SCIENCE CHINA
Information Sciences

• REVIEW •

February 2023, Vol. 66 121101:1–121101:19
https://doi.org/10.1007/s11432-021-3449-x

Toward the third generation artificial intelligence

Bo ZHANG*, Jun ZHU & Hang SU

Institute for Artificial Intelligence, Tsinghua University, Beijing 100084*, China*

Received 9 September 2021/Revised 24 December 2021/Accepted 9 February 2022/Published online 12 January 2023

Abstract There have been two competing paradigms in artificial intelligence (AI) development ever since its birth in 1956, i.e., symbolism and connectionism (or sub-symbolism). While symbolism dominated AI research by the end of 1980s, connectionism gained momentum in the 1990s and is gradually displacing symbolism. This paper considers symbolism as the first generation of AI and connectionism as the second generation. However, each of these two paradigms simulates the human mind from only one perspective. AI cannot achieve true human behaviors by relying on only one paradigm. In order to develop novel AI technologies that are safe, reliable, and extensible, it is necessary to establish a new explainable and robust AI theory. To this end, this paper looks toward developing a third generation artificial intelligence by combining the current paradigms.

Keywords artificial intelligence, symbolism, connectionism, dual-space model, single-space model, triple-space model

Citation Zhang B, Zhu J, Su H. Toward the third generation artificial intelligence. Sci China Inf Sci, 2023, 66(2): 121101, https://doi.org/10.1007/s11432-021-3449-x

1 First-generation artificial intelligence

How is intelligent human behavior formed? Newell and Simon et al. [1–4] proposed a symbolic model to simulate such behavior based on the physical symbol system hypothesis (PSS hypothesis). The hypothesis suggests that intelligent human behaviors can be interpreted as instances of physical symbol systems, which include (1) an arbitrary set of symbols and rules that manipulate the symbols; (2) the manipulators are purely syntactic, i.e., they are concerned only with syntax rather than semantics, and the manipulations consist of the combination and reconstruction of symbols; (3) the syntax has a systematic semantic interpretation, i.e., the symbol is associated with some object and its attributes. In 1955, McCarthy et al. [5] outlined the basic idea of symbolic AI as follows—"It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning and rules of conjecture"—in the Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. They proposed a reasoning model based on knowledge and experience for intelligent human behaviors. Therefore, symbolic AI is also called a knowledge-driven method.

The founders of symbolic AI initially focused on universal methods of reasoning (or search), such as mean-end analysis, divide-and-conquer, and trial-and-error. They tried to solve a wide range of practical problems with these methods. Nevertheless, the general methods are always weak, and can only solve simple "toy world" problems, such as a robot laying out blocks, playing tic-tac-toe, or chess. Most of the previous efforts that aimed for artificial general intelligence (AGI) met with failure, which made symbolic AI fall out of favor in the 1980s. Fortunately, Feigenbaum at Stanford University and others [6] gave up the pursuit of artificial general intelligence just in time. They believed that knowledge, especially domain knowledge, is the key to intelligent behaviors and presented a set of strong AI methods such as knowledge engineering and expert systems, which brought new hope for symbolic AI. They have developed various expert systems, including DENDRAL (1965–1975) [6], MYCIN (1971–1977) [7], and XCON. Both Feigenbaum and Raddy won the 1994 ACM Turing award as the pioneers in designing and constructing large-scale AI systems.

* Corresponding author (email: dcszb@mail.tsinghua.edu.cn)

However, most of these early expert systems were too simple and far from practical. This remained the case until May 1997, when IBM Deep Blue defeated the chess champion Kasparov. This marked a new stage of symbolic AI as being truly capable of solving problems with the development of a large-scale system.

Symbolic AI can also be applied to machine learning such as inductive learning. Taking inductive logic programming (ILP) [8] as an example, we will show the mechanism of symbolic AI-based machine learning. In ILP, machine learning is considered as knowledge-based inductive learning, i.e., searching for a hypothesis in the hypothesis space. The hypothesis should include as many positive examples as possible, exclude negative examples as much as possible, and be consistent with background knowledge. In ILP, the positive and negative samples (instances), background knowledge, and hypotheses are expressed in the form of first-order logic clauses (programs). Generally, the hypothesis space is huge, which makes ILP learning quite difficult. In some cases, the background knowledge can effectively restrict the hypothesis space to make learning more feasible. Obviously, an agent can learn more effectively and efficiently if it has more background knowledge. Moreover, Riguzzi et al. [9] proposed probabilistic inductive logic programming (PILP) to deal with uncertainty.

With background knowledge appropriately harnessed, knowledge-driven methods are robust, and are more effective when the data is insufficient and can be easily transferred to different domains. Take transfer learning [10] as an example. One can update or transfer the knowledge learned from one scenario to another to achieve cross-domain or cross-task generalization. Specifically, we can first identify the general knowledge that can be transferred across domains or tasks, along with the specific knowledge for a single domain or task from the training environment (including training data and methods). In this case, the general knowledge can be used to improve the performance of the target domain or task. The general knowledge can be transferred to the target domain in the following ways: (1) examples available in the source domain, (2) features that can be shared between the source and target domains, (3) available parts of the source domain model, (4) specific rules shared between entities in the source domain. Therefore, knowledge plays an essential role in transfer learning, which is the reason that knowledge-based methods can be easily generalized across different domains and tasks.

Besides Feigenbaum and Reddy (1994), Minsky (1969), McCarthy (1971), Newell and Simon (1975) have also made great contributions to the creation of symbolic AI, and won Turing award successively (the years are indicated in brackets). Take the IBM Deep Blue program as an example. The success of the first-generation AI stems from the following aspects. (1) Knowledge and experience. The evaluation function was split into 8000 parts (parameters). The system determined the optimal values for these parameters by analyzing over 4000 positions and 700000 grandmaster games. It also uses the endgame database which includes a large number of six-piece endgames and five or fewer piece positions. Additionally, the program's chess knowledge was fine-tuned by grandmaster Joel Benjamin. (2) Algorithm. Deep Blue executes the α-β search algorithm in parallel, which effectively improved the search efficiency. (3) Computing power. Using IBM RS/6000 SP2 with 11.38 GFLOPS, Deep Blue can examine 200 million moves per second or 50 million positions within three minutes.

Symbolic AI has a solid foundation in cognitive psychology. It gains strength by taking the symbol system as the model of human higher mental activity. Symbols are characterized by compositionality, i.e., simple atomic symbols may combine and form complex symbol strings. Since each symbol corresponds to a semantic meaning, the compositionality of symbols objectively reflects that of semantic objects, such as how to combine simple components into a whole. Composability is the basis for reasoning. Therefore, symbolic AI is as interpretable as human rational behaviors, which makes it easy to understand. Yet, symbolic AI also has obvious limitations. At present, the existing methods can only solve deterministic problems with complete information in a structured environment. IBM's Deep Blue chess program, which embodies the achievement of such methods, beats humans in a perfect information game-the simplest case of games. However, human cognitive behavior, such as decision-making, usually takes place in a complex environment with imperfect information. Symbolic AI is far from solving such problems. For example, it is difficult for a computer to process the human knowledge represented in natural language (by discrete symbols). Therefore, it is necessary to develop new technologies of knowledge representation. Existing knowledge representation methods, such as production rules or logic programs, are easy for computers to handle. However, these methods are simple but limited in their expression, which makes it difficult to describe more complex and uncertain knowledge. Moreover, the current research is mainly limited to logical reasoning and other deterministic methods. More complex knowledge representation and reasoning methods, such as knowledge graphs [11] and probabilistic reasoning [12], remain open problems.

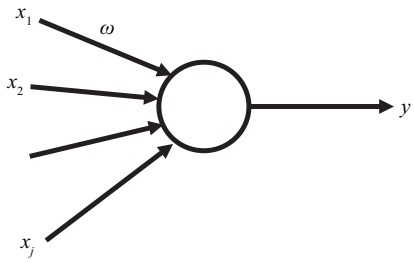

Figure 1 Perceptron.

In general, mathematical tools for the processing of discrete symbols are lacking. It is non-trivial to apply mathematical tools other than mathematical logic, which is one of the main reasons that it is difficult to execute symbolic AI efficiently on computers. Knowledge-driven strong AI can only solve specific problems on a case-by-case basis. It is still a controversial issue as to whether there is a general weak approach that can be applied to a wide range of problems, namely artificial general intelligence. In addition, the acquisition of knowledge from original data (including text, images, speech, and video) is mainly operated manually at present, and therefore is of low efficiency. Therefore, it is important to explore effective automatic methods of knowledge acquisition.

Moreover, the current intelligent systems cannot understand the world and behave reasonably in an unpredictable environment due to the absence of common sense, which is recognized as one of the most significant barriers between the current AI algorithms and a more general AI system. How common sense is acquired, expressed, and inferred remains a challenging problem. Admittedly, the construction of a practical common-sense library is akin to the "Manhattan Project" of AI since the amount of common sense is huge and introducing it into AI will require tremendous amount of time and effort.

