Neuro Symbolic AI

Taxonomy & Overview

Antonis Ganios 13/10/2023

Introduction

NN & Symbolic Systems

- Perception, represented by connectionism or neural systems, and cognition, represented by symbolism or symbolic systems, are two fundamental paradigms in the field of artificial intelligence (AI), each having prevailed for several decades.
- Symbolic methods have several benefits:
  - they require only a few input samples
  - use powerful declarative languages for knowledge representation,
  - clear internal functionality.
- Unable to handle well noisy and (or) unstructured data.

Introduction

NN & Symbolic Systems

- Neural networks serves as the architecture behind the majority of recent successful AI systems.
- Learning happens as weight modification, in a data-driven manner
- The network weights are adjusted in the direction that minimises the cumulative error from all the training samples.
- In some cases like question answering medical diagnosis and autonomous driving relying solely on perception can present limitations or yield unsatisfactory outcomes.
- Another crucial consideration is the compatibility of purely perception-based models with the principles of explainable AI
- Neural networks, being black-box systems, are unable to provide explicit calculation processes.
- An increasing number of researchers have directed their attention towards the fusion of neural systems and symbolic systems aiming to achieve the third wave of AI.
- By unifying these two system types within a comprehensive framework, we create **neural-symbolic learning** systems.

Introduction

Summarize properties for the symbolic systems and neural systems separately.

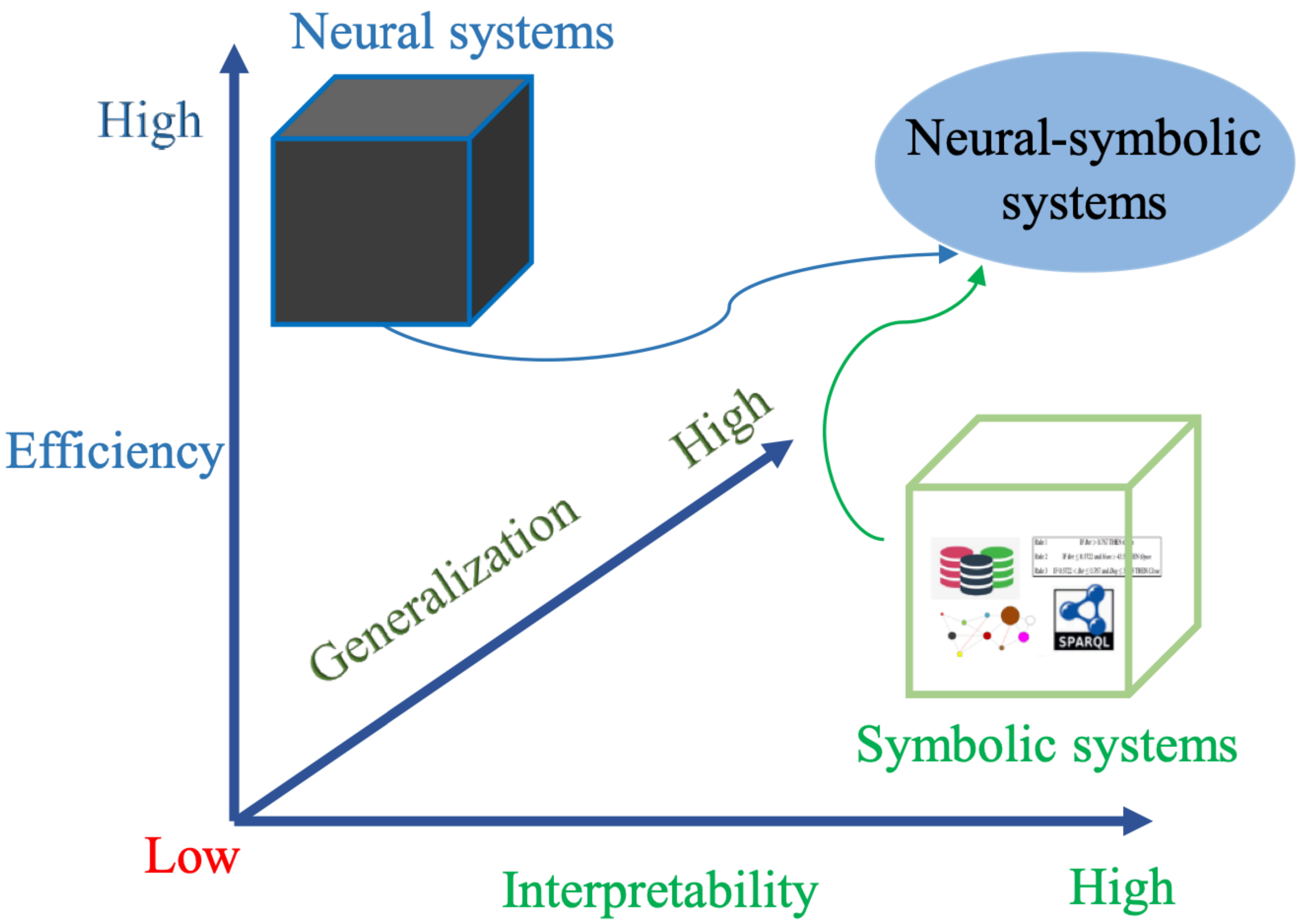
Systems	Processing Methods	Knowledge representation	Primary algorithms	Advantages	Disadvantages
Symbolic systems	Deductive reasoning	Logical representation	Logical deduction	Strong generalization ability Good interpretability Knowledge-driven	Weak at handling unstructured data Weak robustness Slow reasoning
Neural systems (Sub-symbolic systems)	Inductive learning	Distributed representation	BP algorithms	Strong at handling unstructured data Strong robustness Fast learning	Weak generalizability (adaptability) Lack of interpretability Data-driven

