study on the Capability Boundary of LLM and a New Implementation Approach for AGI

**Authors:** Yongcong Chen, Ting Zeng, Xiaoyi Qia, Yongcong Chen, Ting Zeng

**Date:** 2023-05-06T00:00:00+00:00

## Abstract

Through preliminary exploration of the capability boundaries of Large Language Models (LLMs), we contend that current mainstream artificial intelligence predominantly employs a technical approach combining attention mechanisms, deep learning, and reinforcement learning—an approach that cannot be applied to domains where extensive trial and error is difficult. Therefore, to achieve Artificial General Intelligence (AGI) that operates effectively in any field, a paradigm shift is necessary. Consequently, we propose a novel machine learning solution distinct from the conventional "deep learning + reinforcement learning" framework. Our approach adopts few-shot learning and cumulative learning, implements attention mechanisms analogous to those in transformers, and constructs a fully connected knowledge network. Moreover, it enables interactive decision-making with the environment without requiring extensive trial-and-error-style learning. Furthermore, humans can prespecify diverse innate needs to achieve multi-objective balance, thereby attaining substantially higher safety than current artificial intelligence systems. In this paper, we propose a suite of new machine learning techniques that may guide humanity toward realizing AGI.

## Full Text

## Preamble

### A Preliminary Study on the Capability Boundary of LLM and a New Implementation Approach for AGI

Yongcong Chen ((cid:0))
New Millennium Future Technology Co.
YongcongChen@sina.com

Ting Zeng
Tsinghua University
ceng-t@mail.tsinghua.edu.cn

Xiaoyi Qian
Galaxyeye Technology Co., Ltd

Jun Zhang
Shenyang Normal University

Xinyue Chen
Tsinghua University High School

## Abstract

Through preliminary exploration of the capability boundaries of Large Language Models (LLM), we argue that current mainstream artificial intelligence, which generally adopts the technical approach of "attention mechanism + deep learning" + "reinforcement learning," cannot be applied to fields where extensive "trial and error" is difficult or impossible. Therefore, to achieve Artificial General Intelligence (AGI) that works in any domain, we must change our approach. We propose a fundamentally different machine learning solution from "deep learning + reinforcement learning." Our approach adopts small-sample and cumulative learning, implements an attention mechanism similar to the Transformer, and creates a fully connected knowledge network. Moreover, it enables interactive decision-making with the environment without relying on extensive "trial and error" style learning. Additionally, humans can preset different innate needs to achieve multi-objective balance, resulting in far greater security than current AI systems. This paper proposes a new set of machine learning techniques that may guide humanity toward realizing AGI.

**Keywords:** LLM · ChatGPT · GPT-4 · AGI

## Introduction

Currently, most artificial intelligence systems adopt the technical roadmap of "attention mechanism + deep learning" + "reinforcement learning." The "attention mechanism + deep learning" component is primarily used to build knowledge networks, while "reinforcement learning" is mainly used to enhance the continuous interactive decision-making capability between machines and their environment. For example, Google's Gato model, launched in June 2022, can perform over 600 different tasks within a single model [**?**]. Another example is GPT-4, which has demonstrated remarkable progress in knowledge understanding and reasoning [**?**]. But what is the upper limit of current LLMs? Can "attention mechanism + deep learning" + "reinforcement learning" achieve true "artificial general intelligence"? In this article, we explore the capability boundaries of LLMs and propose a new path toward AGI.

## 2 The Upper Limits of LLM

First, we note that current "reinforcement learning" uses external feedback to inform the machine whether outcomes are "good" or "bad" along different decision paths. This external feedback can come from preset reward functions—for instance, in AlphaGo, the feedback derives from the reward function that determines win/lose outcomes. It can also come from human feedback, such as the RLHF (Reinforcement Learning from Human Feedback) technology widely used in LLMs [**?**]. Therefore, the essence of "reinforcement learning" is a "try first, then feedback" approach, enabling machines to obtain different "reward" information through various decision paths. The machine can then derive the relationship between "state" and "policy," thereby acquiring interactive decision-making capability with the environment [**?**]. We believe that the current "reinforcement learning" approach resembles an "evolutionary" learning method when machines need to interact with their environment to make decisions. Its essence is "trial and error" with elimination through external feedback, very similar to biological evolution. Consequently, this type of learning only works in virtual environments such as games, metaverses, and content generation tasks. For tasks requiring interactive learning in real environments—such as caring for the elderly or driving vehicles—reinforcement learning may not work well.

Second, "deep learning" creates knowledge that is difficult for humans to understand, resulting in two separate knowledge systems that are mutually incomprehensible! For example, humans struggle to understand the decision-making processes of LLMs. Similarly, LLMs have difficulty truly understanding the knowledge represented by human language. For instance, there is no AI agent that can read toaster instructions and then operate a toaster to bake bread in different bakeries [**?**] [**?**] [**?**]. Humans, by contrast, can acquire accumulated knowledge about toaster usage by reading instructions, enabling them to make decisions guided by existing knowledge when facing the new task of "operating a bread machine." For example, when opening the top of a bread machine, humans do not need to try various schemes first (such as smashing the top), but directly access accumulated human experience through language.

Therefore, we believe that real machine learning should resemble human learning. When facing a new task, we should be able to predict the "goodness or badness" of different decision paths based on past experience, obtaining decision knowledge to handle the new task by choosing a limited number of cases to try. Furthermore, true learning should occur as children learn—by directly acquiring accumulated human experience through language. When facing a new task, one should succeed on the first attempt without any trial! For example, in a laboratory, when teachers instruct children in experiments, they transmit human decision-making experience directly through language. After acquiring this knowledge, children can complete experiments in different environments by making step-by-step decisions and interacting with the environment, even if it is their first time performing these experiments!

Consequently, we believe that due to flawed knowledge structures and learning methods, current LLM-based AI cannot solve the following serious defects:

## 2.1 Cannot Solve Problems Independently

For example, current artificial intelligence does not take the initiative to help when it sees its owner fall down [**?**]. This is because a machine cannot generate its own goals without having its own needs. Since a machine has no goals of its own, it cannot actively create tasks. Therefore, when facing unexpected situations in real social life, machines must handle them through preset processes. These preset processes can come from prompt flows written by external humans [**?**], or the machine can follow preset instructions to find similar processes in its knowledge base and imitate the entire process.

Some researchers are now attempting to write recursive calling functions. For example, under one task, call a function that says "look for processes that have done similar tasks," and then for each process obtained, call another function that says "look for processes that have done similar tasks in the past" to obtain more processes. This constitutes a recursive call formula. The current representative project is AUTO-GPT, which aggregates knowledge obtained from search engines with knowledge from GPT-4 through storage, then uses this as prompts to find solutions from GPT-4. Through recursive calls of this process, relatively macro tasks can be accomplished. If all processes and parameters needed for the task already exist in the AI's knowledge library (such as an LLM), it can complete the task. However, what AUTO-GPT can do does not exceed the boundaries of what LLMs can do. Moreover, in real life, an endless number of unexpected situations may cause a point where there is no appropriate process and parameter matching the current environment, leading to recursive call failure [**?**] [**?**] [**?**].