2 Second-generation artificial intelligence

How is sensory information (e.g., vision, audio, and touch) stored in memory and how does it affect human behaviors? There are two schools of thought. One is that this information is represented in some coded way in memory (in neural networks), with the representative sample of research on the symbolic AI. The other is that sensory stimuli are not stored in memory, but instead create stimulus-response connections (channels) in neural networks that lead to intelligent behaviors. This is known as the connectionist claim, which is the foundation for connectionist AI. In 1958, Rosenblatt [13, 14] built a perceptron—the prototype of an artificial neural network (ANN)–based on the idea of connectionism. The inspiration for the perceptron comes from two sources. One is from the mathematical model of neurons proposed by McCulloch and Pitts [15] in 1943—the "threshold logic" circuits, which convert the input of neurons into discrete values, usually known as the M-P model. The other is from the Hebb learning rule proposed by Hebb [16] in 1949, which is that "simultaneously firing neurons are connected together". The perceptron function is shown in Figure 1, which can be formulated as

$$y = \begin{cases} 0, & \text{if } \sum\limits_j w_j x_j \leqslant b, \\ 1, & \text{if } \sum\limits_j w_j x_j > b, \end{cases} \tag{1}$$

where b is the threshold value and w is the weight.

The founders of AI were aware of the idea of connectionism from the very beginning. For example, McCarthy et al. [5] raised the question "How can a set of (hypothetical) neurons be arranged so as to form concepts?" and states "Considerable theoretical and experimental work has been done on this problem. ... But the problem needs more theoretical work". The perceptron has only one hidden layer, which is too simple. Minsky and Papert [17] pointed out that the perceptron can only solve linearly separable problems; even if the number of hidden layers is increased, it is difficult to use the perceptron because there is no effective learning algorithm.

This criticism of the perceptron was fatal, which resulted in the nascent connectionism AI falling to a low point for more than a decade. During this difficult period, with the continuous efforts of many researchers, great progress has been made in both neural network models and learning algorithms over

the past 30 years, gradually forming mature theories and technologies of deep learning. The important advances in this period can be summarized as follows. The first is gradient descent, an algorithm that was originally proposed by the French mathematician Cauchy [18] in 1847, and improved as an enhanced and usable version by the Russian mathematician Nesterov [19] in 1983. The second is the BP (back propagation) algorithm, which is customized for ANN. BP was first proposed by Linnainmaa [20], a Finnish student, in his master's thesis in 1970; in 1986, Rumelhart et al. [21] conducted a systematic analysis. These two algorithms infuse new power into ANNs' learning and training. Together with threshold logic and the Hebb learning rule, they constitute the four pillars of ANNs. In addition to the four pillars, there is a series of important studies, including better loss functions (e.g., the cross-entropy cost function [22]), algorithm improvement (e.g., regularization to prevent overfitting [23]), new forms of networks (e.g., convolution neural networks (CNN) by Fukushima (Japan) [24, 25] in 1980, recurrent neural networks [26], long short-term memory neural networks [27], and deep belief nets (DBN) [28]). These methods together have opened a new era of second-generation AI based on deep learning [29].

Data-based learning has a solid mathematical foundation. To illustrate this foundation, let us take a simple example of supervised learning, which can be formalized into the following regression problem. We extract samples $(x_i, y_i) \overset{\text{i.i.d.}}{\longleftarrow} (X, Y)$ from database D. The purpose is to estimate the input-output mapping function as $f : X \to Y$ in the samples. That is, the algorithm searches for and selects a function f^* from the function family (hypothesis space) $F = \{f_\theta : X \longrightarrow Y; \theta \in A\}$ such that it is close to the true f on average. In deep learning, this alternative function family is expressed as a deep neural network described by

$$f^* = \arg\min_{f_\theta \in F} \mathbb{E}_D \left[l \left(f_\theta(x), y \right) \right], \tag{2}$$

where l is the loss function.

In general, there are three basic assumptions in parametric learning. (1) Independence: the selection of the loss function and function family F (or neural network structure) is data-independent. (2) Large capacity: there is a huge number of samples, i.e., (x_i, y_i) $i = 1, \ldots, n, n \to \infty$. (3) Completeness: the training samples are complete and noiseless. If all the assumptions are satisfied, f^* will eventually converge to the optimal function f as we have an increasing number of training samples. If we have enough data with a sufficient level of quality, f^* can approximate any function because of the universality of deep neural network.

Therefore, using deep learning to find the function behind the data has a solid theoretical basis. This assertion has been borne out in many practical applications. For example, in the standard image library ImageNet (20000 categories, 14 million images), the error rate of image recognition was as high as 50% in 2011. By 2015, Microsoft had reduced the error rate to 3.57% by using deep learning methods; this is even lower than the typical human error rate of 5.1% [30]. The speech recognition rate with a background of low noise was generally about 80% before 2001, but it reached over 95% in 2017, meeting the requirements of commercialization. In March 2016, the Google AlphaGo program defeated world Go champion Lee Sedol, which represented another pinnacle of the second-generation AI. Go is known as one of the most challenging games for AI because of its complexity; before 2015, the strongest Go computer program could only play at the level of a human at the amateur 5 dan level. More impressively, these results require no domain knowledge, but only the input of a Go board as an image.

The success of AlphaGo can be attributed to the following factors. The first is data. AlphaGo-Zero has taught itself tens of millions of Go games through reinforcement learning. In comparison, human Go masters have played only 30 million effective games over 1000 years. The second is algorithms, including Monte-Carlo tree search [30], deep learning, and reinforcement learning [31]. The third is computing power. AlphaGo ran on a distributed computer system composed of 1920 CPUs and 280 GPUs. Therefore, the second generation of AI is also called data-driven methods. Among the scholars who have made significant contributions to the creation of second-generation AI, the following five have received the Turing award. They are L. G. Valiant (2010), J. Pearl (2011), and Y. Bengio, G. G. Hinton, Y. LeCun (2018). The year of the award is indicated in the brackets.

However, since 2014, many limitations of deep learning methods have been revealed, indicating that its path to AI has encountered a bottleneck. Take an example, which our group discovered, that illustrates the vulnerability of the deep learning method. In [32], we present the results of an attack on the Inception V3 deep network using the momentum iterative fast gradient sign method (MI-FGSM). A noiseless original image called Albis Mons was correctly classified by the deep network with 94.39% confidence. We used the MI-FGSM method to generate adversarial noise after 10 iterations, and then added the noise to the

original image, yielding an adversarial sample accordingly. Due to the small amount of noise added, the resulting adversarial sample is almost indistinguishable from the original image from human observation. However, the deep network identified the adversarial sample as a "dog" with 99.99% confidence.

Why is deep learning so vulnerable, unsafe, and easily deceived? The reason can only be found in deep learning itself. The success or failure of deep learning is closely related to the above three assumptions. Due to the uncertainty of observed and measured data, the obtained data inevitably is incomplete and contains noise. In this case, the choice of neural network structure (function family) is significant. If the network structure is too simple, there is a risk of under-fitting, but it will also be over-fitting if the network structure is too complex. Recent work on regularization methods can reduce the risk of over-fitting to some extent. But if the data quality is poor, it will inevitably lead to a serious decline in generalization ability.

In addition, the "black box" characteristic of deep learning is another reason for its poor generalization ability. Taking image recognition as an example, deep learning can only find recurring local fragments (patterns), whereas finding semantically meaningful parts proves to be difficult. In [33], we investigated the classification of birds' images using the deep network VGG-16. The inputting of a bird's image neuron 147# in the pool layer 5 of the network showed a strong response. The response corresponded to a local feature of the bird's head in the image. The network uses this local feature as the main basis for distinguishing birds and other objects. Obviously, it is not an invariant semantic feature of "birds". Therefore, when inputting adversarial samples of completely different semantics (e.g., figures, beer bottles, horses, etc.) but with features similar to that of a bird's head, neuron 147# also produces a strong response. The network then mistakenly identified the adversarial samples as "birds".

3 Third-generation artificial intelligence

The first generation of knowledge-driven AI uses three factors of knowledge, algorithm and computing power to construct an AI. The second generation of data-driven AI uses three different factors of data, algorithms, and computing power. Therefore, the first- and second- generations of AI simulate human intelligent behaviors in only one aspect, and each generation has its own limitations. In order to build an AI that can reflect intelligent human behaviors, we need to build robust and explainable AI theories, and develop safe, reliable, and extensible AI technology, i.e., the third-generation of AI. The key idea is to combine the knowledge-driven methods of the first generation and the data-driven methods of the second-generation, in order to construct a more powerful AI by simultaneously using the four elements of knowledge, data, algorithms, and computing power. Currently, there is a dual-space model. At present, there are two schemes of a dual-space model and a single-space model.

3.1 Dual space model

The dual-space model is shown in Figure 2. It is a brain-style model, in which a symbolic model (space) simulates rational behaviors, and the sub-symbolic model (space) simulates perceptual behaviors. These two models are seamlessly integrated into the human brain. If this integration could be achieved in computers, it would be possible for AI to achieve human-like intelligent behaviors. In order to achieve this goal, the following three problems need to be solved.