Introduction

Unified framework

- By combining the two architectures we benefit in 3 major fields:
  - **Efficiency**:
    - Can reason quickly compared to pure symbolic systems, thereby reducing computational complexity.
    - This accelerated computation can be attributed to the integration of neural networks
  - **Generalization**:
    - Outperform standalone neural systems in terms of their capacity for generalization.
    - The incorporation of symbolic knowledge as valuable training data enhances the model's generalization abilities
  - **Interpretability**: By leveraging symbolic knowledge, these systems can provide explicit computation processes, such as traced reasoning processes or chains of evidence for results

Introduction



Learning For Reasoning

- Leverage symbolic systems for reasoning while incorporating the advantages of neural networks to facilitate finding solutions
- In this context there are two aspects to consider:
  - **Accelerator**
  - **Transformer**

Learning for Reasoning

Accelerator

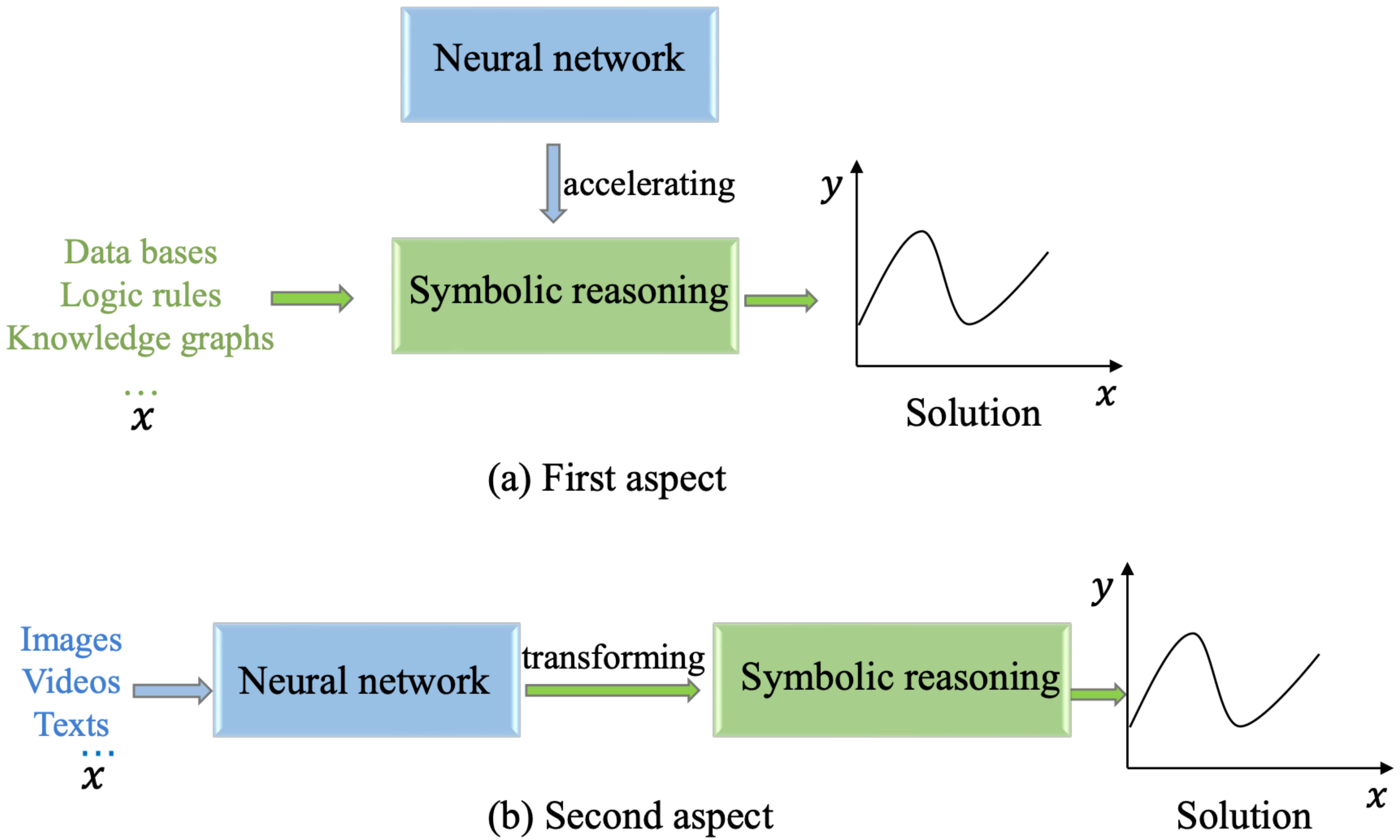
- Symbolic reasoning often involves searching for solutions in a solution space, which can be computationally expensive.
- In learning for reasoning approaches, neural networks are used as accelerators to speed up the search process.
- By leveraging the learning capabilities of neural networks, the search space can be reduced or optimized, leading to faster and more efficient reasoning.
- Techniques such as reinforcement learning can be employed to guide the search process and improve the overall performance of symbolic systems.

