Integrating Automated Reasoning with Machine Learning for Structured Prediction and Scientific Discovery

Nan Jiang
Department of Computer Science, Purdue University

Committee: Yexiang Xue (Chair), Willem-van Jan Hoeve, Jean Honorio and Brian Bullins.

Two Pillars in AI: Machine Learning and Automated Reasoning

Machine Learning

Bottom-up and Inductive: Fit data distributions well.

- E.g.,
  - Perceptron
  - Support vector machine
  - Generative model

ChatGPT

Midjourney

Automated Reasoning

- Top-down and deductive: precise models from problem description.
- E.g.,
  - SATisfiability (SAT) solvers
  - Satisfiability Module Theory (SMT) solver
  - Mixed Integer Programming (MIP) solver

SAT Z3 IBM CPLEX

Two Pillars in AI: Machine Learning and Automated Reasoning

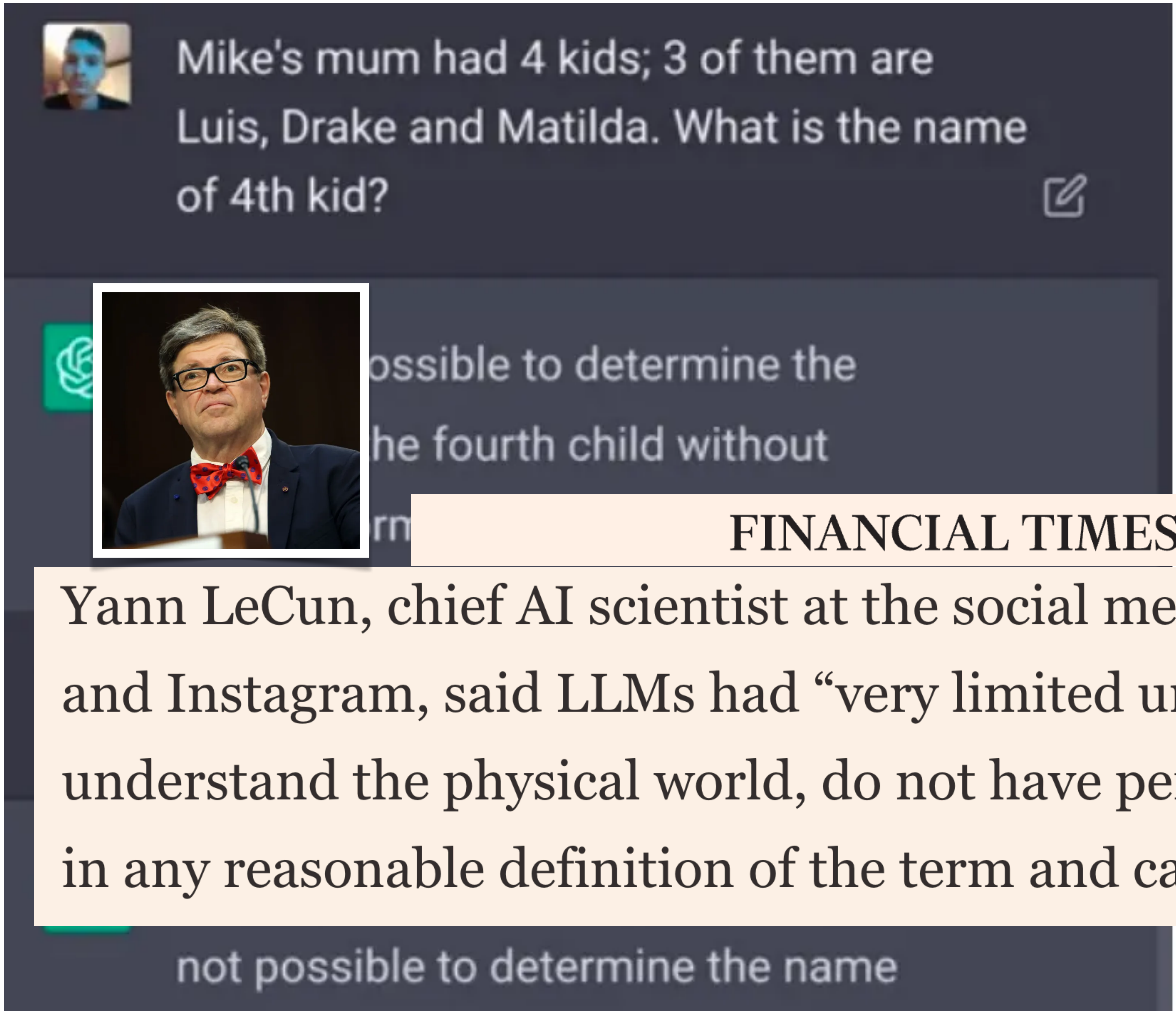
Machine Learning

- Challenging in providing formal guarantees.
- Hallucination: generated outputs are false or fabricated.
- May violate constraints in rare and unseen situations.