3.1.1 *Knowledge and reasoning*

Knowledge (including common sense) and reasoning are the foundations of rational behaviors. Knowledge and reasoning have witnessed significant progress in the first-generation AI, which uses symbolic models to simulate such human behaviors. However, problems in knowledge representation and reasoning methods need to be further addressed. An example of recent progress is the IBM DeepQA project [34]. It proved to be such a winning formula that the Watson dialog system, based on DeepQA, won the American TV quiz show "Jeopardy!" by overwhelming the champions Ken Jennings and Brad Rutter in February 2011. It is worth reviewing the following experience of the Watson dialogue system with respect to knowledge representation and reasoning methods. The system includes (1) a method to automatically generate structured knowledge representation from a large number of unstructured texts, (2) a method of representing uncertain knowledge based on a knowledge quality score, and (3) uncertainty reasoning based on the integration of multiple reasoning models.

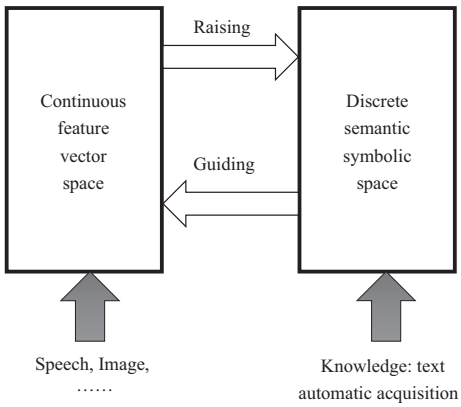

Figure 2 Dual-space mode.

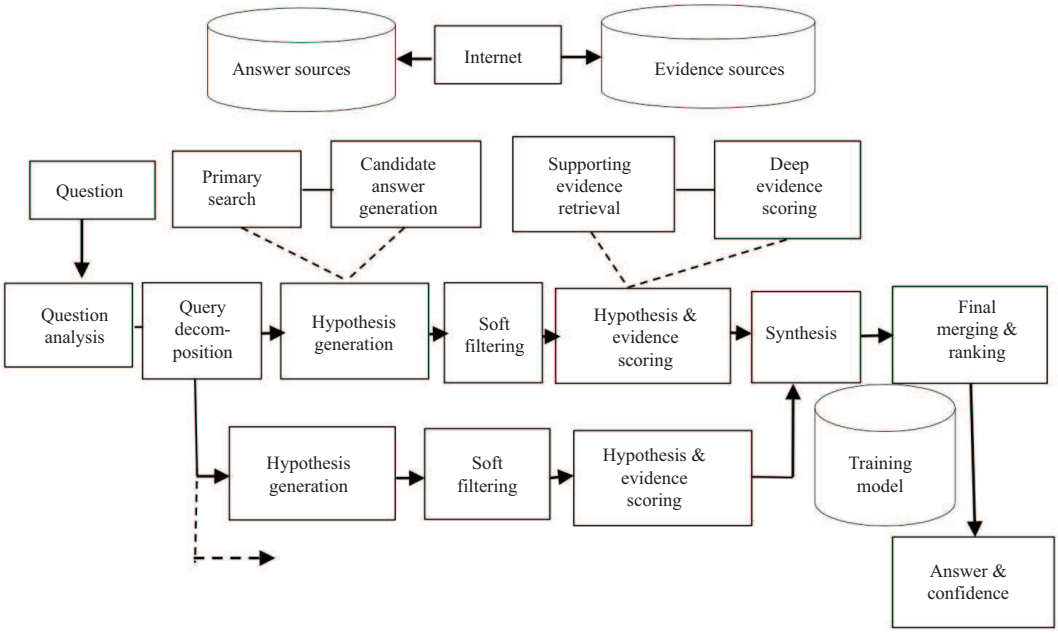

Figure 3 IBM Watson system.

Watson regards each "Question-Answer" pair as a knowledge-based inference from "Question" to "Answer". To reach human-level intelligence, a computer needs to know as much as or even more than a human champion. Watson uses a vast number of information sources, including encyclopedias, dictionaries, thesauri, newswire articles, and literary work, as well as the large online databases of DBPedia, WordNet, Yago, etc. In order to build Waston, the designers ensured that it could access at least 200 million pages of structured and unstructured documents, which are automatically transformed into a structured and manageable representation. The Watson system uses a representation called Extended Corpus as follows. First, the baseline corpus is given to identify the seed documents; then, Watson collects relevant files from the Internet according to the seed files, and mines "text nuggets" from them; finally, it scores the text nuggets and integrates them into the final "Extended Corpus" based on the score.

In addition to the automatically generated Extended Corpus, Watson's knowledge base also includes existing corpora, such as DBpedia, WordNet, Yago, as well as many other reference materials. Watson uses hundreds of reasoning mechanisms to turn "questions" into "answers" (see Figure 3). Firstly, the "question" is analyzed, classified, and decomposed. According to the decomposed results, Waston searches for the hypotheses and candidate answers from the source (corpus). After preliminary filtering, it selects about 100 candidate answers. Then Watson collects evidence from information sources and scores the candidate answers. The evaluation process also considers the reliability of data sources, and several candidate answers are synthesized based on the score. The synthesized answers are sorted according

to the degree of confidence, and then the system outputs the final answers. Watson also learned to understand "questions" through 155 live competitions with humans and 8000 experiments.

3.1.2 *Perception*

Symbolism uses a symbol system as a model of the human mind in order to achieve reasoning similar to that of humans. But, from the cognition perspective, they are fundamentally different in terms of the symbol grounding problem [35]. In the machine symbol system, the "objects" and "relations" of the objective world are represented by symbols. But the symbol itself has no meaning. We have to assign it artificial semantics, i.e., the externally imposed "parasitic semantics", which are not known to the machine. This is completely different from the "intrinsic semantics" that exists in the human brain. "Intrinsic semantics" is mainly acquired through senses (audio, visual, etc.) and the interaction between senses and movement, with very few exceptions (some atomic concepts and common sense are innate). They can translate iconic representations and categorical representations into symbolic representations. That is what deep learning is supposed to do. Unfortunately, current deep learning models are not up to the task. This is because deep learning deals with a low-level feature space, which is very different from semantic space. Only "local fragments" with no explicit semantics can be learned in this space. These fragments are not combinational and therefore cannot be used as "intrinsic semantic" representations of "objects". In other words, the current deep learning can achieve only "sensation" but not perception unless it can learn the invariant parts of an object. Taking the object "dog" for example, it needs to learn about the head, legs, tail, and other parts of a dog. The combination of these parts then forms the "intrinsic semantics" of a dog. The basic idea is that, with knowledge as the guide, sensory information is raised from a feature vector space to a semantic symbol space, as shown in Figure 2. There has been a lot of research in this area [36–39].

The preliminary progress can be illustrated by the work of our team. In [40], we describe how to improve the performance of image classification by using a triple generative adversarial network (Triple-GAN). It consists of three components: (1) a classifier C that (approximately) characterizes the conditional distribution $p_c(y|x) \approx p(y|x)$, where y is a label and x is data; (2) a class-conditional generator G that (approximately) characterizes the conditional distribution $p_g(x|y) \approx p(x|y)$; and (3) a discriminator D that distinguishes whether a pair (x, y) comes from the true distribution $p(x, y)$. All the components are parameterized as neural networks. The desired equilibrium is that the joint distributions defined by the classifier and the generator converge to the true data distribution. If the utility function is properly designed, with Triple-GAN, the generator can learn the representation of an "object" in the sample (i.e., prior knowledge) in an unsupervised (or weakly supervised) manner. The prior knowledge is used to improve the performance of the classifier. This study shows that the prior knowledge of "objects" can be learned through ANN's unsupervised learning, which is the "intrinsic semantics" of the "objects". Using this prior knowledge with "intrinsic semantics" to improve the recognition rate of the classifier will fundamentally solve the contradiction between "where" and "what" that exists in computer vision. This will facilitate learning with small data and improve its robustness and generalization.

There is an alternative way to address the problem. Let us go back to the artificial neural network used for deep learning (Figure 4). In the case of vision, an ANN is rather simple compared to human visual neural networks. It has neither feedback connections, horizontal connections, and inhibitory connections on the same layer, nor complex mechanisms such as sparse discharge (coding), memory, and attention. If we can introduce these mechanisms into ANN, it will gradually improve the perceptual ability of an AI system. Since we know very little about the working mechanism of the brain's visual neural networks, currently we can only explore it step-by-step along the path of "Brian Inspired Computing".

There is some exploratory work at present. Take one project in our group as an example. As described in [41], the mechanism of sparse discharge (coding) is applied to the calculation of each layer of ANN. There are six layers in the network, including Gabor filtering and Max pooling. In each layer, we introduce sparse regularization in the optimization procedure, which encourages the ANN to select the more representative features. When we train the network using images such as "human", "car", "elephant" and "bird", with simple backgrounds, neurons on the output layer of the neural network respond strongly to the outline of "face", "car", "elephant", and "bird", which represent these "categories". In other words, the semantic information of "whole object" is extracted, yielding the "symbolic representation" accordingly.