Learning for Reasoning

Transformer

- Symbolic systems traditionally operate on structured data represented by symbols.
- Neural networks excel at processing unstructured data such as images, texts, and videos.
- In these approaches, neural networks play a crucial role in transforming unstructured data into symbolic representations that can be processed by symbolic reasoning systems.
- By abstracting and extracting relevant features rom the unstructured data, neural networks provide the symbolic systems with meaningful inputs for reasoning and decision-making.

Learning for Reasoning



(a) First aspect

(b) Second aspect

Learning for Reasoning

pLogicNet (Accelerator)

- pLogicNet is designed to solve the problem of predicting the missing triplets in a KG through reasoning with the observed facts.
- Logic rule-based approach is the Markov Logic Network (MLN), which is able to leverage domain knowledge with first-order logic and meanwhile handle their uncertainty.
- Inference of MLNs is usually very difficult due to the complicated graph structures.
- Different from MLNs, knowledge graph embedding methods learn effective entity and relation embeddings for reasoning, which are much more effective and efficient however, they are unable to leverage domain knowledge.
- pLogicNet is proposed to integrate existing rule-based methods and knowledge graph embedding methods.
- pLogicNet models the distribution of all the possible triplets with a Markov logic network, which is efficiently optimized with the variational EM algorithm.
- In the E-step, a knowledge graph embedding model is used to infer the hidden triplets
- in the M-step, the weights of rules are updated based on the observed and inferred triplets.