Automated Reasoning

- Rigid models: problem formulation must be agreed a-priori.
- Difficult to adapt to evolving data distributions.
- Cannot understand data like text and images.

Machine Learning has intrinsic difficulty



ChatGPT struggle with questions in logical reasoning and context comprehension.

FINANCIAL TIMES

Yann LeCun, chief AI scientist at the social media giant that owns Facebook and Instagram, said LLMs had "very limited understanding of logic . . . do not understand the physical world, do not have persistent memory, cannot reason in any reasonable definition of the term and cannot plan . . . hierarchically".

Automated Reasoning has intrinsic difficulties

Input $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$

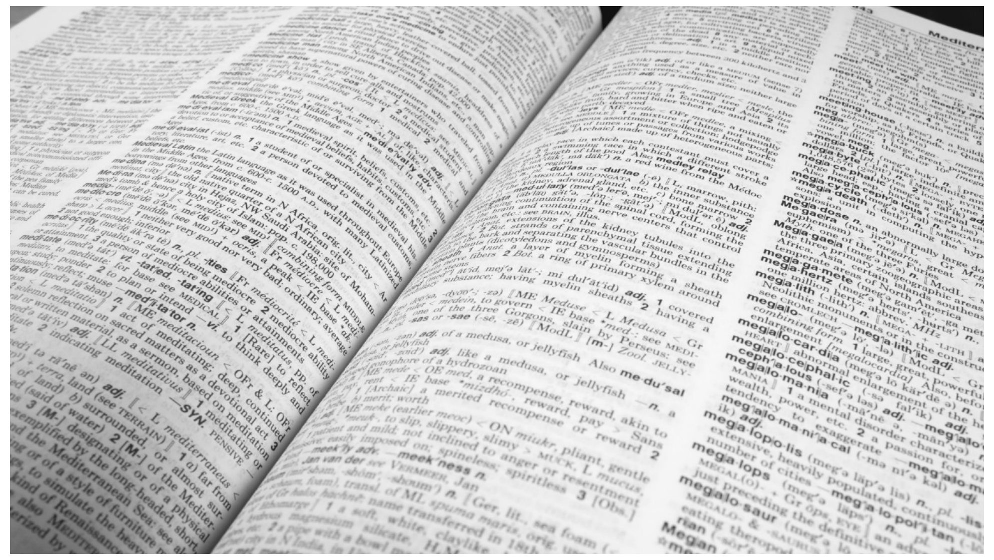
↓

SATisfiability Solvers

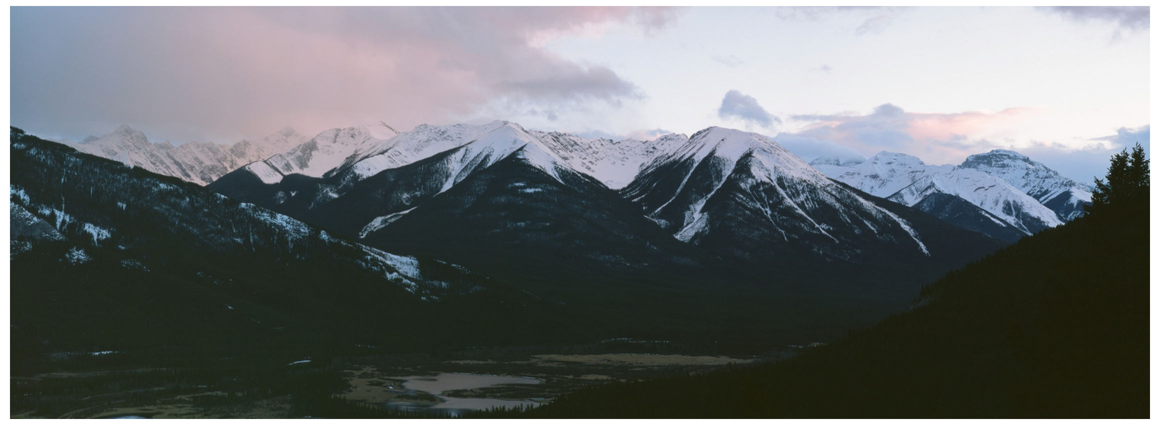
↓

Feasible variable assignment

- hard to encode data distribution.
- hard to handle complex input data, like
  - Millions of words in language



  - Millions of pixels in image



Bridging Machine Learning and Automated Reasoning is Crucial!