As a result, LLMs do not spontaneously create new processes! A large model (such as an LLM) is essentially a high-level programming language whose syntax is natural language. Understanding this reveals the boundaries of large models. No matter how many high-level functions we add to the large model, no matter how many tools and apps we integrate, the large model will not spontaneously create new processes. All its processes are modeled after past processes (recursive calls) or processes that humans preset for them (main program calls). In both cases, it is essentially "using preset processes to deal with all problems" [**?**]. No matter how many "if…else…" branches are considered, they are all preset and preexisting—not created on the fly for specific tasks!

Therefore, the essence of large models is to provide a programming platform for human beings, where the programming language is human language. Previous programming languages were C++, Java, Python, and other computer languages. Large models also provide a set of functions that can be called in human languages. For example, the "Create PPT" function is a human language command plus the required function parameters. The large model completes

tasks in this command + parameter mode. In this way, everyone can be a programmer in the future. Those with a better understanding of large models who can write better programs would be called "Prompt engineers"—the programmers of the future.

Now, large models continue to enrich their functions. For example, in the future, function input parameters may include images, videos, actions, or any sensor input, and outputs may also include images, videos, actions, or other multimodal sequences [**?**]. Therefore, in the future, all processes can be programmed in large model languages (natural languages). Large models are machine language platforms, fundamentally no different from Python. For example, in Python programs, human minds are transformed into Python program flows, and the Python platform translates functions into machine programs understood by various underlying drivers to accomplish tasks, thereby implementing human preconceived problem-solving flows.

Thus, the large model is the new Python. Python is better suited to human interfaces than C++, making it more popular. GPT-4 is better for human interfaces than Python, so it will be more popular in the future. But C++ is not disappearing because it has advantages in certain areas. Similarly, various dedicated large models, or Python, are not disappearing—there is still room for them. Therefore, in the future, all software can be rewritten in large model languages, and Prompt engineers will be the next generation of "white-collar workers" skilled in writing large model programs [**?**].

Consequently, the essence of large models is a more user-friendly "programming language." Its interface is natural language, its functions are diverse and multimodal, and it is expected to establish its own ecosystem soon. But it is still very much a programming language! It is a programming language with very low barriers to entry and greatly expanded capabilities—that is its essence. So the future based on large models can accomplish any task with a predictable flow. Complex tasks are nothing more than writing more if…else… branches.

But in real life, there are plenty of tasks whose flow cannot be predicted! Such as caring for the elderly, driving, cooking, spending time with children, farming, etc. Humans can consider all kinds of possibilities for helping, but in the process of interacting with real environments, there are always surprises. How machines will handle these contingencies is impossible for humans to predict. This can be very dangerous, especially when large model functions are involved in all aspects of human life, potentially leading to unacceptable losses.

## 2.2 Knowledge Cannot Be Updated in Real Time

Currently, with the adoption of big data training in artificial intelligence, knowledge cannot be updated in real time. However, real-time knowledge updating is crucial for machines that interact with their environment because the interaction between machine and environment is the process by which the machine acquires new knowledge. If the knowledge acquired by the machine cannot be

updated in real time, the machine will be unable to update its decision-making knowledge based on environmental feedback. Therefore, such a machine, when faced with the same environment, will keep making the same mistakes [**?**]. It cannot be applied to domains that require interaction with real environments.

In fields where interaction with the real environment is required—such as autonomous driving, housework, patient care, etc.—the machine needs to establish knowledge of interactive decision-making between its own behavior and the external environment. These domains do not permit extensive trial and error, so machines cannot build decision-making knowledge in these domains through reinforcement learning by interacting in real environments. Therefore, current technical solutions in artificial intelligence cannot be applied to these fields [**?**].

## 3.1 How to Describe the Information Contained in a Matrix?

Although a matrix may contain many vectors, our first concern is: how many independent vectors are there? That is, what is the rank of the matrix? We can then, through matrix decomposition, find the corresponding set of eigenvectors, which numerically equals the rank of the matrix and also constitutes a set of coordinate basis clusters for this matrix. This set of coordinate basis clusters is complete, orthogonal, and represents the simplest description of the matrix. Any vector in the matrix can be expressed using this set of coordinate basis clusters. If we establish a coordinate basis cluster that is not orthogonal but is complete, we can still use this coordinate basis cluster to express any information in the matrix. If the coordinate basis cluster is incomplete, then some vectors in the matrix cannot be expressed by the coordinate basis cluster, necessitating an increase in the dimension of the coordinate basis cluster.

So how do we identify what information a vector in the matrix contains? Obviously, if it is a complete basis cluster, each basis is a dimension, and any vector projected onto the coordinate basis can accurately obtain all information contained in the vector. If the basis coordinate cluster is orthogonal, we achieve the simplest coefficients to express all information of the vector. If the basis coordinate cluster is not completely orthogonal, we want it to be as close to orthogonal as possible, because then the obtained coefficients are sparse.

By combining the sparse coefficient matrix with a set of basis coordinate clusters, we can understand the information contained in any vector in the matrix, where the information components are relatively independent. Therefore, the relationship between two vectors is reflected in the relationship between their coordinate basis components.

Consequently, if two vectors in a matrix have non-zero components in the same dimension, they are considered to have local similarity. We can assume there is some connection between the two vectors. If there are multiple local similarities between two vectors, we can assume there is a stronger connection between

them. If a similar connection occurs repeatedly throughout the matrix, we can assume this connection is a general rule. This is knowledge.

Therefore, in a matrix, searching for knowledge means searching for all universal rules within it. The rules themselves are expressed by the partial basis components of the matrix, including dimensions as well as dimensional relationships. They represent common connection relations obtained from a large number of vectors, having lower dimensionality and wider applicability. Thus, knowledge is generalized from a large number of vector relations. Each piece of knowledge is itself a vector in the matrix and can be expressed as a coefficient matrix. A large amount of such knowledge constitutes a knowledge network.

Therefore, if we need to discover all knowledge in a matrix, the most important task is to find a set of basis coordinate clusters and use it to decompose any vector in the matrix; then, through coefficient matrix relations, obtain connections between different vectors; and finally find those connections that can be repeated—these are the knowledge extracted from the matrix information.

However, a matrix can have multiple sets of basis coordinate clusters. As long as the dimension does not fall below the rank of the matrix, essentially any basis coordinate cluster can be adopted. To obtain the most concise body of knowledge, we can simply use the orthogonal coordinate cluster. However, if the matrix is very large, it is difficult to find a set of orthogonal basis clusters. In this case, we can follow the most repetitive knowledge and use it as the basis coordinate cluster. In this way, at least most of the knowledge in the matrix can be represented by a sparse matrix. That is, we obtain a set of coordinate basis clusters that can concisely express common knowledge in the matrix.

With the basis coordinate cluster, any vector can be decomposed into the basis coordinate cluster and represented by the coefficient matrix. The similarity between any vectors can be expressed by their spatial distance, which can be represented by Euclidean space distance. The mapping relationship between any vector can be realized through the mapping relationship of the coefficient matrix.

## 3.2 How Does Deep Learning Create Knowledge?

Suppose a group of people on an alien world build a 4-dimensional information space containing a large number of images, sounds, and actions. The humans there use pixels, syllables, and movement patterns as a baseline for their perceptual abilities, making them unable to see connections between complex combinations of pixels, syllables, and motion patterns.

Therefore, they begin to use trial and error, trying different basis coordinate clusters in hopes of finding a specific basis coordinate cluster where the combinations of pixels, syllables, and motion patterns they are interested in could form separate clusters. This would be the basis cluster they need.