Nevertheless, these methods can only extract semantic information as a whole, but not the semantic

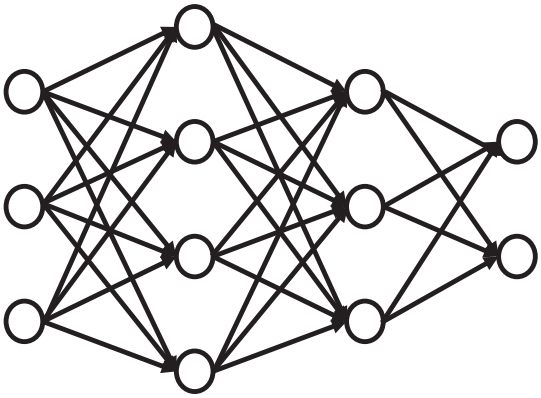

Figure 4 Artificial neural network.

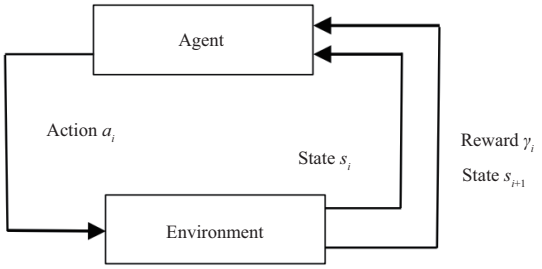

Figure 5 Reinforcement learning.

information at different levels, such as "whole", "parts", and "sub-parts". Extensive research is needed on how to extract complete semantic information, namely the intrinsic semantics, in the future.

3.1.3 *Reinforcement learning*

As mentioned above, it is possible to learn some basic knowledge (concepts) through sensory information, but sensory information is far from enough to acquire common-sense knowledge. An agent can acquire such capability only through their interaction with the environment, which results in intelligent behaviors arising from a generic objective to maximize the reward. The paradigm is recognized as reinforcement learning (RL), which is the most basic human learning behavior, and an important path towards true AI.

Reinforcement learning aims to simulate such learning behavior of human beings. It constantly interacts with the environment through the "trial-and-error" mechanism to learn effective strategies, which largely mirrors the operating mechanisms of the feedback system by which the human brain makes decisions (see Figure 5). RL has become a significant approach that may lead to breakthroughs in artificial intelligence. A recent study in [42] proposed that an agent can learn through the experience of trial-and-error by maximizing the reward, which can drive an agent to exhibit general AI abilities.

Formally, reinforcement learning is usually recognized as a random control process, i.e., the interaction process between an agent and the environment. In the learning process, the agent explores the environments by performing actions following a specific policy, and perceiving the consequences accordingly. The decision process can be modeled as a Markov decision process (MDP). The purpose is to search for the best policy for a given MDP, i.e., the policy that can maximize the reward. Many achievements have been made in video games [43, 44], chess and card games [45, 46], robot navigation and control [47, 48], human-computer interaction, and other fields. In these cases, the AI systems approach or even surpass human levels in some tasks [49, 50], which demonstrates that powerful reinforcement learning could be a possible solution to general artificial intelligence.

The success stories in reinforcement learning such as Go and video games are relatively simple. In these tasks, the environment is fully observable and the feedback is definite; the state is mainly discrete, with clear regulations; and a large amount of data can be obtained at a low cost. These all match the requirements of the current artificial intelligence algorithms. Nevertheless, for the tasks of high uncertainty, incomplete information, and insufficient data, the performance of current reinforcement learning algorithms tends to degenerate significantly. In this section, we enumerate only some of the typical challenges for the current state of reinforcement learning, and enumerate our work in these directions.

(1) Reinforcement learning in partially observable Markov decision processes (POMDPs). In real-world problems, the agent cannot reliably identify complete information about the environment; i.e., the information is incomplete. Formally, it can be modeled as a more natural model of POMDP for sequential decision-making under uncertainty. As for POMDP, the agent might need to consider all the previous observations and actions, rather than just the current state of reinforcement learning in MDP. Existing POMDP solutions optimize the policies by exploiting the inner structures in value functions using a belief-state MDP by defining beliefs over the unknown parameters. However, the complexity of the problem could increase exponentially, making the algorithms infeasible. Recently, we proposed a novel algorithm by estimating the unobservable states via a probabilistic inference [51]. Specifically, we

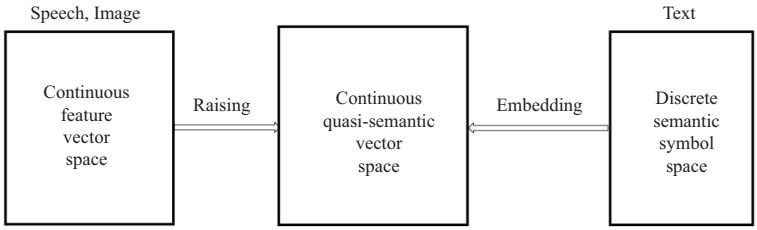

Figure 6 Single-space model.

propose sequential variational soft Q-learning networks (SVQNs) which formalizes the inference of hidden states and maximum entropy reinforcement learning under a unified graphical model. Experimental results demonstrate the effectiveness of our algorithms under the partially observable MDPs. However, it remains a challenging problem to reduce the computational complexity and speed up the convergence.

(2) The integration of domain knowledge in reinforcement learning. Similar to perceptual tasks, it is critical to integrate domain knowledge with reinforcement learning. However, without properly harnessing domain knowledge, reinforcement learning faces many problems, such as slow convergence, high sampling complexity, and low generalization. These problems could be addressed by introducing human knowledge into the learning process, which is also in line with the human mind in general decision-making procedures. One possible solution is relational reinforcement learning, which endows the agent with both knowledge representation in the form of the (probabilistic) first-order logic and reinforcement learning in complex and uncertain worlds. In [52], our group proposed to integrate human knowledge, which is represented as computable symbolic expressions, with reinforcement learning as a posterior regularization. Experiments demonstrate that we greatly improved the performance of reinforcement learning, and our method won the championship in the FPS competition in 2018 [53].

(3) The integration of game theory and reinforcement learning. At its early stage, reinforcement learning was designed for single-agent tasks in stochastic stationary environments. However, there may exist multiple agents for policy learning in numerous domains, due to either the complexity of tasks or the decentralization inherent in video games, robotics, distributed controls, etc. A basic learning framework to describe multi-agent systems involves introducing game theory into reinforcement learning. Game theory characterizes the strategic interactions between the agents. The combination of game theory and reinforcement learning provides a generic method to model the cooperation and competition between multiple agents. Examples include zero-sum/non-zero-sum games and complete/incomplete information decision-making in a multi-agent system; the latter proves to be of value, e.g., AlphaStar [44]. Our group also conducted exploratory research in this area by modeling the exploration of an RL-agent within a game theory framework [54]. We present a posterior sampling algorithm with the technique of counterfactual regret minimization (CFR), which is a novel design of interaction strategies for the RL agent with a provably theoretical guarantee.

Moreover, reinforcement learning is challenged by the gap between the simulated and real worlds, the dilemma of exploration-exploitation, better world modeling techniques for planning, etc. Compared with the success of supervised learning, the development of reinforcement learning is still at its early stage and is confronted with numerous unsolved problems.

3.2 Single-space model

In a single space model, both symbolic and sub-symbolic processing is conducted in a single vector space, as shown in Figure 6. It can make better use of the computing power of a computer, thereby increasing the processing speed, which is why it is called a computer-style model. The problem is that there exist significantly different mechanisms between deep learning and the human brain's learning, which results in major limitations in deep learning, such as interpretability and robustness. We discuss a few key issues below.

3.2.1 *The vectorization of symbols*

Knowledge is usually expressed in the form of discrete symbols in natural language. In order to build a single-space model, we must first transform the symbolic words, phrases, sentences, and chapters, as well as the knowledge graph, into vector representations. There exist various methods for word transformation,

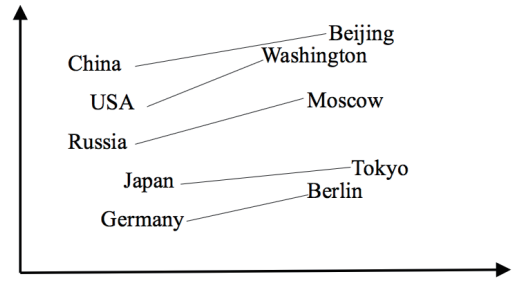

Figure 7 Word embedding graph.

also called word embedding, such as Word2Vec [55] and GloVe [56]. In this subsection, we show the skip-gram [57] strategy used in Word2Vec which can transform words from symbols to vectors as

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(c|w;\theta), \tag{3}$$

where w is a given target word, c is an optional word from its context, $p\left(c|w;\theta\right)$ is the probability of the occurrence of the word c for a given word w, D is all w-c pairs extracted from the corpus, and θ is the model parameter.

After further parameterization of (3), we obtain $p(c|w;\theta) = \frac{\mathrm{e}^{v_c v_w}}{\sum_{c'\in C} \mathrm{e}^{v_{c'} v_w}}$, where v_c, $v_w \in \mathbb{R}^d$ are vector representations of the words c and w, respectively; and C is all available text; θ is the parameter with a proper dimension. By optimizing (3), we can obtain the vector representation for each word of v_w.