	Machine Learning	Automated Reasoning
Good at	Learn data distribution	Feasible output
Difficult to	Provide formal guarantees	Encode evolving data distribution

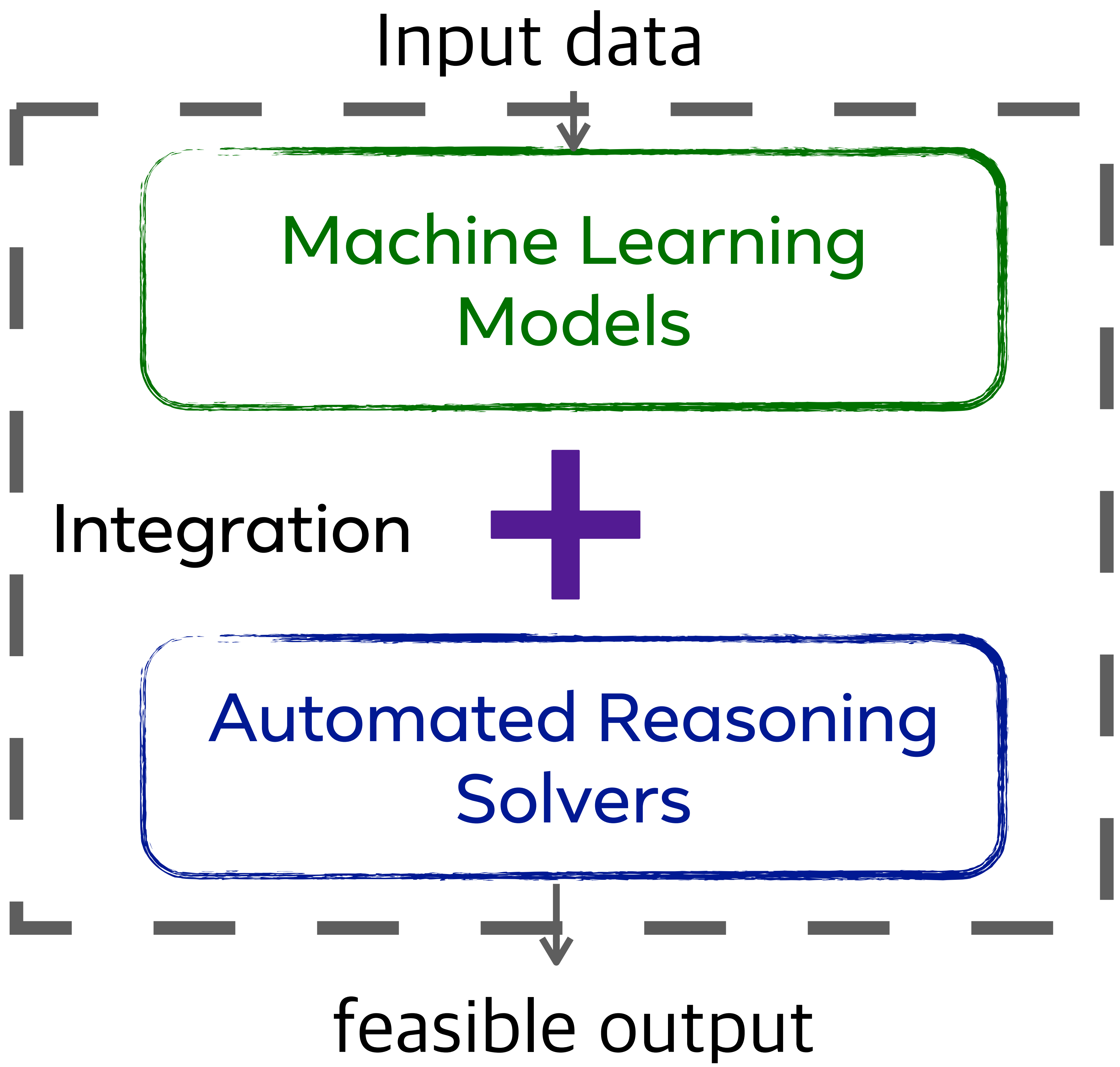
Structured prediction and scientific discovery problems are beyond the reach of machine learning and automated reasoning, when they are applied in isolation.

My Research: Integrate Learning with Reasoning

Key insight: Embed diverse reasoning solvers as differentiable modules into neural networks.

The benefits are:

- Formal guarantee of constraint satisfaction.
- Scalability: Accelerate learning for higher-dimensional data.