Since the dimensionality of the original data is very high—for example,

$64\$ \times 64 images have dimensions of 64 \times 64, using 64 \times 64 two-dimensional impulse functions as the original basisc$ because their goal is to find common feature combinations rather than having to express all information. Some information will inevitably be lost, and they cannot fit all original information into the final dimension. Therefore, one possible trial-and-error method is to discard or compress some transformed dimension information after each attempt at a coordinate basis cluster, then compare it with the target to see if the error increases or decreases, and decide the next direction to change the coordinate basis cluster. Obviously, with each attempt, a portion of information is lost. After many attempts, the information loss becomes too great, potentially leading to the loss of useful information and preventing task completion. Therefore, the proportion of useful information in the overall information and the information loss rate after each change determine the maximum number of changes. However, it may be difficult for the machine to find the optimal coordinate basis cluster within limited transformation trials. Therefore, a feasible solution is to add back some of the lost information after each transformation, allowing the number of transformation layers to be increased and improving the probability of finding the optimal coordinate basis cluster. This is known as the residual network. Of course, we can also consider inserting weak nonlinear functions in the mapping process of multiple neural networks to increase the number of possible transformations.

In searching for the optimal coordinate basis cluster through trial and error, the direction to look for is error reduction, and the tool to achieve it is the BP algorithm. The data appearing in neurons is actually the coefficient matrix under the basis coordinate cluster, while the basis coordinate cluster itself is implicit and does not appear in the multi-layer neural network. The inter-layer transformation coefficient is the coordinate coefficient transformation matrix in the transformation from one implicit basis to another implicit basis. The BP algorithm adjusts the coordinate coefficient transformation matrix from one implicit coordinate basis to another implicit coordinate basis waiting to be tried.

Of course, if the selected basis coordinate cluster itself is not an orthogonal coordinate system, modifying coefficients in one dimension may affect coefficients in another dimension. This may result in a situation where the overall error no longer decreases through any adjustment of coefficients. The core reason is the non-orthogonal nature of the basis coordinate clusters. If they were orthogonal, this would not occur.

Therefore, it is necessary to move as far as possible toward the orthogonal coordinate cluster in the attempted path. A sign of approaching an orthogonal coordinate cluster is a sparse coefficient matrix. Therefore, the direction of the entire attempt needs to increase the sparsity constraint of the coefficient matrix, which is achieved through various regularization methods.

Additionally, it should be pointed out that the nature of convolution, pooling, or other variants of deep learning has not changed. Convolution, for example, is essentially a neural network map with an artificially large number of zero

coefficients in the coordinate coefficient transformation matrix. Pooling is nothing more than a layer of neural network mapping where some dimensions are removed and certain nonlinear functions are adopted. This is the essence of deep learning: a way to find a coordinate basis in a matrix. If there are labels in the data space, then the number of labels is the dimension of the basis site-cluster ultimately required. Their goal in finding the final basis coordinate cluster is to use the combination of minimum resolution features (in this case, pixels, syllables, and action patterns) common to each type of labeled data as a representative of each type of label, as well as a basis coordinate cluster. The resulting coefficient matrix is sparse. Thus, we can see that the essence of deep learning is also to find a suitable set of basis coordinate clusters in the information matrix. If the required information is viewed as an information subspace, then the basis coordinate cluster obtained by deep learning can express any vector in this subspace, which is supervised learning. If the subspace is large and directly contains all information, then the basis coordinate cluster obtained by deep learning can express any vector in the entire information matrix, which is unsupervised learning. If the main purpose of learning is to cluster and discard information that cannot be clustered, then this is also unsupervised learning.

## 3.3 What Is the Nature of the Attention Mechanism?

The core of the attention mechanism is to discover common arrangements among the elements of the information matrix. These common arrangements can be chosen as the "framework" of the information space. The so-called "framework" is that they generally exist in the information matrix, and using them as the coordinate basis cluster enables concise description of common vectors in the matrix.

More generally, we can think of each character as a dimension in the linguistic information space. If we adopt such a coordinate basis cluster, we can use it to describe any vector in the linguistic information space. However, such a coordinate basis cluster may not be optimal. If we take the linguistic information space as a matrix, then the optimal coordinate basis cluster is obviously the basis cluster composed of the eigenvectors of this matrix. The basis cluster composed of eigenvectors has the smallest rank and the most concise description information.

For example, we can take the sentence "I am going to attend a friend's wedding today" and decompose it with each character as a dimension, obtaining a 9-dimensional coefficient matrix: "I, am, going, to, attend, a, friend's wedding, today." But we can also use "subject…predicate…object" as a coordinate basis, where the coefficient of this coordinate basis is "I…attend…wedding." Then we can use "auxiliary word + predicate" as a coordinate basis, where the coefficient is "to attend today," and finally use "attributive + object" as a coordinate basis, where the coefficient is "friend's wedding." Obviously, the latter uses a more concise basis coordinate cluster to express the same information. Moreover, the basis of the latter coordinate cluster is framed. These frame coordinate clusters,

under different coordinate components, can constitute a large amount of similar information.

The core of the attention mechanism is to establish these "frame" coordinate basis clusters [**?**]. The method adopted is to extract the common underlying framework between the elements of the information matrix. This process is very similar to human learning. When we learn information from a book, we follow the same principle: "Books need to be thinned first and then thickened." To "be thinned first" is to summarize the framework information in a book—this is a process of information compression. "Then thickened" is to add different details on the basis of the framework information to form new knowledge created by us—this is a knowledge generalization process [**?**] [**?**] [**?**]. Therefore, the core of Transformer-class models is the attention mechanism [**?**]. The core purpose of the attention mechanism is to obtain the common arrangement of elements in the information matrix and weight them according to how common they are. The more common the arrangement, the higher the weight. Those high-weight permutations constitute the main framework for how all information is organized in the information matrix.

This process is similar to signal processing in communications. In the time domain, seemingly chaotic and complex signals are converted to frequencies, where their low-frequency components determine the general trend of the signal—these are also the main components of the signal. These low-frequency components are the common form of organization found in this type of signal. If we think of each low-frequency component as a basis coordinate cluster component, then they are analogous to attention mechanisms. The low-frequency components express common connective relationships between information, and they are the "frame" of information. Thus, the attention mechanism obtains the "framework" of how information is organized by looking for the weighted connections between pieces of information. These "frames" are the basis of generalization. The mapping relationship between "frame" and "frame" represents the algorithm from one "vector" to the next "vector." Input "frame" + different details constitutes the specific input vector, and through the algorithm from "vector" to next "vector," the output vector can be obtained—this is the knowledge generalization process.

In fact, humans adopt the same approach in the learning process. The common combinations of features—the specific "concepts"—are those with higher weight. They are just common combinations in space and time. Common combinations of features further summarized from concrete "concepts" are "abstractions." This process can be iterative. Therefore, there are many hierarchical "concepts" in human society, which are frames. Thus, the abstract frame "cat" is a common arrangement of matrix elements in space and time, and this arrangement may contain multi-modal matrix information elements of "cat," such as language, text, sound, image, action, touch, etc. In this arrangement, some matrix elements may have higher weights because they are more common, and they may all be the concept of "animal." With fewer elements, "animal" has a

smaller scope but wider applicability, so between “cat” and “dog,” the knowledge related to the combination of features they share (such as the concept of “animal”) can be directly generalized.