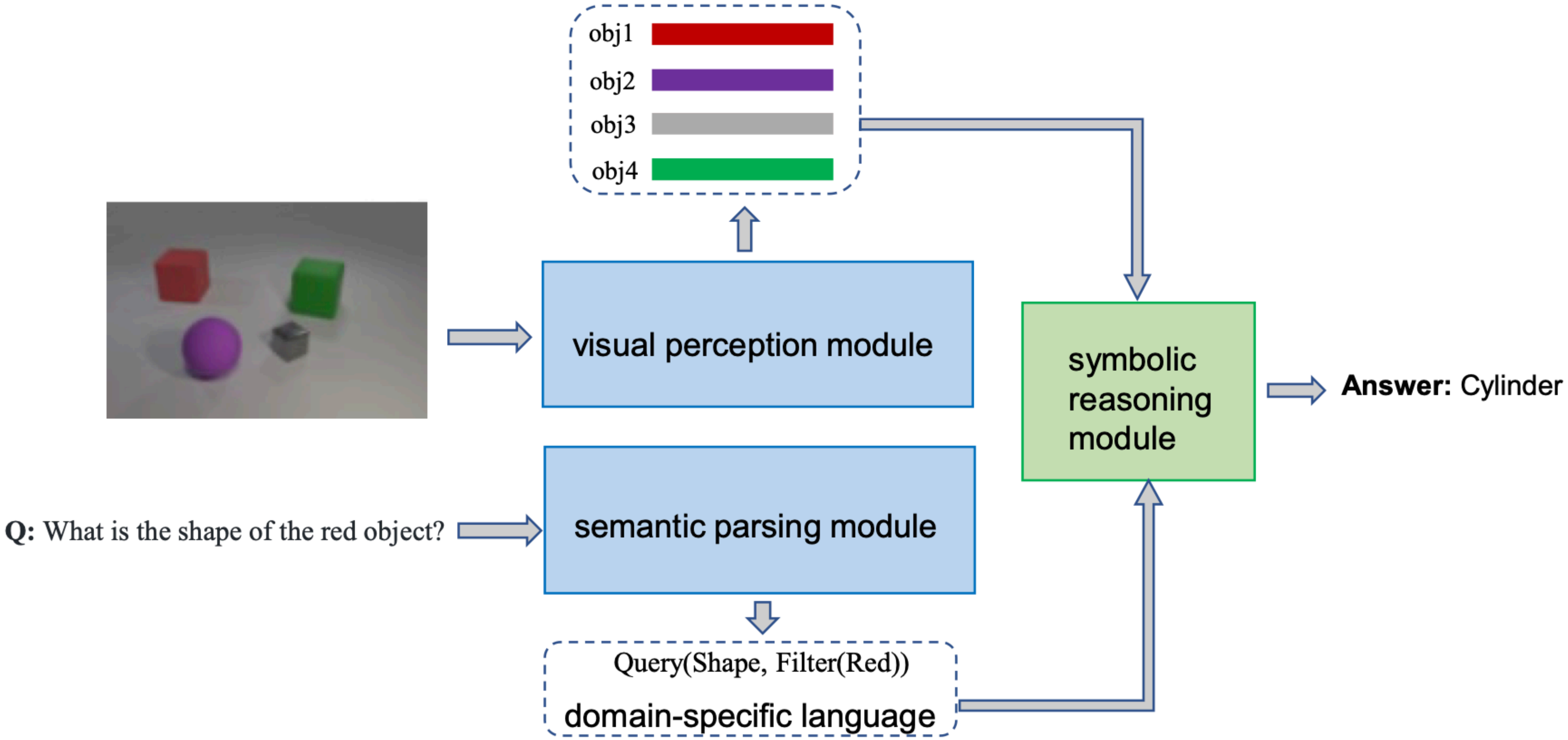
Learning for Reasoning

NS-CL Model (Transformer)

- NS-CL is designed to solve tasks in visual question answering (VQA).
- This model includes three modules:
  - A visual perception module.
  - A semantic parsing module
  - A symbolic reasoning module.
- The visual perception module extracts object-based symbolic representation for a scene in an image.
- The semantic parsing module transforms a question into an executable program.
- The symbolic reasoning module applies a quasi-symbolic program executor to infer the answer based on the representation and an executable program.

Learning for Reasoning

NS-CL model (Transformer)



Reasoning for Learning

- Utilisation of symbolic systems to support the learning process of neural systems