Outline

1 Formal guarantee: Integrate reasoning with learning to ensure constraint satisfaction for structured prediction.

Jinzhao Li, **Nan Jiang ,** et al**.** AAAI 2024. **Nan Jiang** et al. AAAI, 2023. **Nan Jiang** et al. JMLR, 2022. **Nan Jiang** et al., . UAI 2021. Maosen Zhang, **Nan Jiang**, et al. EMNLP 2020.

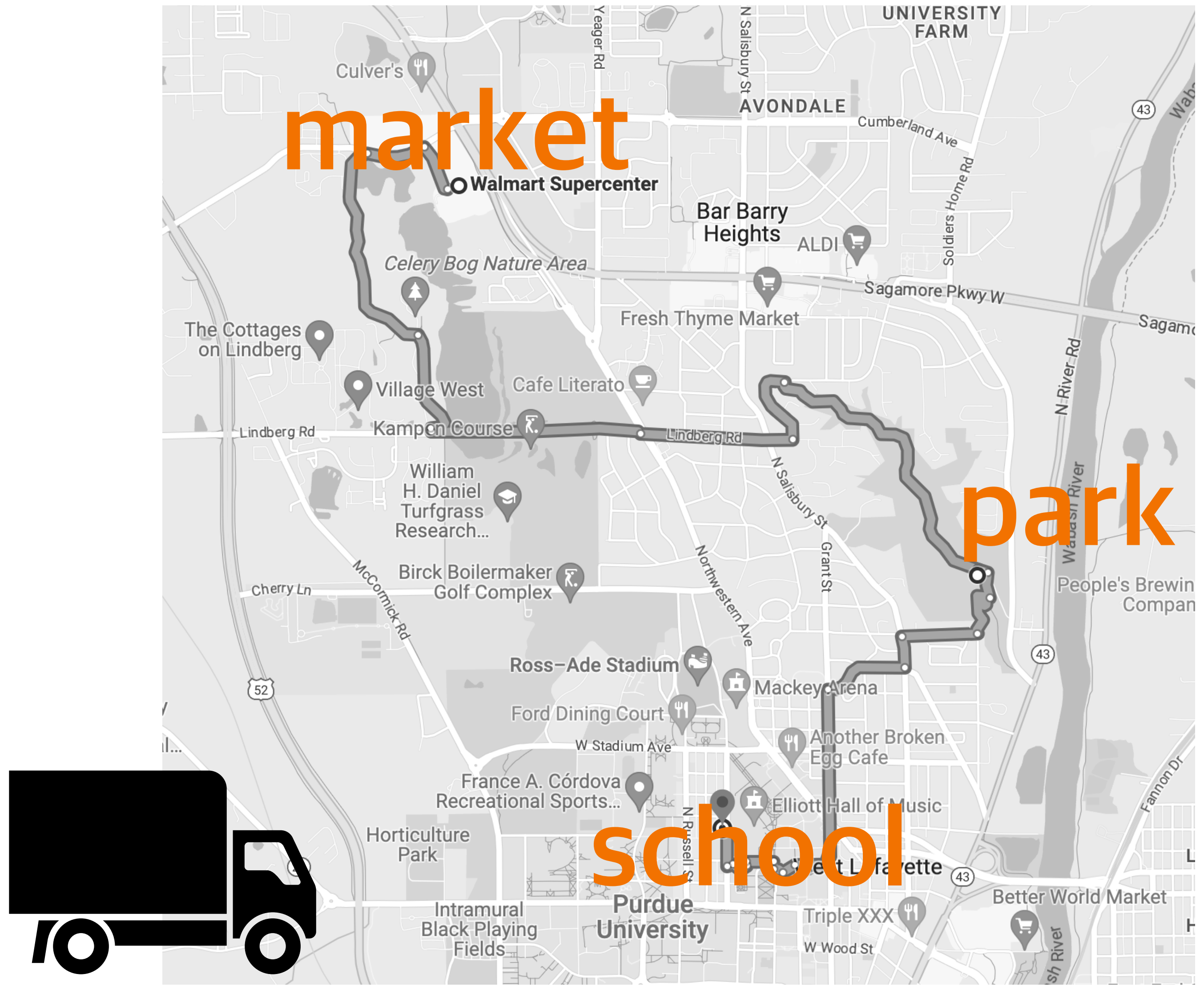
2 Scalability: Integrate reasoning with learning to accelerate scientific discovery.

3 Future work

Example 1: Delivery Route Planning

Task: Recommend routes that
- satisfy delivery requests;
- meet agent' implicit preferences.