The central ability of the attention mechanism, then, is to instantiate the statistical associations between languages into embodied input. The statistical association between languages is obtained through pre-training, and this statistical association is a kind of incomplete statistical association. It does not count the correlations between the elements of all possible combinations of languages in all their arrangements, because that is an impossible task. Therefore, the actual correlations between languages in specific permutations need to be further optimized based on statistical correlations. This step is accomplished by the attention mechanism.

The core purpose of the attention mechanism is to find the correlation between input information, and between input and output information, using trial and error and using the relationship between human language as self-supervision, expressing this correlation through weights. This correlation, however, is very similar to the human learning process. The machine uses deep learning—essentially trial and error—to find an optimal set of coordinates. This set of coordinate bases is likely to be very close to common human concepts. Therefore, the core of deep learning is using trial and error to find a basis coordinate cluster, and the core of attention is using trial and error to bring a basis coordinate cluster closer to human-like concepts.

The core task of LLM is to form a large number of frames, which fuse with each other to form a network. They are hierarchical. After data entry, the most important task is to find the best matching frame—this is the basis for generalization.

## 3.4 Why Do Large Models Have the Ability to Emerge?

Why do large models “emerge”? It’s simple. For example, if an American comes to China, they can complete correct translation based on a large amount of common background information between humans (such as personal needs, social structure, etc.) and a moderate amount of comparison between Chinese and English. The large model is like an alien. It has no common background information with humans. What it sees is only how human information is connected. Therefore, it needs to extract the connections between human information to predict how that information will develop. In the beginning, when it doesn’t have enough samples, the “information frame” it extracts is very different from the human “information frame,” so it keeps making mistakes, groping in the dark, always hitting walls. As the number of samples increases, its “information frame” and the human “information frame” have a higher probability of aligning. But it’s not a linear process. Like an anthropologist deciphering ancient languages, it gropes in the dark with little progress until it reaches a certain threshold. At a certain point, if the number of correct answers reaches a thresh-

old, the decryption process speeds up dramatically. This is the phenomenon of "emergence."

Machines emerge not from intelligence but from finding the "common association of the right pieces of information." This common association of the right information is similar to how humans use information. And because all evaluation criteria are human criteria, when it has enough bases and they're close enough to human bases, its capabilities emerge.

The ability of large models to "emerge" is, at its core, to bring concepts generated by the large model closer to human concepts through the attention mechanism. Therefore, the training data must be sufficient for emergence to occur. The ability to generalize occurs because concepts are close to human concepts. Human concepts contain many abstract concepts that serve as frameworks for information mapping. For example, "cat" is an abstract concept because it doesn't represent a specific cat. Given a lot of input and output information, the machine can build up frame information in those inputs and frame information in those outputs, and establish a mapping process from the input frame to the output frame. Then input frame + details can obtain output frame + details through the same mapping process. This is the knowledge generalization process.

When the training data is large enough, the machine is likely to find common complex patterns in it. The spatial association of common combinations of patterns is a thing, and the temporal association of common combinations is a process. The association of common combination patterns in space and time is knowledge. Thus, the large model seems to have knowledge about things and flows. This framed knowledge is called the "world model." It is on the basis of their own framework of knowledge that humans understand and interact with all things. We can think of the large model as analogous to looking at things in the frequency domain. Similar to a picture, we can use a small number of low-frequency components to obtain the main content of the picture—this is the core of image compression technology. The attention mechanism, at its core, is a similar way of getting at the main content of our world's information using a small amount of weight. A picture, in its low-frequency component, can receive different stylistic adjustments by configuring different high-frequency components. Therefore, the core of generalization is to configure different details through the "frame."

At the core of today's large models is the ability to create a matrix of transitions from an "input vector" to the next "vector" through deep learning. With this transition matrix and frame information, the machine can generalize knowledge. As long as humans provide it with similar knowledge from "input vector" to next "vector," it can perform volume generation by imitating the transformation from "input frame" to next "frame," embedding different details. For example, by establishing the attention mechanism of "company and founder," it can generalize from "Steve Jobs and Apple" to "Lei Jun and Xiaomi." Thus, large models perform tasks through imitation and creation, which is very similar

to human beings. Therefore, it is not surprising that large models can generalize with small samples or zero samples.

The magic of "Let's think step by step" is also not surprising, because "Let's think step by step" propels machines to choose frameworks similar to step-by-step thinking to imitate. Deep learning is a neat, elegant path, and the attention mechanism is one of the markers on that path, pointing us in the right direction. "World models" are the fruits of this journey to artificial intelligence.

## 3.5 Could RLHF Finally Solve the Problem of the Larger Model?

There are two serious problems with large models: (1) The problem of toxic content [**?**]. Machine knowledge, whose meaning is difficult for humans to understand but which machines can use, seems not to be a problem. But the problem is actually quite serious. At the heart of the problem is this: humans cannot imitate the form of knowledge networks that machines have built by presenting them with some innate knowledge! This is the nature of the problem. Because a machine cannot be preset with a priori knowledge, it is impossible to preset a machine with some basic requirements in the form of a knowledge network that is impossible to mimic. Without a machine's own needs, it would be impossible for a machine to have self-perceived rewards and punishments. Without self-perceived rewards and punishments, it is impossible for a machine to spontaneously create projections of various things (i.e., combinations of various basis coordinate clusters) into self-rewards or punishments. That is, the basis coordinate cluster created by the machine lacks basic dimensions such as reward, punishment, happiness, and sadness which humans have and must have! Because these dimensions are missing from the basis cluster, it is impossible for the machine to project input information onto these dimensions and identify the information contained in the input. It is also not possible to predict potential rewards or penalties for outputs by projecting onto them using different combinations (i.e., different decision paths for the machine) when preparing combined basis coordinate clusters as outputs.

The current remedy for large models is RLHF. This is equivalent to adding a reward dimension to some training vectors after the fact—that is, adding a reward dimension to the basis coordinate cluster of the machine. If the component value of the reward dimension is added to a large number of different types and sufficient number of vectors in the training data, it is equivalent to establishing the common component combination in these training vectors to the projection of the reward dimension. This is the reward function of the machine. Therefore, the machine can also predict the reward component of the output vector produced by different decisions (i.e., different combinations). Thus, the machine will choose the output with the highest reward component. This is the amazing effect of RLHF learning.

Because the knowledge learned through RLHF can actually be generalized, when

a machine has its own dimensions of reward and punishment, it has its own preliminary "consciousness of seeking benefits and avoiding harm," which is why we can see the shadowy glimpse of "consciousness" from the current large model. Because it has the decision-making tendency to seek benefits and avoid disadvantages, it may have such behavioral tendencies.

But this is a patching-after-the-fact approach, meaning that machines try and humans score and provide feedback, and it can only be used in fields where there is a lot of trial and error. This is similar to a young person graduating with a PhD but having no idea of "right and wrong" at all. Parents can only follow behind, shouting "No," "My God," "Yes" to give them a sense of "right and wrong," and they cannot communicate directly, only through "yes" and "no." Therefore, this kind of learning effect is inefficient, and you may encounter unexpected corner cases forever! (2) Nonsense in a serious manner [**?**].

The attention mechanism is weighted by finding connections between information. The machine obtains the "framework" of how information is organized through attention mechanisms (weights) + deep learning (trial and error). These "frames" are the basis for generalization. The mapping relationship between "frame" and "frame" represents the algorithm from one "vector" to the next "vector." Input "frame" + different details constitutes the specific input vector, and through the algorithm from "vector" to next "vector," you can obtain the output vector—this is the knowledge generalization process.