- Key idea is leverage neural systems for machine learning tasks while incorporating symbolic knowledge **into the training** process to enhance performance and interpretability.
- Symbolic knowledge is typically encoded in a format suitable for neural networks and used to guide or constrain the learning process. Most often is represented as a regularization term in loss function.
- It can be broadly divided into **regularization models** and ***knowledge***

Reasoning for Learning

Regularization Models

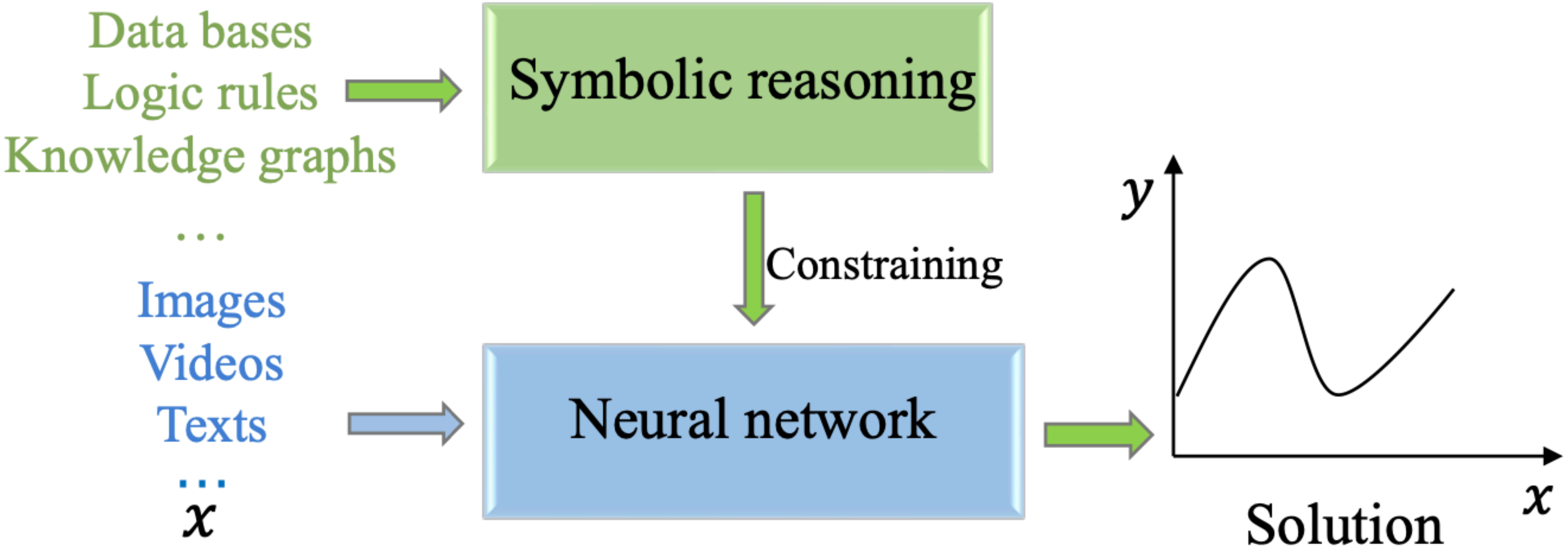
- Integrate symbolic knowledge into the training process by adding regular terms to the objective function of the model.
- Regular terms serve as constraints or penalties that encourage the model to adhere to the symbolic knowledge during training.
- Different regularization models may employ different strategies for modeling symbolic knowledge as regular terms, such as semantic embeddings and logical rules