Historical Dataset:
- Input: $\{$market, park, school$\}$
- Output: market $\rightarrow$ park $\rightarrow$ school.



	Machine Learning (e.g., Transformer)	Reasoning Solvers (e.g., traveling salesman problem solver)
Good at	Learn agent' preferences	Generate a feasible route
Difficult to	Always satisfy delivery requests	Extract and encode implicit preferences

Nan Jiang et al., Constraint Reasoning Embedded Structured Prediction. JMLR, 2022.

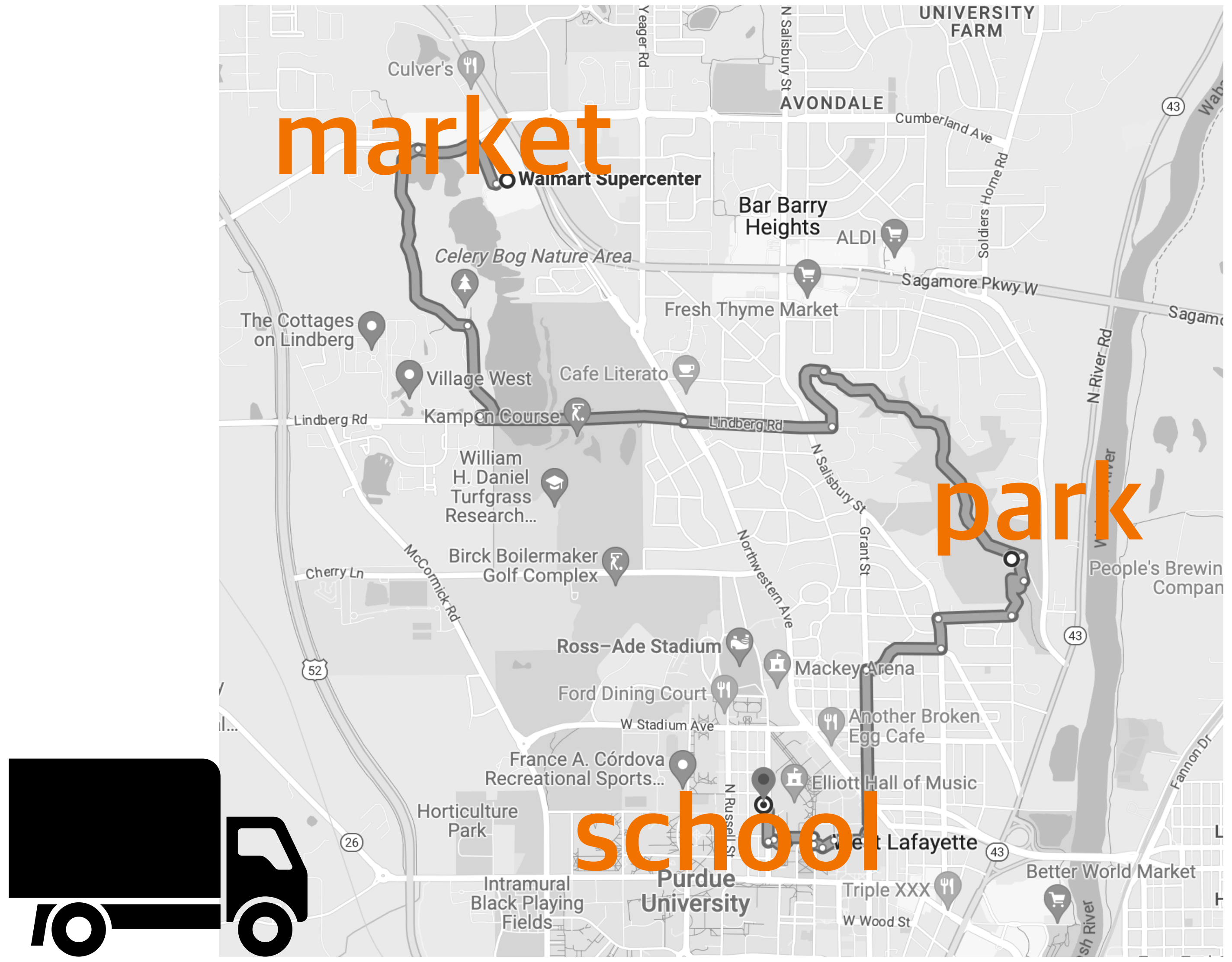
Example 1: Delivery Route Planning

Task: Recommend routes that

- satisfy delivery requests;
- meet agent' implicit preferences.

Historical Dataset:

- Input: {market, park, school}
- Output: market → park → school.