But it needs to be noted that the machine's mapping from "frame" to "frame" may produce "facts" that do not exist! For example, the machine finds that many journalist profiles are followed by links to other web pages of the journalist's articles or by awards the journalist has won in the past. If the machine sees a lot of this pattern of information organization, it becomes a mapping of "frame" to "frame." So if the input information contains a similar frame but only the name of the reporter is different, then the machine can map from "frame + detail" to "frame + detail," producing many web links or awards in the output. But these web links and awards are also built through other "frame + detail" mappings to "frame + detail"—they probably do not exist!

These are problems that large models struggle to solve. One solution is that RLHF can be used to break up this "frame" to "frame" link mapping so that the machine does not produce links to articles or awards, but this also reduces the power of the larger model. Another option would be to use a search engine to add information related to the input to the user's question, so that the input obtained by the machine contains more details, thus generating more personalized knowledge in the process of mapping "frame + detail" to "frame + detail." Again, this is a palliative solution, because the knowledge gained by search engines is not necessarily correct, and there is a limit to how much knowledge they can acquire for a particular problem.

Therefore, we consider RLHF a solution, but not the final solution.

## 4 Attention Mechanism + Deep Learning + Reinforcement Learning Is the Right Path Toward AGI? Is the LLM the Right Path Toward AGI?

We think the answer is no.

Deep learning obtains an optimized set of coordinate bases from a large number of samples and uses such basis clusters to express vectors. By combining deep learning and attention mechanisms, you can produce optimized coordinate clusters similar to how humans express them. This is the real reason why Transformer can generate intellectual "surge."

In the field of NLP, humans moved from the early bag-of-words model, word vectors to ELMO [**?**], until Transformer truly realized the attention mechanism and seamlessly combined it with deep learning to create an incredible miracle. We have noticed that the path adopted by these technologies is "vectorization first, establishing initial relationship; then through trial and error, adjusting the coordinate basis cluster; and then vectorizing again under the preferred coordinate basis cluster to get the correct relationship." Such a mechanism results in a huge amount of required data, and knowledge is formed in one pass through the training process, making it difficult to update in real time [**?**] [**?**].

First, we have noticed that deep learning differs from human learning in two aspects: (1) It adopts different minimum information elements. Humans use the minimum local features they can perceive as basic elements in the information space. Deep learning, on the other hand, uses easily accessible pixels, syllables, or action patterns as basic elements. These basic elements are actually represented as strings of data arranged in space and time. Therefore, the smallest unit of information (element) in the matrix that deep learning builds, although it may be similar to human elements, may not be the same. Perhaps it could find a more concise and efficient set of elements.

Similarly, deep learning, based on these minimal units of information, can create basis coordinate clusters that are just as inconsistent with human concepts as they are difficult for humans to understand. But it may also create a more concise and efficient set of concepts. So deep learning is really an elegant solution! But it's much better suited to the world of machines. When humans judge machines by human standards, we assume that machines are sometimes mentally retarded.

Secondly, the above problem does solve part of the problem when attention mechanisms are introduced. Through the attention mechanism, deep learning first focuses on relationships among the smallest information units, then creates basis coordinate clusters based on these relationships. However, since the machine is confronted with data, the smallest information unit obtained from the data may still be inconsistent with the smallest information unit of humans, so the "concept" created by the machine may still be very different from that of humans. This is the core problem that prevents the machine from truly

understanding language!

Machine knowledge, which is difficult for humans to understand but which machines can use, seems to be a problem. But the problem is actually quite serious. At the core of the problem is this: humans cannot imitate the form of knowledge networks built by machines, presetting them with some innate knowledge! Since there is no way to preset a machine with innate knowledge, it is impossible to preset a machine with some knowledge of its basic needs, mimicking the form of knowledge networks the machine has built.

At present, the biggest flaw in artificial intelligence is that machines have no demands of their own. Without its own demands, the machine will not generate its own goals. Without spontaneous goals, there can be no spontaneous action. A machine that has spontaneous behavior is a machine that programs itself. A machine that can program itself is a truly intelligent machine. A machine that needs to be programmed externally will always be a machine driven by human intelligence.

How do you establish the demands of the machine? First, the requirements of the machine must themselves be part of the knowledge. Because only then can the machine make decisions based on its knowledge and interaction with the environment to meet its demands. So, demands are knowledge.

To realize that needs are knowledge, we need to mimic the ultimate network form in the memory library by presetting the machine with an innate "least pros and cons" kernel. Through "pros and cons kernel + small sample learning + continuous accumulation," the final form of "fully connected knowledge network with pros and cons information" is achieved. With the "fully connected knowledge network with pros and cons information," the machine can independently predict possible rewards and penalties under various decision paths according to its own knowledge. Thus, the machine can make its own decisions in accordance with the principle of pursuing advantages and avoiding disadvantages. Therefore, all the goals of the machine are created by the machine itself! Only in this way can the machine, in a complex environment, from the top down, according to the specific situation, autonomously create sub-goals, make autonomous decisions, and complete tasks autonomously! Any AI whose process is preset by a program and then executed is still program-driven, whether the program interface is natural language or not. They may hit a brick wall in the real world with many surprises in the future!

With a machine capable of self-programming, it must also be matched with corresponding knowledge to truly realize the process of interactive decision-making with the real environment. At present, the knowledge of interactive decision-making between machine and environment is mainly accomplished through reinforcement learning. And reinforcement learning only works in fields where there's a lot of trial and error. What about areas where trial and error is difficult, like patient care or farming?

More than a decade ago, some of the inventors of our patents discussed the need

to build artificial intelligence based on human learning patterns, based on sample learning. Therefore, at the beginning, they tried to go from "symbol expression" to "common sense + cause-effect logic" to "knowledge network." After trying for a few years, I found that the first step on this road was impassable. Because "symbol expression" → "dog"—how to express it? All the characteristics of "dog" need to be picked out. But a "dog" can be an animal or a person! It can be "a character to be sung about" and it can be "a character to be despised"… So the essence of "dog" is the sum total of "dog" in relation to everything else. This is a definition of "dog" modeled after Marx's definition of man. So "dog" must be put into the whole network of knowledge, defined by its relationship to all other knowledge. Thus, Symbolism faces a huge challenge! Because "dog" cannot be separated from other knowledge! "Fully connected knowledge networks" similar to deep learning must be built—this is our first conclusion.

Because the "dog" must be placed within the whole knowledge network, defined by its relationship to all other knowledge, you have to have enough knowledge to make this whole thing clear. So "the amount of knowledge has to be sufficient" to have enough background knowledge to understand what a dog is. That's our second conclusion. When we look back, isn't that what large models do? "Attention mechanism + deep learning" creates the fully connected network; the large model does "use a lot of knowledge to build a fully connected knowledge network."

So why don't we see robots walking around the streets? Because knowledge networks alone won't do! They must also be able to "interact with the environment and make continuous decisions"! The current AI, on the other hand, relies on reinforcement learning to train its decision knowledge to interact with the environment. The AIXI algorithm, which is essentially the most idealized reinforcement learning algorithm, requires more computing and predictions than there are atoms in the universe—impossible to achieve. "AlphaGo" uses the AIXI algorithm and trims the amount of computation through "Monte Carlo Tree" to reduce the computation needed to play Go. So one possible path to general artificial intelligence is: large model + AIXI algorithm (the strong reinforcement learning algorithm).