Reasoning for Learning

Knowledge Transfer Models

- These models leverage existing symbolic knowledge or semantic information to guide the learning of models in a different domain.
- Knowledge transfer integrates knowledge graphs (pre-defined) that represent semantic information into deep learning models, which compensates for the lack of available data through the transfer of semantic knowledge.
- “Encode” symbolic knowledge into the structure of the neural networks as an initiative to improve their performance.

Reasoning for Learning



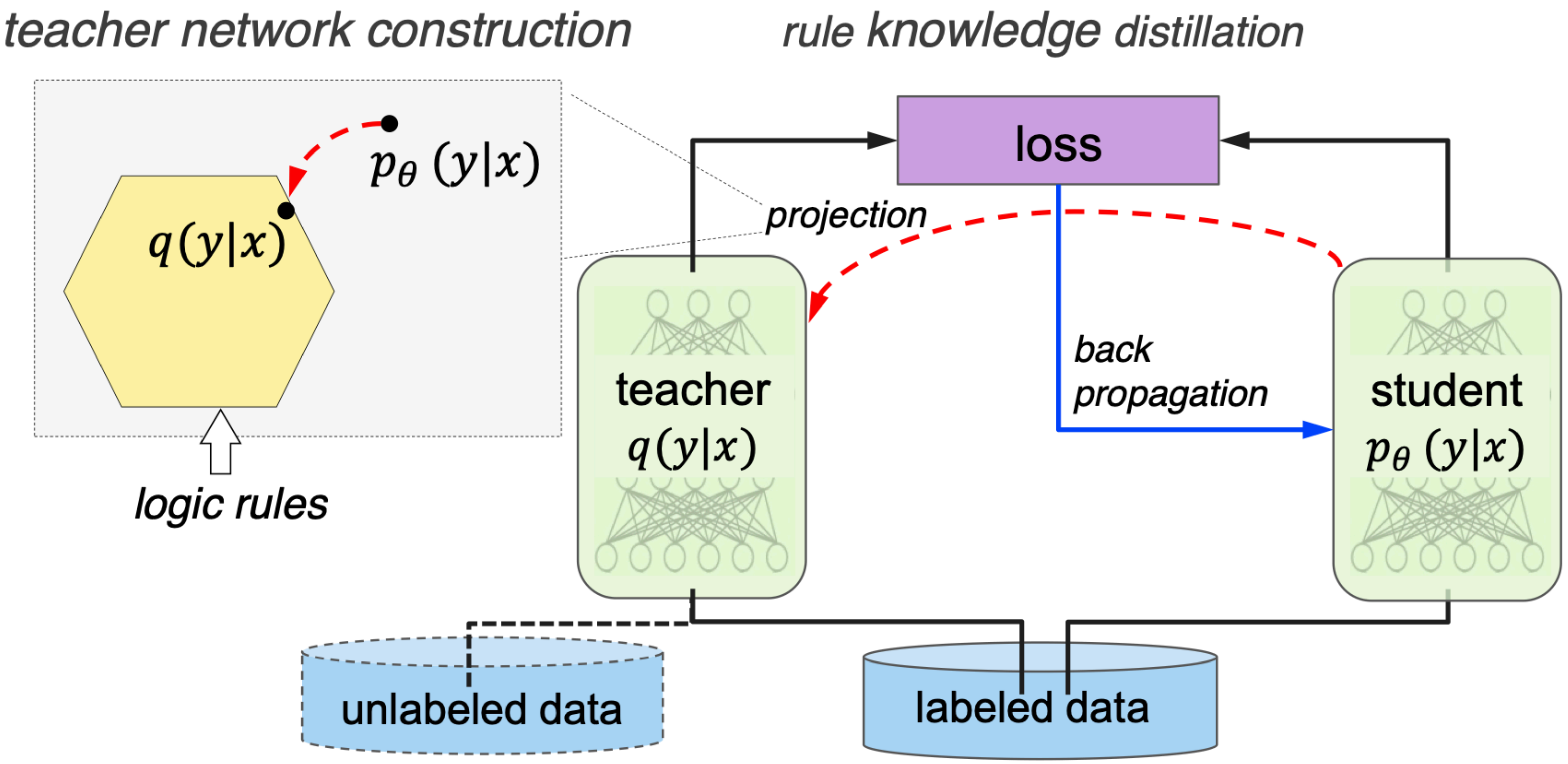
Reasoning for Learning

HDNN (Regularization)

- Leverages the concept of knowledge distillation to harness the power of logic rules in training deep neural networks.
- Consists of two components:
  - Teacher network
  - Student network
- Teacher network encodes logic rules and guides the student network during training.
- Teacher network learns information from the labeled data and logic rules (unlabeled data) and teaches the student network by the loss function.
- Based on the above process, the structured information encoded by logic rules can constrain the learning of the student network.

Reasoning for Learning

HDNN (Regularization)



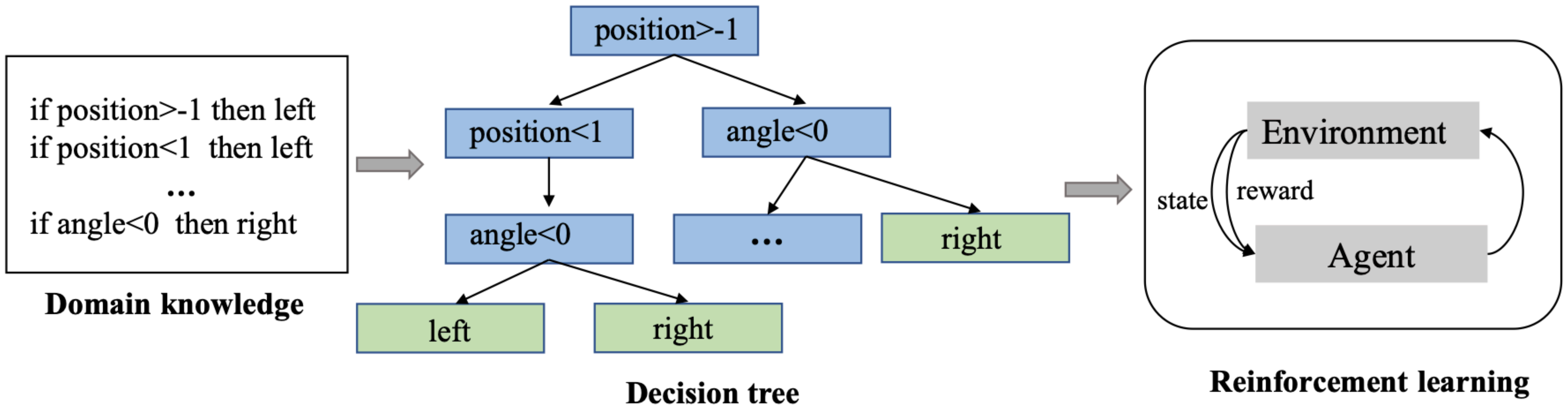
Reasoning for Learning

PROLONETS (Knowledge transfer)

- Propositional Logic Nets directly encode domain knowledge as a collection of propositional rules within a neural network.
- This approach enables the neural network to leverage domain-specific information and improve its learning and reasoning capabilities.
- Aids in “warm starting" the learning process in deep reinforcement learning.
- 3 basic steps:
    - Policies and actions express domain knowledge in the form of propositional rules
    - Neural network initialization, wherein the nodes of the decision tree are directly transformed into neural network weights.
    - Training, during which the initialized network interacts with the environment, collecting data that is subsequently used to update parameters and rectify domain knowledge