These word vectors have the desirable property that "semantically similar words have very similar word vectors" which can be guaranteed for the following assumption. The higher the co-occurrence frequency of two words in the context, the more likely they are to be semantically similar or semantically related. These characteristics of the embedded word vectors indicate that the word vectors carry some semantics, which is so-called quasi-semantic space, as is illustrated in Figure 7. In general, Eq. (3) is intractable to solve since it needs to enumerate all the possibilities which can be approximated using a deep neural network. Similar embedding techniques can be used to transform "phrases", "sentences" and "texts" or knowledge graphs into quasi-semantic vector spaces [58].

Knowledge representation in vector form has the useful property that it can be manipulated as data with various mathematical tools. Therefore, it has been widely used in text processing with remarkable results, such as neural machine translation [59, 60].

The basic task of neural machine translation is that, with a given source sentence (e.g., Chinese) $\boldsymbol{x} = x_1, x_2, \ldots, x_I$, it will search for the target sentence (e.g., English) $\boldsymbol{y} = y_1, y_2, \ldots, y_J$, which can be formulated as the likelihood of a sequence of words in the target sentence as

$$p(\boldsymbol{y} \mid \boldsymbol{x};\boldsymbol{\theta}) = \prod_{j=1}^{J} p\left(y_j \mid \boldsymbol{x}, \boldsymbol{y}_{<j};\boldsymbol{\theta}\right), \tag{4}$$

where $\boldsymbol{\theta}$ is a set of model parameters, $\boldsymbol{y}_{<j} = y_1, \ldots, y_{j-1}$ is the partially translated target sentence. In order to construct the neural translation models, the training set is a collection of "source-target sentence" pairs $\{\langle \boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)} \rangle\}_{n=1}^{N}$, and the objective is to maximize the log likelihood as

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \arg\max_{\theta} \left\{ \sum_{n=1}^{N} \log p\left(\boldsymbol{y}^{(n)} \mid \boldsymbol{x}^{(n)};\boldsymbol{\theta}\right) \right\}, \tag{5}$$

where $\boldsymbol{\theta}$ is a set of model parameters. The target sentence can be obtained via optimizing the equation. In general, neural machine translation can achieve a lower error rate than the traditional methods. However, it shares the common drawbacks of deep learning methods, such as not being explainable to humans, unavoidable fatal errors, and poor robustness. It is expected that future research will overcome these limitations by integrating prior knowledge, posterior regularization, etc.

3.2.2 *The improvement of deep learning*

Recently, much work has been done to overcome the limitations in the current deep learning techniques in terms of explainability and robustness. In this subsection, we will elaborate on representative work by our group.

Explainability. The black-box nature of the recent deep learning models challenges their use in mission-critical scenarios such as healthcare and self-driving vehicles. Therefore, explainable artificial intelligence (XAI) has attracted significant attention from numerous researchers. XAI aims to generate high-quality interpretable or human-understandable explanations of AI decisions. The key challenge for XAI is to bridge the gap between the feature space and the semantic space, with the latter being interpretable for human beings [61]. The core idea of machine-learning technologies is to map the raw data to a suitable internal representation or feature vector to realize purposes such as classification or regression, no matter whether they are working with early algorithms based on hand-crafted features or the current deep learning algorithms based on feature learning. However, feature vectors generally are not interpretable to human beings. In contrast, a human generally makes decisions in semantic space using their background knowledge, which has significant differences with data in a feature space in terms of the inherent structures. The purpose of explainable artificial intelligence is to bridge the gap between the feature space and semantic space.

At present, the motivation and techniques for explainable models are diverse and sometimes discordant. The previous work broadly falls into two categories. The first category comprises post-hoc interpretable techniques which focus on explaining predictions without elucidating the inner mechanisms. Common approaches include natural language explanations, visualizations of learned representations, and explanations by using examples.

The second type, explainable artificial intelligence, attempts to develop transparent models with the aim of investigating how the model works at different levels, i.e., the entire model, individual components (e.g., parameters), and the training algorithm. The key challenge for the transparent models is to investigate the processes of how humans make decisions and enhance the unconscious features to a more structured symbolic concept representation. Both types of methods are in the process of rapid development, and play important roles in the study of interpretability.

Visualization is a common approach to generating the post-hoc interpretation for an artificial intelligence algorithm. Now that deep learning is recognized as a kind of "black-box" method, since the inner mechanism is opaque and "unexplainable", we can use the tool of visualization to open the "black box" such that everything will be clear. In order to provide users with a tool to better understand, diagnose, and refine deep CNNs, Liu et al. [62] proposed to formulate a deep CNN as a directed acyclic graph (DAG), and presented a novel hybrid visualization that integrates a DAG with rectangle packing, matrix visualization, and a biclustering-based edge bundling method. We further have developed a visual analytics system, named CNNVis, which is one of the first tools for the visual analysis of deep learning models. CNNVis has attracted widespread attention from both industry and academia. More recently, we also proposed to analyze training dynamics, which provide a tool to understand the failure cases and to assist in debugging in the training process, with an example of a DGMTracker that can assist experts in understanding the training dynamic of the deep generative models.

The parameter redundancy in deep neural networks poses the greatest hindrance to interpretability by humans. A popular way to tackle parameter redundancy, and thereby better explain a model, is to extract the important sub-components by using machine learning techniques. In order to extract the structured subnetworks, we borrow ideas from network pruning techniques, and obtain a partial structure of the original full model with comparable predictive performance [63]. Specifically, the critical subnetworks can be recognized as a group of important channels such that the performance would deteriorate severely if the subnetworks were suppressed to zero. By formulating the problem under a knowledge distillation framework [64], we associate a control gate with the layer's output channels, which reflects the importance of each channel in the network. After solving the problem, we find the semantic concepts contained in the critical subnetwork representations; i.e., the subnetworks reflect consistent input patterns in the intraclass samples, which can help identify outlier examples in the dataset and in the more accurate and reasonable explanatory regions.

The aforementioned methods focus on the post-hoc interpretation of the models which seek an explanation of the prediction but may be farfetched in general. It remains a question whether the explanation conforms to the internal mechanism of the neural networks. Due to the inherent difference between the

latent feature space for deep learning and the human interpretable semantic space, it is imperative to bridge the two spaces in order to derive a more transparent AI model. To address this issue, our group proposes to improve the interpretability of the DNNs by leveraging the rich semantic information described by human knowledge [65]. By extracting a set of semantically meaningful topics from the human descriptions, we propose to integrate them into the model with an interpretive regularization, yielding a model with more interpretable feature vectors. The result shows that human-interpretable knowledge can guide the learning procedure of a deep neural network, yielding an interpretable representation.

Robustness. The growing sophistication of AI techniques has dramatically empowered AI systems, but, as a byproduct, it also brings about potential new vulnerabilities. Adversarial attacks on an AI system can generate imperceptible perturbations which can deceive an AI model and significantly alter its response. In this case, it is essential to identify the vulnerabilities, and further improve the defensibility of an AI system functionally and substantially [66].

To deal with the threat of adversarial examples, extensive research has been conducted on building robust models to defend against adversarial attacks. Such work can be roughly classified into three categories.

(1) Input transformation. Several defenses adopt transformation on the inputs before feeding them to the classifier [67], or project adversarial examples onto the data distribution [68]. Among these methods, the denoiser-based method has attracted wide attention due to its plug-and-play nature that does not change the structures or properties of the models. However, a denoiser with good performance on general Gaussian noise may fail to remove adversarial perturbations. Also, the small residual perturbation could be amplified to a large magnitude in the top layers, which is the so-called "error amplification effect" and leads to mistakes in prediction. In order to solve this problem, our group proposed a novel method for a high-level representation guided denoiser (HGD) [69], which tries to minimize the difference between the outputs of the target model induced by the original and the adversarial examples at the top levels. Experiments demonstrate that HGD consistently produces more robust feature representation by suppressing the progressive amplification of adversarial noise across the layers. In the NeurIPS2017 competition on adversarial machine learning, our HGD solution won first place and outperformed the alternative models by a large margin.

(2) Robust training. Robust training is a basic strategy for defending against adversarial attacks by augmenting training data with a batch of adversarial examples. The problem can be formulated as a min-max optimization problem, in which the goal of the inner maximization is to find the adversarial examples while the outer minimization aims to train a robust classifier [70]. However, most of the existing AT methods solve the inner maximization problem based on a specific attack algorithm, yielding poor generalization for other unseen attacks. Recent work has demonstrated that even methods with state-of-the-art robustness against commonly used attacks can still be defeated by other attacks [71]. To address this issue, we propose a novel framework of adversarial distributional training (ADT) which learns an adversarial distribution to characterize the potential adversarial examples around a natural example [72]. Through theoretical analysis and empirical evaluation, we demonstrated that our ADT training algorithm can lead to a more robust AI model that can resist adversarial examples, especially the ones generated by unseen attacks.