So why haven't we seen rollout robots that walk the streets? The central obstacle to this path is that the AIXI algorithm requires two preconditions [**?**]: (1) The machine needs to know the reward information it can obtain under different decision paths. (2) The machine needs to search all decision possibilities through traversal.

These two conditions can be perfectly satisfied in a game. Because in the end, winning or losing is the reward function. By playing the game ten million times, the machine can sum up the pros and cons of each decision. And the decision knowledge search space of the game is limited, so the computer power required by the machine is capped. However, in real life, one only lives once, and it is impossible to acquire interactive decision-making experience by replicating endlessly. And unlike games, in the face of a task, there is no clearly defined

scope of information search! So training a machine to play a game, a meta-universe, and generate words and images can be trial-and-error, but what about driving a car, cooking a meal, caring for a child, or caring for a sick person in a real environment? These areas cannot be trial-and-error, and the current artificial intelligence technology roadmap cannot solve them!

## 5 What Is the Right Road Forward to AGI?

We believe that true "general artificial intelligence" through true "machine learning" requires three prerequisites:

**Prerequisite 1:** Sufficient knowledge + full internet connection, no plugins! Any plugin, and knowledge network cannot be integrated, prone to accidental mental retardation!

**Prerequisite 2:** Let the machine predict reward and punishment information in various decision paths! So the machine has to be like a human: it can predict the rewards and penalties for various decision paths by itself, with only a few attempts; it doesn't have to try everything a million times!

**Prerequisite 3:** Machines can learn directly from the accumulated experience of humans! The nature of current "reinforcement learning" technology is trial and error. It should be called "reinforcement evolution"! It took hundreds of millions of years for human beings to evolve from the intelligence of a single-celled organism to today! Therefore, machines must be able to directly learn the accumulated experience of human civilization history and cannot go down the old road of "evolution"!

It took us ten years to propose a set of technical solutions [**?**] [**?**] [**?**] [**?**] under the guidance of these three preconditions required for truly "general artificial intelligence." It realizes true AGI through true machine learning, which mainly includes: (a) To build a set of fully connected knowledge networks that human beings can understand. The way to build it is through the mechanism of attention and the mechanism of memory and forgetting. Memory and forgetting mechanisms mainly achieve statistical correlation, while the attention mechanism of information is based on statistical correlation, through the chain association activation process, through multi-path accumulation of active values, through the extinction of active values over time to achieve. (b) Because our fully connected knowledge network is an intelligible form of network organization (it is, in fact, a database), we can mimic the ultimate network form in a memory bank by presetting machines with an innate "least pros and cons" kernel. Through "pros and cons kernel + small sample learning + continuous accumulation," the final form of "fully connected knowledge network with pros and cons information" is achieved. With the "fully connected knowledge network with pros and cons information," the machine can independently predict possible rewards and penalties under various decision paths according to its own knowledge. Thus, the machine can create its own goals and make its own decisions in accordance with the principle of pursuing advantages and avoiding

disadvantages. Only in this way can the machine, in a complex environment, from the top down, according to the specific situation, autonomously create subgoals, make autonomous decisions, and complete tasks autonomously! Any AI whose process is preset by a program and then executed is still program-driven, whether the program interface is natural language or not. They may hit a brick wall in the real world with many surprises in the future! (c) The environment of the machine is very different, and the task of the machine is also very different, so the machine cannot be in any environment, through training to get any task-related interactive decision-making knowledge! This is an impossible task!

So we have to find a new way. Take inspiration from human learning. In fact, when humans are faced with a task, all decisions are made around the core of advantage and disadvantage avoidance. That's where avoidance, rejection, and seeking more help come in. These behaviors are essentially new behaviors created by humans. Instead of tackling tasks directly, humans turn any task into one task of "how to meet their needs." Machines must go the same way. The process of making machines learn "how to satisfy their own demands" as they go about their daily lives. Therefore, for any task, the machine converts it into the "how to meet its needs" task according to its own needs. And on this task, it has a great deal of experience to generalize, because all of its learning is centered around this task. So, the solution we propose is that the learning process of the machine should not be task-oriented, but should be oriented to the needs of the machine itself. If the machine has its own needs and the knowledge of how to meet those needs, then the machine can create new behaviors to meet those needs. In other words, the machine can "program" itself, and the task of programming is only "how to meet its own needs," and all the knowledge of the machine is built around "how to meet its own needs." Therefore, our machines can complete any task.

It is in the process of "how to meet its own needs" that the machine also completes the specific tasks given by human beings. The result of completion may be "done," "rejected," or "further seek more information to evaluate." A machine is safe if it is preset with multiple innate needs for positive feedback from humans. And it will actively take steps to break down tasks and complete them because the machine is always learning "how to satisfy its own needs" to complete a task, and it is also always dealing with "how to satisfy its own needs" to complete a task. Completing specific tasks is a by-product of doing the "how to satisfy your needs" task.

The way machines organize their knowledge involves presetting their needs. Then, the knowledge created by the machine needs to include knowledge related to the requirements. What is knowledge? It is the way common features are arranged in space and time! What is requirement-related knowledge? It is the arrangement of common features in time and space that contains information about the requirements of the machine. The arrangement of common features in time and space, if it includes the needs of the machine, is subjective common sense—that is, the relationship between "the world" and "me." The arrangement

of common features in time and space is objective common sense if it does not include the needs of the machine—that is, the relationship "between outer everything." So, the way common features are arranged in space and time is common sense. Common sense is at the heart of an AGI. With common sense, machines will take the initiative to solve tasks and create processes according to their own needs. In other words, machines program themselves! Now that's true intelligence! And the current large model + everything APP road still hasn't divorced itself from the way of human programming! (d) The machine needs to integrate the knowledge network + machine demand + value evaluation into the same network. In daily life, machines are constantly learning "how to meet their own needs." So it has a lot of experience that it can generalize to this question. If we preset the machine with a variety of preconceived requirements for positive feedback from humans, then the machine will be safe and will take the initiative to break down and complete the task step by step.

Innate needs need to include the machine's own operational needs so that the machine will maintain its own operations. They also need to include preset human needs for the machine, such as the machine's desire for positive feedback from humans, just like a human child. In this way, we can interact with machines and train them from an early age to align their values with those of humans. To establish the machine's instinctive needs, we also establish the machine's higher-order demands, such as "morality" and "obeying the law." At the same time, we can also preset a small amount of innate knowledge, which is mainly used in trial-and-error domains, such as minimal "cliff avoidance" knowledge. In this way, we create a child who has needs, who is selfish, and who has a small amount of instinctive knowledge related to survival, but who craves human approval. It has an innate language for communicating with humans (such as innate knowledge of nodding or shaking one's head). Then, based on an innate language for communicating with humans, humans can slowly develop more complex ways of communicating with machines, such as language.

Then, by learning in the real world, machines can acquire both knowledge through self-summary and direct human knowledge through language learning. They may also use their unmatched ability to discover common features arranged in ways that are not obvious to humans, even though the patterns of arrangement in time and space are not obvious to humans. But a machine can find them through statistics, and it can imitate the way humans use symbols to express common permutations and symbols to express newly discovered common permutations. This is the new knowledge that machines create.