Reasoning for Learning

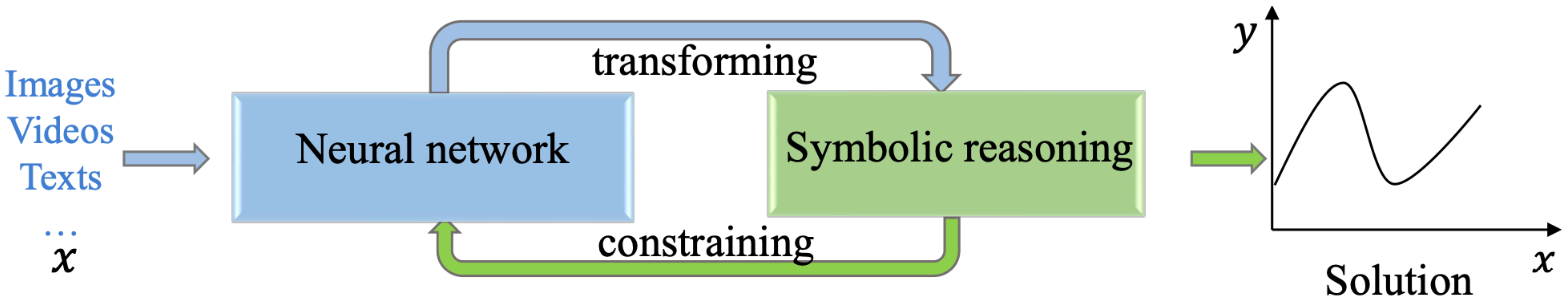
PROLONETS (Knowledge transfer)



Learning-Reasoning

- The interaction between neural systems and symbolic systems is bidirectional, with both paradigms playing equal roles and working together in a mutually beneficial way.
- Goal is to strike a balance between the involvement of neural systems and symbolic systems in the problem-solving process.
- The output of the neural network becomes an input to the symbolic reasoning component, and the output of the symbolic reasoning becomes an input to the neural network.
- Neural networks provide the ability to process complex data and generate predictions, while symbolic reasoning provides a structured and interpretable framework for representing and reasoning about knowledge.
- Many models follow the same principles:
  - Modelling of complex problems is defined in a logic programming language.
  - Neural network is used to define simple concepts in a logic programming language.