(3) Model enhancement. Another category of methods is to train a more robust deep learning model by modifying the network structure, the activation function, or the loss function of the models. Among these, the model ensemble is an effective defense strategy in practice [73, 74]. However, the general ensemble model may fail to defend against adversarial attacks because the adversarial examples have strong transferability among different models. To address this issue, our group proposed to introduce an adaptive diversity promoting (ADP) regularization to boost the diversity among the different models [75]. In particular, we introduced a logarithm of ensemble diversity (LED) term into the ADP, which encourages the non-maximal predictions of each member to be mutually orthogonal, while at the same time preserves the maximal prediction consistent with the ground truth. Extensive results on various datasets verify that our method can improve adversarial robustness significantly while maintaining state-of-the-art accuracy on normal examples.

However, most of these methods are heuristics that have been subsequently shown to fail to defend against the more powerful adversaries. One of the major reasons is that most of the current deep learning uses function approximation algorithms, which in general are unequipped to understand the inner mechanism of human-like intelligence [76]. Therefore, it is imperative to pursue a variety of techniques that can explain their rationale, and characterize the strength and weakness. An alternative path is to

integrate data and knowledge that can help overcome the brittle nature of current AI models, which may fail when presented with cases not included in the training phase.

3.2.3 *Bayesian deep learning*

As shown in Figure 6, visual and auditory signals are processed in the feature space, where the features are often insufficient in capturing semantics. One promising way to increase the amount of semantics in feature representation is to incorporate knowledge into deep learning. Below, we will take Bayesian deep learning as an example to illustrate the idea.

As stated before, deep neural networks are insufficient in characterizing the uncertainty in observing empirical data. Such uncertainty, as well as the unknown physical process underlying the data, makes it difficult to judge the correctness of the outputs of deep learning methods. Meanwhile, because the available data is basically finite and often fixed, the uncertainty of the model increases as the model size quickly grows (e.g., networks with billions or trillions of parameters). Therefore, there exist many models that perform well on training data, but much less satisfactorily on testing data. Bayesian machine learning (or Bayesian learning in short) provides an elegant framework to consider prior knowledge as well as the uncertainty of both the data and the model. It can gradually refine the understanding of the data (or task) when more evidence arrives, which is known as incremental learning. Such a paradigm can sufficiently leverage knowledge during learning, and it can not only provide an estimate of the prediction confidence, but also improve the efficiency and accuracy of learning.

The formal definition of Bayesian learning is as follows. Let D denote all the data. Given the observed data set $\boldsymbol{d} \subset D$, it calculates the posterior probability distribution of each hypothesis via Bayes' rule in (6) [12, 77] as

$$p(h_i|\boldsymbol{d}) = \alpha p(\boldsymbol{d}|h_i)p(h_i), \tag{6}$$

where $p(h_i)$ is the prior distribution (before seeing the data), and $p(\boldsymbol{d}|h_i)$ is the likelihood of $\boldsymbol{d}$ given the hypothesis h_i. With the posterior distribution, we can make a prediction on unknown variable X as

$$p(X|\boldsymbol{d}) = \sum_i p(X|\boldsymbol{d}; h_i)p(h_i|\boldsymbol{d}) = \sum_i p(X|h_i)p(h_i|\boldsymbol{d}), \tag{7}$$

where the hypothesis space is assumed to be finite for simplicity; for continuous hypothesis spaces, the summation will be integral.

One challenge of Bayesian learning involves computation; when the hypothesis space is huge, Eq. (7) can be infeasible, which hinders practical application. A common solution is to use approximation inference algorithms, which fall into two main categories: variational inference and Monte Carlo methods [78]. In addition, there are some common approaches to simplifying the calculation: (1) instead of marginalizing over all hypotheses when predicting X, use only a single hypothesis h_i that maximizes $p(h_i|\boldsymbol{d})$, known as maximum a posteriori (MAP) estimate; (2) by assuming a uniform prior $p(h_i)$, the MAP estimator can be further simplified as finding a hypothesis that maximizes $p(h_i|\boldsymbol{d})$, known as the maximum likelihood estimate (MLE); (3) if not all variables are observable, i.e., some variables are hidden, it is typical to adopt the expectation maximization (EM) algorithm to solve the MLE problem as in (8). The EM algorithm alternates between two steps: the E-step uses the observed data X and current estimate of model parameters $\theta^{(i)}$ to calculate the posterior of hidden variable $p(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{x}; \theta^{(i)})$, and the M-step updates the model parameters $\theta^{(i+1)}$ according to the inferred z and x as

$$\theta^{(i+1)} = \arg\max_{\theta} \sum_{\boldsymbol{z}} p(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{x}; \theta^{(i)}) L(\boldsymbol{x}, \boldsymbol{Z} = \boldsymbol{z}|\theta). \tag{8}$$

Bayes' rule in (7) is a procedure that implies the posterior distribution from a given prior and data likelihood. In order to incorporate knowledge more flexibly, we propose a framework of regularized Bayesian inference (RegBayes) [79]. RegBayes is built on a variational formulation of Bayes' rule [80], and we introduce the notion of posterior regularization to directly constrain the property of the targeted post-data posterior distribution. Under a unified formulation, RegBayes can very flexibly consider domain knowledge (e.g., the knowledge in first-order logic [81]) or the objective of the task (e.g., max-margin loss for classification [82]).

Bayesian deep learning integrates the fundamental principle of Bayesian inference and representation learning via deep neural networks as illustrated in Figure 8. There are two types of integration:

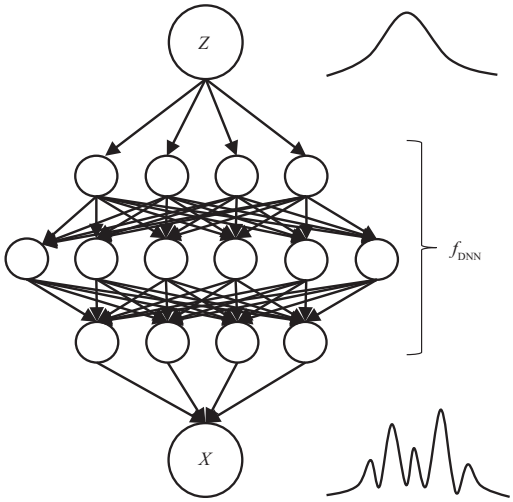

Figure 8 A general framework of deep Bayesian models with DNN as function approximator.

(1) perform Bayesian inference over deep neural networks (e.g., Bayesian neural networks and Gaussian processes), with the main objectives of calculating uncertainty and avoiding overfitting; and (2) use deep neural networks as a nonlinear function approximator to improve the capacity of Bayesian models; examples include generative adversarial networks (GANs), variational auto-encoder (VAE), and invertible network-based flow models. The first type of integration was pioneered by Hopfield, Hintion, and their students in 1990s [83, 84]. Back then, due to the limitation on computing power and training data, networks of medium size typically had a high risk of overfitting. Therefore, one major motivation was to perform Bayesian inference to protect neural networks from overfitting and select proper models. In the era of deep learning, as the network depth increases, Bayesian methods have attracted further attention, which led to major progress on efficient algorithms for Bayesian deep networks. One major challenge involves the over-parameterization of neural networks in Bayesian inference. To address this challenge, we have developed various methods, including implicit variational inference [85,86] and functional variational inference [87].

As for the second type of integration, we know that a simple random variable z can be transformed into a more complex random variable $y = f(z)$ through a function f. When f is a bijective function, the distribution of y will be $p(x) = p(z)|\frac{\mathrm{d}z}{\mathrm{d}y}|$. We have used this trick when drawing samples from Gaussian and many other distributions, where the function f often has some simple analytical form. However, when we deal with complex data, the function f is unknown and may be very complicated. Therefore, we desire to directly learn f from empirical data. By leveraging the expressiveness of deep networks, we can parameterize f as a deep network, and learn an optimal transforming function f_θ under some task-related objective. As illustrated in Figure 8, this idea has been shown effective in practice, with many representative examples such as VAE, GAN, and flow-based models. Even when trained under a fully unsupervised setting, such models can generate high-quality natural images or faces.

One key difference among these models is the transforming function of defining y. In VAE, $y = f_\theta(z)+\epsilon$, where ϵ is a noise variable (e.g., white Gaussian noise), while in GAN and flow-based models, there is no explicit noise variable. This difference leads to different strategies to estimate parameters. VAE and flow-based models adopt MLE, while GAN formulates the learning as a min-max game via adversarial training. Similar to the first-type of integration, though such methods are powerful, they also raise various challenges on inference and learning. For instance, the training of GAN is often unstable, facing gradient vanishing or explosion issues. We investigated this issue and presented a stabilizing approach to train GAN by drawing inspiration from control theory [88]. On the other hand, flow-based models adopt an invertible transforming function, which can lead to a "dimensionality bottleneck" [89]. One possible solution is to augment the dimension by performing variational inference (e.g., [89]).