It's an iterative process. The machine programs itself to discover more knowledge, and it will develop into super intelligence!

Machines need to learn through small samples and accumulations. Only in this way can knowledge be updated in real time. In real life, a large number of tasks need to be completed step by step through machine-environment interaction. Therefore, any feedback from the environment must immediately become the basis for the machine's next decision. And this decision knowledge needs to

be updated immediately into the machine's knowledge base. Otherwise, the machine will make the same mistake next time. At present, human intelligence uses big data samples, and knowledge is mainly accomplished through a single training session. Even fine-tuning for a task cannot be updated in real time. Therefore, the real learning path should be small-sample, cumulative learning. This way of learning, which is more similar to human learning, can realize real-time updates.

## 6 The Evolution Direction of Artificial Intelligence

We believe that the development of artificial intelligence can be roughly divided into different stages: (1) The stage before the realization of true consciousness can be considered the "feature exploration" stage. Before deep learning, this was mainly concentrated in the "human worker exploration" stage. Artificial exploration can be "expert system," "knowledge encyclopedia," "probability statistics," etc. After deep learning, the focus shifted to the "machine exploration" phase, allowing the machine to "discover characteristics" from large samples. (2) After the implementation of true attention (Transformer), the machine can be considered to have achieved "knowledge generalization" as its "knowledge" and human "knowledge" are initially aligned. In tasks from humans, the machine can exhibit some intelligence through "knowledge generalization." (3) In the future, we think AI needs to evolve to the next stage: the stage of "autonomous interaction." "Autonomous" means that the machine is no longer a silent "machine"; it is capable of generating behaviors spontaneously (which is equivalent to programming itself), and the machine will seek out knowledge on its own (such as actively interacting with the environment to acquire knowledge). "Interactive" means that a machine can interact with its environment in real time, updating its knowledge in real time, making continuous decisions, and performing complex tasks in unfamiliar environments.

The core of achieving "autonomous interaction" is that machines should have their own demands. The needs of the machine must be part of the machine's knowledge, so that the machine can use its knowledge to create behavior to satisfy its own demands. And the core of realizing machine requirements is to first create a knowledge network that human beings can understand. Only in this way can human beings imitate the form of knowledge network and preset the requirements of machines. Then let the machine learn around its own needs, thus establishing all connections between information and needs. So the machine's knowledge is all about its demands. In this way, the machine can transform any task into a single task: how to satisfy itself. And all the machine's exploration and learning process revolves around the task of "how to satisfy its own needs." Therefore, when a machine is faced with "how to meet its own needs," it has a great deal of experience that can be generalized. Only in this way can machines handle tasks that are difficult to try and error. In fact, we believe that humans use similar methods to acquire knowledge and deal with problems.

It is a paradigm shift for AI to be oriented toward "its own demands" rather

than toward "tasks from humans." We believe this paradigm shift is necessary. Because external tasks are so diverse, many of them have to interact with the real world. They are hard to trial and error, hard to get big data samples, and hard to gain decision-making knowledge through reinforcement learning. In addition, training machines for each type of task in a real environment is an impossible task.

Artificial general intelligence is the beginning of AI and also the crown of AI. We put forward a set of technical solutions to realize AGI a few years ago. In references [**?**] [**?**] [**?**] [**?**], we present detailed technical details for achieving this path, which may be a viable path to lead mankind toward AGI.

In this scheme, the requirements of the machine are preset by human beings and can have multiple requirements, so the goals generated by the machine are also multi-objectives. At present, artificial intelligence is single-goal. From the point of character, it can be regarded as the artificial intelligence of "All means necessary for the goal." Because this kind of artificial intelligence only pursues a single goal and does not think about anything beyond the goal, it is very dangerous! In our scheme, the multi-objective also includes the alignment of human values, including the needs of "morality," "rules and laws," and "recognition." Therefore, in our scheme, the machine will comprehensively consider the needs of "morality," "rules and laws," and "recognition." Therefore, our scheme is a feasible way to solve the security problem of artificial intelligence.

## 7 A New Approach to AGI

We believe that from the point of view of information, all information in the world is a multi-dimensional information matrix. Knowledge is another way of describing this information matrix. This is essentially the same way that we can describe the same matrix in the time domain or in the frequency domain.

Common knowledge, or "common sense," is a description of the common tokens arrangement in the information matrix. By adopting these coordinate basis clusters, we can conveniently describe common vectors in matrices. Therefore, using "common sense" to describe information in the matrix is a concise description of the matrix information, which is essentially a form of information compression. These concise descriptions are the "principal components" of the knowledge matrix, which is a kind of frame information. They are similar to the low-frequency principal components in image information.

The nature of the attention mechanism is to find common token arrangements in the information matrix and assign weights to these common token arrangements. If this Tokens information is one-dimensional, such as language information, then the attention mechanism will find common language combinations. With it, we can generalize language. If this Tokens information is two-dimensional, such as image information, then the attention mechanism will find common image token combinations. With it, we can achieve image generalization. If this Tokens information is three-dimensional, such as stereoscopic spatial informa-

tion, then the attention mechanism will find common image token combinations. With it, we can achieve generalization of spatial structure. If this Tokens information is 3D + time dimension, such as process information, then the attention mechanism will find common process token combinations. With it, we can implement process generalization. And the generalization of processes will bring unmatched new capabilities to machines.

So, the attention mechanism solves the problem of "objective common sense." "Objective common sense" is the relationship between external information created by the machine. But compared to human common sense, large models are currently unable to create "subjective common sense." "Subjective common sense" is the relationship between "external information" and "machine's own requirements." Human beings understand the world more from "subjective common sense" to gradually understand the world. And objective common sense serves "subjective common sense" more.

At present, large models cannot create "subjective common sense." The core problem is that they do not have "machine's own needs." At present, humans can only add a "reward and penalty" dimension to the information space through RLHF as an ex post remedy. The robots created in this way act in a "one-track thinking" way: they think only about a single goal, not about the right means. It would be very, very dangerous if they took over our lives!

We propose a new machine learning approach. It takes the "connectionism" road, uses small samples, cumulative learning, and achieves the same attention mechanism as the Transformer. But because it uses cumulative learning, its knowledge can be updated in real time. This technical solution has two main features: (A) Small samples, cumulative learning, real-time update of knowledge. (B) Machines have their own requirements.

Thus, we accomplish three things: 1) Small samples, cumulative learning, achieving the same goal, the attention mechanism and fully connected network, with knowledge that can be updated in real time. 2) Knowledge network + machine demand + value assessment are integrated into the same network. 3) The machine transforms all tasks into a single task of "how to satisfy its own requirements."