Our integrated system (neural network + reasoning solver):

- Neural network: Learn agent' implicit preferences.
- Reasoning solver: Satisfy delivery requests.

Nan Jiang et al., JMLR, 2022.

Example 2: Code generation from language

Task: predict a SQL program that
- Understand user query in natural language;
- The program is executable.

Input Query:

How many schools did player number 3 play at?

Output SQL Query:

SELECT COUNT “School” WHERE “No.” = “3”

Input Table:

	Player	No.	Position	School
0	Antonio	21	Guard-Forward	Duke
1	Voshon	2	Guard	Minnesota
2	Marin	3	Guard-Forward	Butler CC

	Machine Learning (i.e., Transformer)	Reasoning Solver (i.e, SQL grammar engine)
Goot at	understand the natural language	Generate executable SQL query
Difficult to	Always generate executable SQL query	understand the natural language

Nan Jiang et al., JMLR, 2022.

Example 2: Code generation from language

Task: predict a SQL program that
- Understand user query in natural language;
- The program is executable.

Input Query:
How many schools did player number 3 play at?

Output SQL Query:
SELECT COUNT “School” WHERE “No.” = “3”

Input Table:

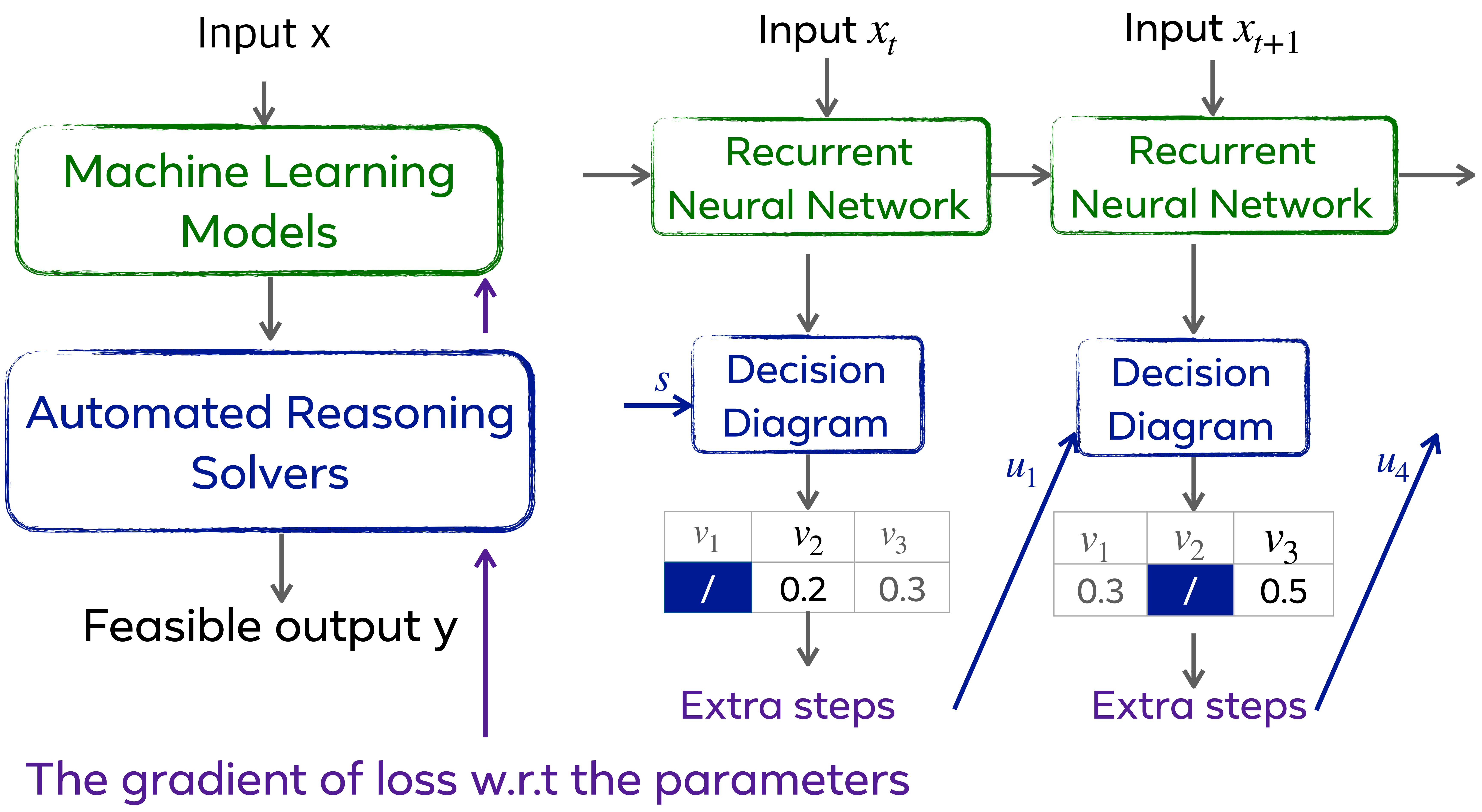
	Player	No.	Position	School
0	Antonio	21	Guard-Forward	Duke
1	Voshon	2	Guard	Minnesota
2	Marin	3	Guard-Forward	Butler CC