Learning-Reasoning



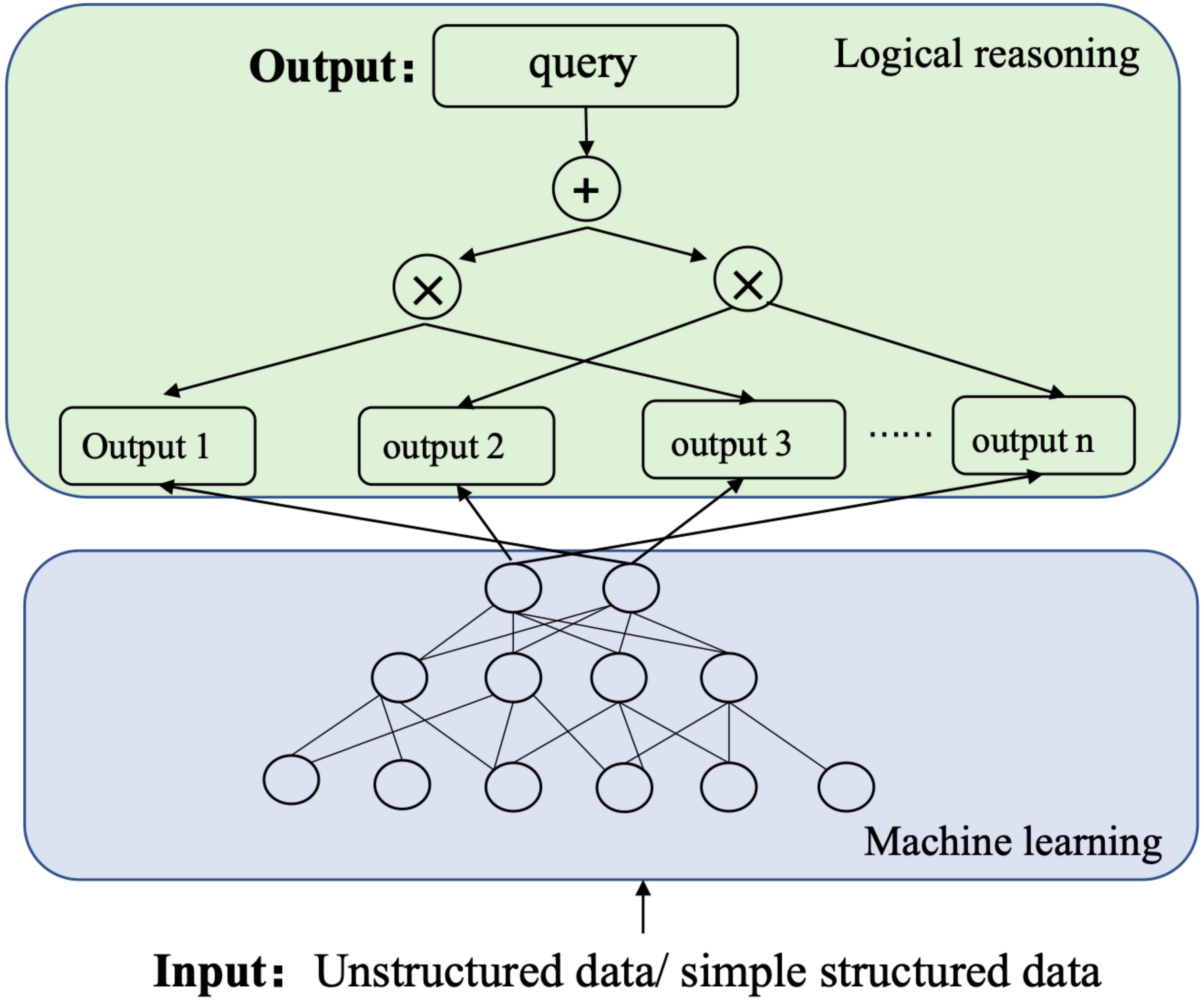
Learning-Reasoning

DeepProbLog

- Based on ProbLog
- A framework that combines a generic deep neural network with probabilistic logic.
- ProbLog + Neural Predicates
- Intuition behind neural predicates:
  - In a probabilistic logic, atomic expressions of the form $q(t_1, \ldots, t_n)$ have a probability p.
  - Extend this idea by allowing atomic expressions to be labeled with neural networks whose outputs can be considered probability distributions
  - Consequently, the output of neural network components can be encapsulated in the form of “neural” predicates as long as the output of the neural network on an atomic expression can be interpreted as a probability.

Learning-Reasoning

DeepProbLog



Learning-Reasoning

DeepProbLog Example (1)

- Consider the predicate $addition(X, Y, Z)$ where X and Y are images of digits and Z is the natural number corresponding to the sum of these digits.

- The goal is that after training, DeepProbLog allows us to make a probabilistic estimate on the validity of our addition addition(, , 8)

- While such a predicate can be learned directly by a standard neural classifier, such an approach cannot incorporate background knowledge such as the definition of the addition of two natural numbers.

- In DeepProbLog such knowledge can easily be encoded in rules

DeepProbLog Program

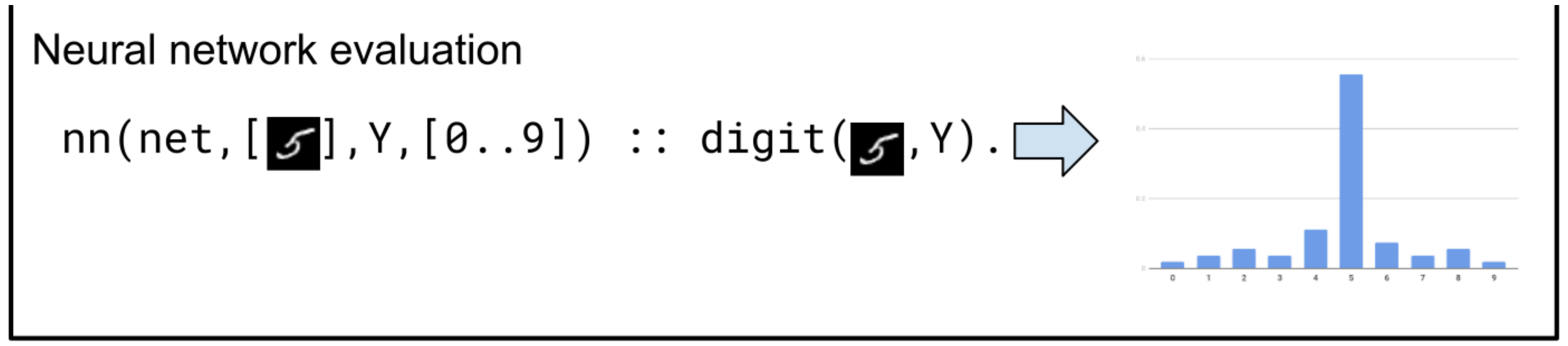
```
%Neural predicate
nn(net,[X],Y,[0..9]) :: digit(X,Y).

%Background knowledge
addition(X,Y,Z):- digit(X,N1),digit(Y,N2),
                  Z is N1+N2.
```

Learning-Reasoning

DeepProbLog Example (2)

- All that needs to be learned in this case is the neural predicate digit which maps an image of a digit to the corresponding natural number



- The trained network can then be reused for arbitrary tasks involving digits.

- It's also important to note that the neural network trained inside the DeepProbLog model can recognize single digits, whereas the convolutional baselines can only classify sums.

- The separation between the logic and neural aspects results in a more flexible model