Therefore, we solve the following problems of current AI: (1) Lack of common sense: especially subjective common sense. So when faced with a new task, you need to learn by trial and error. There is no way to predict pros and cons, so only trial and error is possible, which limits the application area. (2) In the face of complex tasks, it cannot autonomously find additional help to solve the problem: because the machine has no own needs, it will not have autonomous goals; without autonomous goals, there will be no autonomous decision-making; without autonomy, you don't actively seek out extra help to make decisions. A robot that does not create tasks autonomously will not be an AGI. (3) The problem of lack of real understanding of language: because our knowledge is an arrangement of tokens in space and time. The temporal and spatial arrangement

of multi-modal tokens activated by input tokens (e.g., linguistic tokens) can be imitated. So machines can learn from human experience through language, and when faced with unfamiliar tasks, they may succeed on the first attempt. (4) Problem with limited application domains: The application domain of our machine is not just "content generation." The machine converts any external task into a process of "how to satisfy its needs." The machine in daily life is constantly learning the "how to meet its own needs" process. Thus, it has a large amount of experience that can be generalized. Machines perform external tasks in the process of "satisfying their own needs," similar to humans. (5) Self-learning: Currently, when AI sees its owner fall, it doesn't come to help. But our machines have their own needs and their own initiatives. They will arrange tasks for themselves, which is equivalent to programming themselves, realizing autonomous learning and self-iteration. (6) Safety problem: Our machine's "needs" are given by humans. Humans can train the machine through a variety of "needs" so that its values align with human values. The machine achieves multi-objective balance, so its safety is far better than current artificial intelligence. (7) Here, we briefly introduce the implementation process of our scheme in the following. More detailed implementation steps will be disclosed in subsequent documents.

The implementation process of our scheme: 1) The integration of "knowledge network + machine demand + value evaluation" is our core technology. The implementation scheme is to imitate the final network form in the memory library and preset an innate "minimum pros and cons" kernel to the machine. Why can we preset a priori knowledge? Because of our knowledge network, humans can understand! Large models can't do that now! 2) Through the "pros and cons kernel + small sample learning + continuous accumulation," the "fully connected knowledge network with pros and cons information" is finally formed. 3) With the "fully connected knowledge network with pros and cons information," the machine can autonomously predict possible rewards and punishments under various decision paths based on its knowledge. So the machine can make its own decisions according to the principle of seeking benefits and avoiding disadvantages. So all the targets of the machine are created by the machine itself! Only in this way is it possible for the machine to autonomously create sub-goals on the spot, autonomously find new information to assist decision-making, autonomously find relevant tools, autonomously make decisions, and autonomously complete tasks in a complex environment, from top to bottom, according to specific circumstances! So our machines can program themselves, self-iterate, and self-evolve. 4) When information input (external information or machine self-monitoring information) is input, some reward symbols and punishment symbols are activated. Each activation path from the input to the reward symbol and the punishment symbol is a logical link that potentially generates a reward or punishment. If on this logical link, every underlying feature is truthfully realized, then the reward or penalty propagated by this logical link is also realized. So the response of the machine to any input information is the same: increase the probability of the reward logic chain and decrease the

probability of the punishment logic chain, so as to achieve the purpose of seeking advantages and avoiding disadvantages. 5) How to increase the reward link and reduce the probability of the penalty link? It is to increase or decrease the realization probability of the underlying feature with a high activation value on the link. Tokens with high activation values on a bridge are the tokens with high weight for this link. When they are true, then the activations propagated along this link are true, so the reward or penalty that is eventually activated is also true. How does it work? From the activation value transfer path of input information → reward and punishment symbol, the N bottom features with the highest activation values are selected, which are the top realization path that leads to reward or punishment. The goal of the machine is to: 1) implement the tokens on the reward path (that is, mimic past experience and make them appear in the input). Make the tokens on the penalty path unattainable (that is, avoid them from the input by mimicking past experience). 6) The policy space search of the machine is bounded in the set of all tokens that are activated. The machine estimates the potential reward value by passing the activation value of each input → reward/penalty symbol. So our machine, even during training, pre-estimates the reward value. However, current artificial intelligence, in the training process, gets the reward value after the fact. This determines that the policy-states knowledge of AI and environment interaction decision-making must come from a lot of trial and error process.

## Conclusion

After analyzing the capability boundaries of the LLM, we propose a new machine learning scheme to break through the current LLM capability boundaries. We believe it may be a feasible path to create a machine that can create tasks autonomously, iterate and evolve itself. Moreover, the machine learns during its life and updates its knowledge in real time. It can directly access all the experience accumulated in the history of human civilization through human language.

In references [**?**] [**?**] [**?**] [**?**], we propose a specific implementation plan, and we will provide more details in the future.

## References

[1] Reed S, Zolna K, Parisotto E, et al. A Generalist Agent[J]. 2022.
[2] Optimizing Language Models for Dialogue. https://openai.casa/blog/chatgpt/
[3] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022.
[4] Vinyals O, Ewalds T, Bartunov S, et al. StarCraft II: A New Challenge for Reinforcement Learning[J]. 2017.
[5] Quanjun Zhang, Chunrong Fang, Yuxiang Ma, Weisong Sun, Zhenyu Chen. A Survey of Learning-based Automated Program Repair. arXiv:2301.03270, 2023

[6] Antonio Mastropaolo, Luca Pascarella, Emanuela Guglielmi, Matteo Ciniselli, Simone Scalabrino, Rocco Oliveto, Gabriele Bavota. On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot. arXiv:2302.00438, 2023
[7] Introducing ChatGPT. https://openai.com/blog/chatgpt/
[8] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, Lijuan Wang. arXiv:2210.09150, 2023
[9] An experimental open-source attempt to make GPT-4 fully autonomous. https://GitHub.com/Significant-Gravitas/Auto-GPT
[10] Zhuosheng Zhang, Aston Zhang, Mu Li, Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493, 2023
[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022), 2022a. https://arxiv.org/abs/2201.11903
[12] AUTO GPT, https://auto-gpt.ai/
[13] ERNIE Bot, https://mp.weixin.qq.com/s/0-8X9FPouteKzNiK6DPaiA
[14] Mccann B, Bradbury J, Xiong C, et al. Learned in Translation: Contextualized Word Vectors[J]. 2017.
[15] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. arXiv, 2017.
[16] Cheng J, Dong L, Lapata M. Long Short-Term Memory-Networks for Machine Reading[J]. 2016.
[17] Lin Z, Feng M, Santos C, et al. A Structured Self-attentive Sentence Embedding[J]. 2017.
[18] Parikh A, Tckstrm O, Das D, et al. A Decomposable Attention Model for Natural Language Inference[J]. 2016.
[19] Paulus R, Xiong C, Socher R. A Deep Reinforced Model for Abstractive Summarization[J]. 2017.
[20] Tamkin A, Brundage M, Clark J, et al. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models[J]. 2021.
[21] Kodge S, Roy K. BERMo: What can BERT learn from ELMo?[J]. 2021.
[22] Lee H, Xu G. Optimizing Policy via Deep Reinforcement Learning for Dialogue Management[C]// 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE Computer Society, 2018.
[23] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022.
[24] Sunehag P, Hutter M. Principles of Solomonoff Induction and AIXI[C]// Ray Solomonoff memorial conference. Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia; Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia, Department of Computer Science, ETH Zurich, Switzerland, 2011.
[25] Yongcong Chen, Ting Zeng, Xingyue Chen. Method for implementing universal artificial intelligence. CN111553467A[P]. 2020.
[26] Yongcong Chen, Ting Zeng, Xingyue Chen. A Machine Intelligence

Implementation Method Imitating Human Intelligence, CN111582457A[P]. 2020.
[27] Yongcong Chen, Ting Zeng, Xingyue Chen. Establishment of general artificial intelligence system, CN112016664A[P]. 2020.
[28] Chen Y, Zeng T, Chen X. ESTABLISHMENT OF GENERAL-PURPOSE ARTIFICIAL INTELLIGENCE SYSTEM:, US2022121961A1[P]. 2022.

*Note: Figure translations are in progress. See original paper for figures.*