It is clear from the above illustration that Bayesian deep learning provides powerful modeling language to conjoin the advantages of probabilistic inference and representation learning via deep networks, while the key challenges are on inference and learning algorithms. Recently, much progress has been made on algorithms (see above). Meanwhile, various probabilistic programming libraries have been designed to support the development and deployment of Bayesian deep learning models. One early example is the

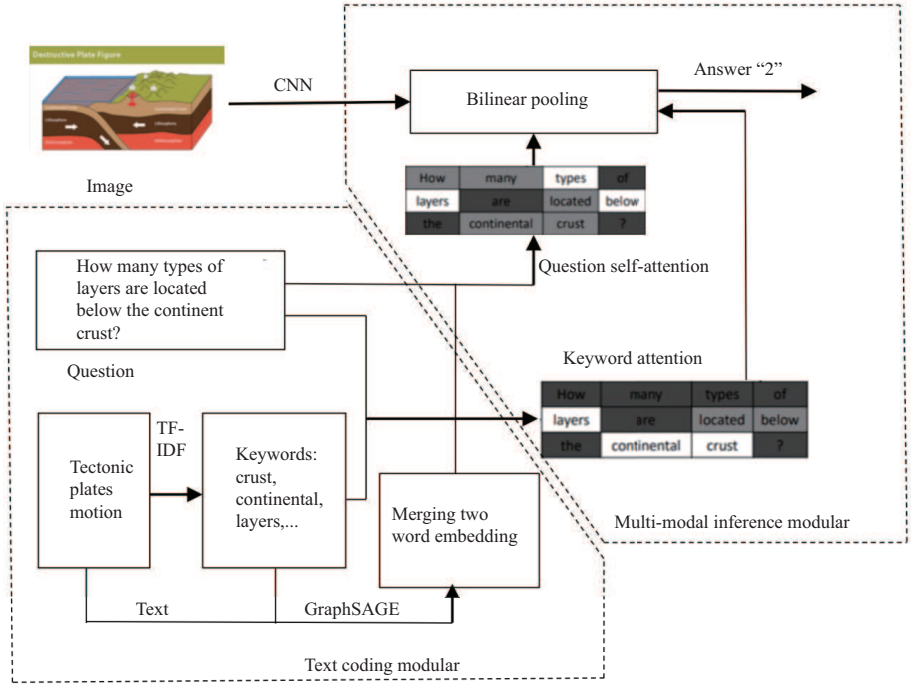

Figure 9 (Color online) The architecture of image-text Question-Answer system.

"ZhuSuan" library [90][1)]. Finally, in terms of application, Bayesian deep learning has been used for various tasks with promising results, including time-series forecasting, semi-supervised learning, unsupervised learning, few-shot learning, and continual learning.

3.2.4 *Calculation in a single space*

As shown in Figure 6, it is difficult for us to calculate the embedded vector from text and the feature vector from visual (image) or/and auditory in a single vector space. This is because the words of the text lose a lot of semantics when they are converted from symbols to vectors after embedding, and the features extracted from visual (or/and auditory) perception generally belong to low-level features with no semantics.

We take an "essay-level image-text question answering" task from our team [91] as an example to introduce a preliminary attempt in this field. More work can be referred to in [92, 93]. Both images and text in an "essay-level image-text question answering" task must be processed in a single vector space. As shown in Figure 9, the task is to answer the question "How many types of layers are located below the continental crust?" based on the given image. In addition to the textual representation of the question, there is also an essay "Tectonic Plates Motion" related to the image.

First, we use the word embedding based on skip-gram to transform words represented by discrete symbols in "the problem" and "essay" into vectors. After the image is processed by the ResNet, we use the high-layer feature vector to represent the image [57]. Then, the word vectors in "question" and "essay" are fused with the feature vectors of "image" to predict the "answer". For better integration, the "key words" in "problem" and "essay" are first identified through the attention mechanism since these keywords can better reflect the theme (semantics) of the "problem". Based on the keywords, the spatial attention mechanism is used to find the features of the key areas in the image, since these key areas are more consistent with the theme expressed by the keywords. Moreover, we use multimodal bilinear pooling as the integration method. The current performance of machine "image-text QA" is not comparable to that of human beings. In the case of multiple-choice questions, for example, the level achieved so far is only slightly better than random guesses.

1) https://zhusuan.readthedocs.io/.

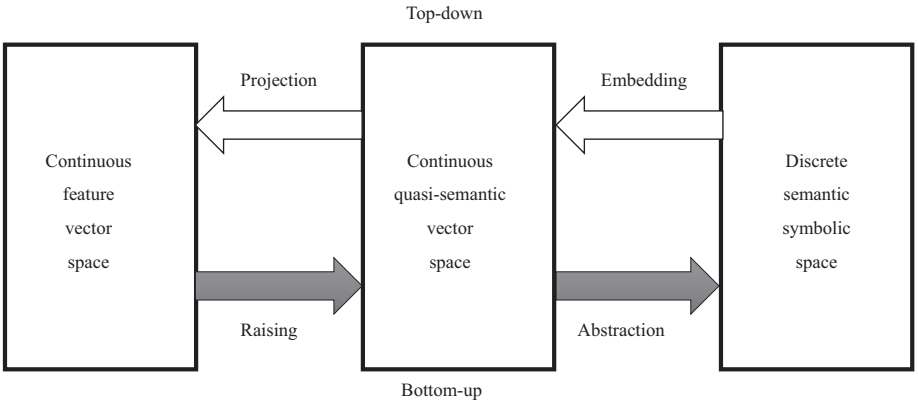

Figure 10 Triple-space integration model.

4 Conclusion

In order to realize third-generation AI, we propose a triple-space model which is an integration of the dual-space and single space models. Specifically, the dual-space model uses a brain-like working mechanism. If it can be realized, the machine will behave in an interpretable and robust manner that will be close to that of the brain. For the dual-space model, the visual/auditory sensory signals (feature vectors) are upgraded to concepts (discrete symbols). Then concepts (symbols) are generated by perception. Symbols have their grounding, and symbols and symbolic reasoning have their intrinsic semantics. The problems of interpretability and robustness can be solved fundamentally. Nevertheless, the symbol grounding problem is a difficult task that needs to be explored.

The dual-space model imitates the working mechanism of the brain, but, because we know very little about the working mechanism of the brain, there are some uncertainties in this path. One question is whether the "intrinsic semantics" obtained via reinforcement learning is the same as that acquired by human perception. Can machines also be conscious? Despite these difficulties, we believe that, as long as the machine takes a step in this direction, it will be closer to real AI.

In a single space model, both symbolic and sub-symbolic processing is conducted in a single vector space. In addition to the need to raise (visual, auditory, etc.) vector representation from feature space into semantic space, it is also necessary to ensure that the semantics will not be lost when the word (or sentence, chapter, etc.) is embedded. In general, a single-space model is based on deep learning, which can make full use of computing power, and in some aspects can demonstrate better performance than humans. However, there still exist great uncertainties about the potential for improvements to deep learning, due to the fundamental limitations in terms of its interpretability and robustness. However, we believe that every step of improvement in deep learning will promote the development of AI.

Considering the above uncertainties, the best strategy is to proceed along these two routes at the same time, through the integration of the three spaces: continuous feature vector space, discrete semantic symbolic space, and continuous quasi-semantic vector space as is illustrated in Figure 10. In this way, we can maximize the use of brain-like working mechanisms, and make full use of the computing power of the computer, which is expected to result in a more powerful AI, i.e., the third-generation AI.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61620106010).

References

1 Simon H A. Models of Man. New York: Wiley & Sons, 1957
2 Newell A, Simon H A. Computer science as empirical inquiry: symbols and search. Commun ACM, 1976, 19: 113–126
3 Newell A. Physical Symbol Systems. Cogn Sci, 1980, 4: 135–183
4 Fodor J A. Methodological solipsism considered as a research strategy in cognitive psychology. Behav Brain Sci, 1980, 3: 63–73
5 McCarthy J, Minsky M L, Rochester N, et al. A proposal for the Dartmouth summer research project on artificial intelligence. 1955, 27: 4
6 Lindsay, Robert K, Bruce G. Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. New York: McGraw-Hill Book Company, 1980

7 Buchanan B G, Shortliffe E H. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Boston: Addison Wesley, 1984

8 Muggleton S, de Raedt L. Inductive logic programming: theory and methods. J Logic Programm, 1994, 19-20: 629–679

9 Riguzzi F, Bellodi E, Zese R. A history of probabilistic inductive logic programming. Front Robot AI, 2014, 1: 6

10 Yang Q, Zhang Y, Dai W Y, et al. Transfer Learning. Cambridge: Cambridge University Press, 2020

11 Ehrlinger L, Wolfram W. Towards a definition of knowledge graphs. In: Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems (SEMANTiCS2016) and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16), Leipzig, 2016

12 Russel S J, Norvig P. Artificial Intelligence: A Modern Approach. 2nd ed. New York: Pearson Education Inc., 2003

13 Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev, 1958, 65: 6

14 Rosenblatt F. Principles of Neurodynamics. Arch Gen Psychiatry, 1962 7: 218–219

15 McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys, 1943, 5: 115–133

16 Hebb D O. The Organization of Behavior: A Neuropsychological Theory. London: Psychology Press, 1949

17 Minsky M, Papert S A. Perceptrons: An Introduction to Computational Geometry. Cambridge: MIT Press, 1969

18 Cauchy A. Methode generale pour la resolution des systemes d'equations simultanees. Comp Rend Acad Sci, Paris, 1847, 25: 536–538