Our integrated system (neural network + reasoning solvers):
- Neural network: understand the natural language;
- Reasoning solver: satisfy the SQL grammar.

Nan Jiang et al., JMLR, 2022.

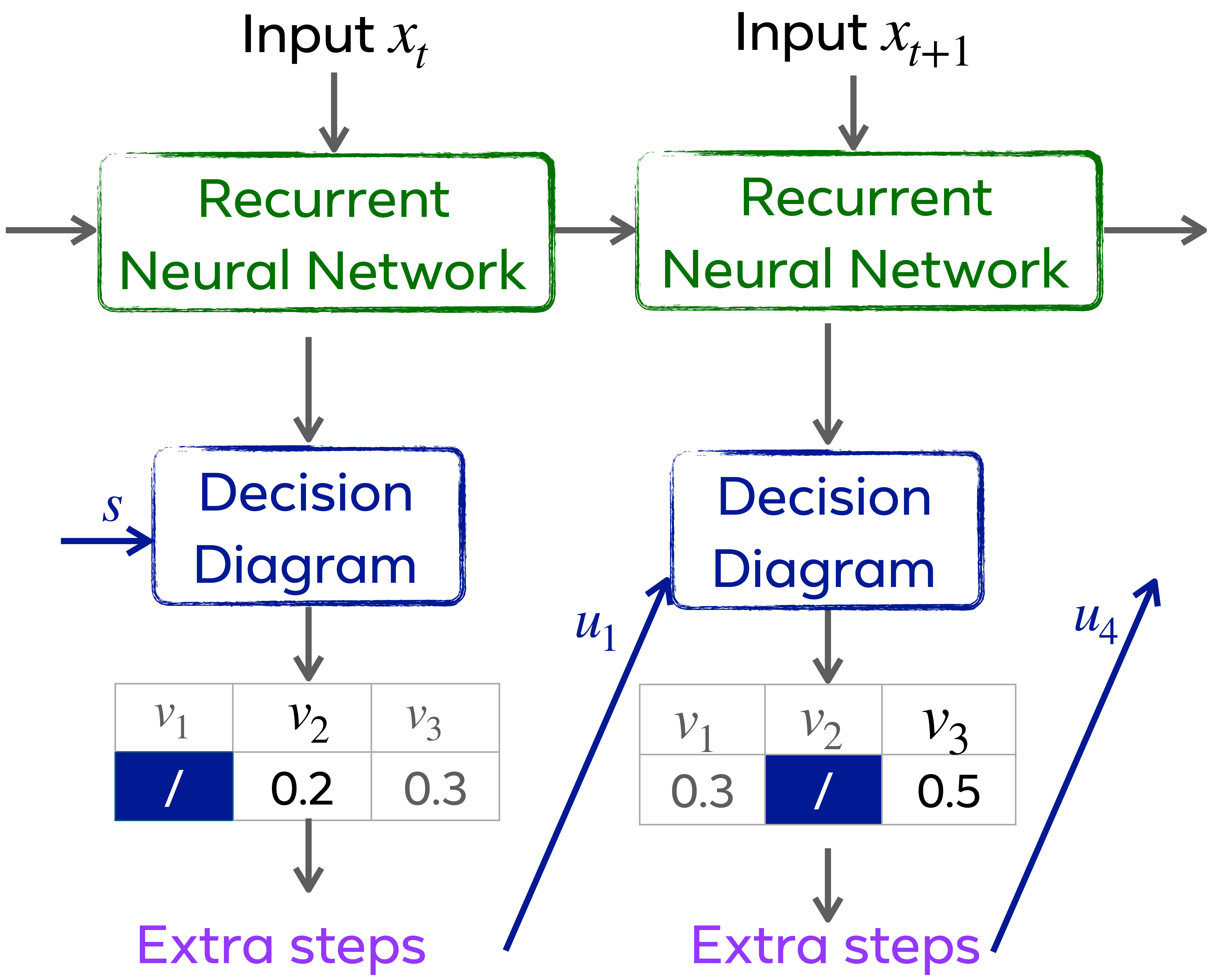
Design principle of the integrated system

- Learn from data
- Satisfy different types of constraints
- Differentiable



Design principle of the integrated system

Our solution:
COnstraint REasoning embedded Structured Prediction (CORE-SP)



Nan Jiang et al., JMLR, 2022.