19 Nesterov Y E. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Soviet Math Doklady, 1983, 27: 372–376

20 Linnainmaa S. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors (in Finnish). Dissertation for Master's Degree. Helsinki: University of Helsinki, 1970

21 Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. Nature, 1986, 323: 533–536

22 Janocha K, Czarnecki W M. On loss functions for deep neural networks in classification. Schedae Inform, 2016, 25: 49–59

23 Wan L, Zeiler M, Zhang A X, et al. Regularization of neural networks using DropConnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML), Atlanta, 2013

24 Fukushima K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybernetics, 1980, 36: 193–202

25 Derevyanko G, Grudinin S, Bengio Y, et al. Deep convolutional networks for quality assessment of protein folds. Bioinformatics, 2018, 34: 4046–4053

26 Lipton J C, Berkowitz J. A critical review of recurrent neural networks for sequence learning. 2015. ArXiv:1506.00019v4

27 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9: 1735–1780

28 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Comput, 2006, 18: 1527–1554

29 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444

30 Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search. In: Proceedings of International Conference on Computers and Games, Berlin, 2006

31 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2015

32 Dong Y P, Liao F Z, Pang T Y, et al. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018

33 Dong Y, Su H, Zhu J, et al. Towards interpretable deep neural networks by leveraging adversarial examples. In: Proceedings of IJCAI Workshop on AISC, Sydney, 2019

34 Ferrucci D, Brown E, Chu-Carroll J, et al. Building Watson: an overview of the DeepQA project. In: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI), Atlanta, 2010

35 Harnad S. The symbol grounding problem. Phys D-Nonlin Phenom, 1990, 42: 335–346

36 Chen X, Duan Y, Houthooft R, et al. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), Barcelona, 2016. 2172–2180

37 Liu Y, Wei F Y, Shao J, et al. Exploring disentangled feature representation beyond face identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Salt Lake City, 2018. 2080–2089

38 Higgins I, Matthey L, Pal A, et al. Beta-VAE: learning basic visual concepts with a constrained variational framework. In: Proceedings of the 32nd International Conference on Logic Programming (ICLP), New York City, 2016

39 Siddharth N, Paige B, Desmaison A, et al. Inducing interpretable representations with variational autoencoders. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), Barcelona, 2016

40 Li C, Xu K, Zhu J, et al. Triple generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), Long Beach, 2017

41 Hu X L, Zhang J W, Li J M, et al. Sparsity regularized HMAX for visual recognition. Plos One, 2014, 9: e81813

42 Silver D, Singh S, Precup D, et al. Reward is enough. Artif Intell, 2021, 299: 103535

43 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533

44 Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 2019, 575: 350–354

45 Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529: 484–489

46 Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. Nature, 2017, 550: 354–359

47 Duan Y, Chen X, Houthooft R, et al. Benchmarking deep reinforcement learning for continuous control. In: Proceedings of International Conference on Machine Learning (ICML), New York, 2016

48 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017. ArXiv:1707.06347

49 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge: MIT Press, 2015

50 François-Lavet V, Henderson P, Islam R, et al. An introduction to deep reinforcement learning. FNT Machine Learn, 2018, 11: 219–354

51 Huang S Y, Su H, Zhu J, et al. SVQN: sequential variational soft Q-learning networks. In: Proceedings of International Conference on Learning Representations (ICLR), 2020

52 Huang S Y, Su H, Zhu J, et al. Combo-action: training agent for FPS game with auxiliary tasks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI), Honolulu, 2019

53 Song S H, Weng J Y, Su H, et al. Playing FPS games with environment-aware hierarchical reinforcement learning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), Macau, 2019

54 Zhou Y C, Li J L, Zhu J. Posterior sampling for multi-agent reinforcement learning: solving extensive games with imperfect information. In: Proceedings of International Conference on Learning Representations (ICLR), Addis Ababa, 2020

55 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. In: Proceedings of International Conference on Learning Representations (ICLR), Scottsdale, 2013

56 Pennington J, Socher R, Manning C D. Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, 2014

57 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), Lake Tahoe, 2013

58 Kusner M J, Sun Y, Kolkin N I, et al. From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, 2015

59 Zhang J, Liu Y, Luan H B, et al. Prior knowledge integration for neural machine translation using posterior regularization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017

60 Ding Y, Liu Y, Luan H B, et al, Visualizing and understanding neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017

61 Zhang B. Artificial intelligence in the post-deep learning era (in Chinese). CAAI Trans Intell Technol, 2017, 7: 3–5

62 Liu M, Shi J, Li Z, et al. Towards better analysis of deep convolutional neural networks. In: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2016

63 Wang Y L, Su H, Hu X L. Interpret neural networks by identifying critical data routing paths. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018

64 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), Montreal, 2014

65 Dong Y P, Su H, Zhu J, et al. Improving interpretability of deep neural networks with semantic information. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017

66 Dong Y P, Fu Q-A, Yang X, et al. Benchmarking adversarial robustness on image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 2020

67 Dziugaite G K, Ghahramani Z, Roy D M. A study of the effect of jpg compression on adversarial images. 2016. ArXiv:1608.00853

68 Samangouei P, Kabkab M, Chellappa R. Defense-GAN: protecting classifiers against adversarial attacks using generative models. In: Proceedings of International Conference on Learning Representations (ICLR), 2018

69 Liao F Z, Liang M, Dong Y P, et al. Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018

70 Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks. In: Proceedings of International Conference on Learning Representations (ICLR), 2018

71 Tramer F, Carlini N, Brendel W, et al. On adaptive attacks to adversarial example defenses. 2020. ArXiv:2002.08347

72 Dong Y P, Deng Z J, Pang T Y, et al. Adversarial distributional training for robust deep learning. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), 2020

73 Kurakin A, Goodfellow I, Bengio S, et al. Adversarial attacks and defences competition. 2018. ArXiv:1804.00097

74 Liu X Q, Cheng M H, Zhang H, et al. Towards robust neural networks via random selfensemble. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018

75 Pang T Y, Xu K, Du C, et al. Improving adversarial robustness via promoting ensemble diversity. In: Proceedings of International Conference on Machine Learning (ICML), Long Beach, 2019

76 Castelvecchi D. Can we open the black box of AI? Nature, 2016, 538: 20–23

77 Ghahramani Z. Probabilistic machine learning and artificial intelligence. Nature, 2015, 521: 452–459

78 Zhu J, Chen J, Hu W, et al. Big learning with Bayesian methods. Natl Sci Rev, 2017, 4: 627–651

79 Zhu J, Chen N, Xing E P. Bayesian inference with posterior regularization and applications to infinite latent SVMs. J Mach Learn Res, 2014, 15: 1799–1847

80 Williams P M. Bayesian conditionalisation and the principle of minimum information. Br J Philosophy Sci, 1980, 31: 131–144

81 Mei S, Zhu J, Zhu X. Robust RegBayes: selectively incorporating first-order logic domain knowledge into Bayesian models. In: Proceedings of International Conference on Machine Learning (ICML), Beijing, 2014

82 Zhu J, Ahmed A, Xing E P. MedLDA: maximum margin supervised topic models. J Mach Learn Res, 2012, 13: 2237–2278

83 MacKay D J C. Bayesian methods for adaptive models. Dissertation for Ph.D. Degree. Pasadena: California Institute of Technology, 1992

84 Neal R M. Bayesian learning for neural networks. Dissertation for Ph.D. Degree. Toronto: University of Toronto, 1995

85 Shi J, Sun S, Zhu J. A spectral approach to gradient estimation for implicit distributions. In: Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, 2018

86 Zhou Y, Shi J, Zhu J. Nonparametric score estimators. In: Proceedings of International Conference on Machine Learning (ICML), Vienna, 2020

87 Wang Z, Ren T, Zhu J, et al. Function space particle optimization for Bayesian neural networks. In: Proceedings of International Conference on Learning Representations (ICLR), New Orleans, 2019

88 Xu K, Li C, Zhu J, et al. Understanding and stabilizing GANs' training dynamics using control theory. In: Proceedings of International Conference on Machine Learning (ICML), Vienna, 2020

89 Chen J, Lu C, Chenli B, et al. VFlow: more expressive generative flows with variational data augmentation. In: Proceedings of International Conference on Machine Learning (ICML), Vienna, 2020

90 Shi J, Chen J, Zhu J, et al. ZhuSuan: a library for Bayesian deep learning. 2017. ArXiv:1709.05870

91 Li J Z, Su H, Zhu J, et al. Essay-anchor attentive multi-modal bilinear pooling for textbook question answering. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), San Diego, 2018

92 Ren M, Kiros R, Zemel R S. Exploring models and data for image question answering. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), 2015

93 Zhu Y, Groth O, Bernstein M S, et al. Visual7W: grounded question answering in images. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016