Compile constraints as Decision Diagram

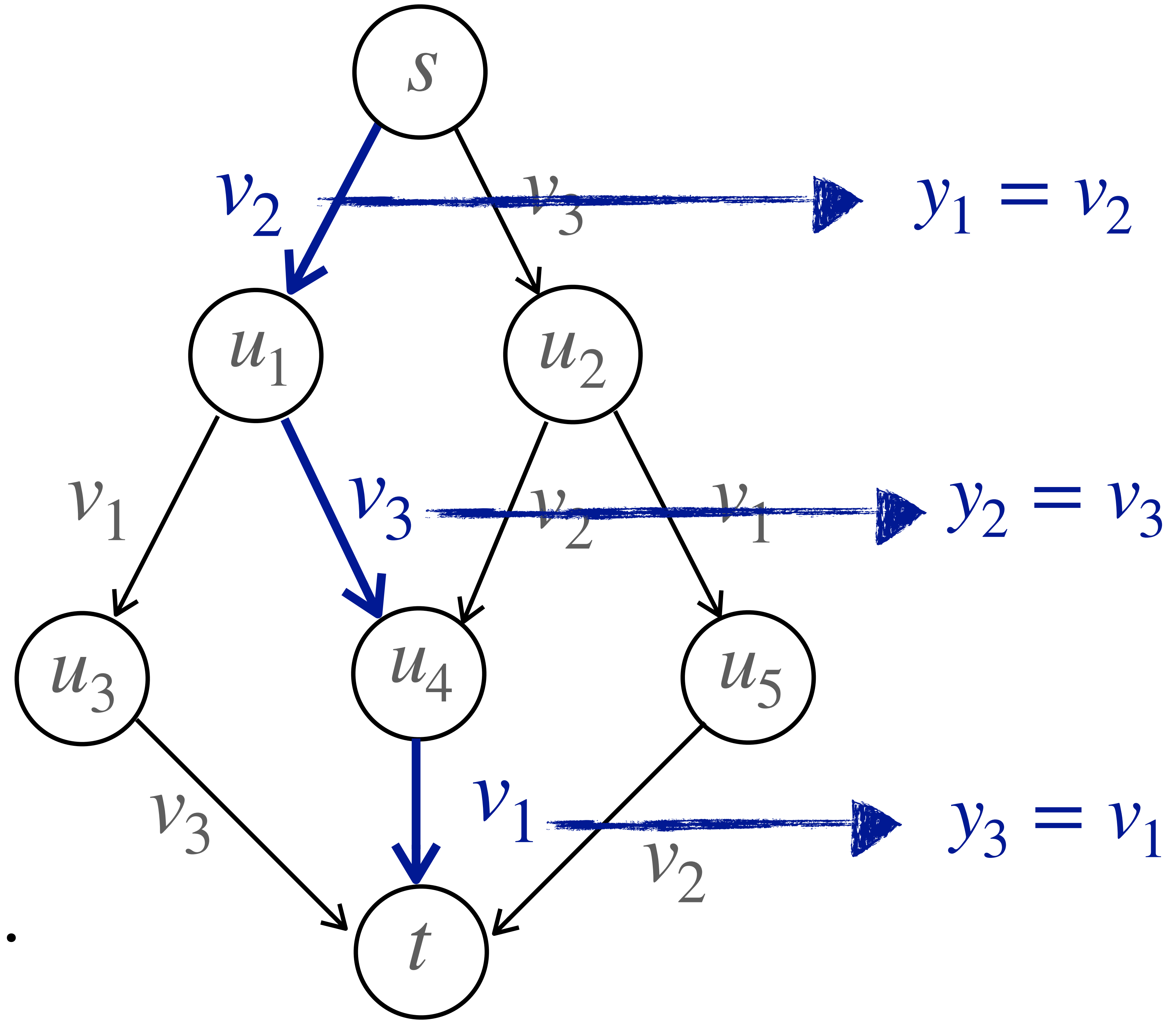
Decision Diagram: represents feasible solutions to combinatorial optimization problem as a space-compact directed acyclic graph.

Three **variables** $\{y_1, y_2, y_3\}$ takes values in $\{v_1, v_2, v_3\}$.

Every path in the graph is a variable assignment.

Compilation: delete paths that violate constraints.

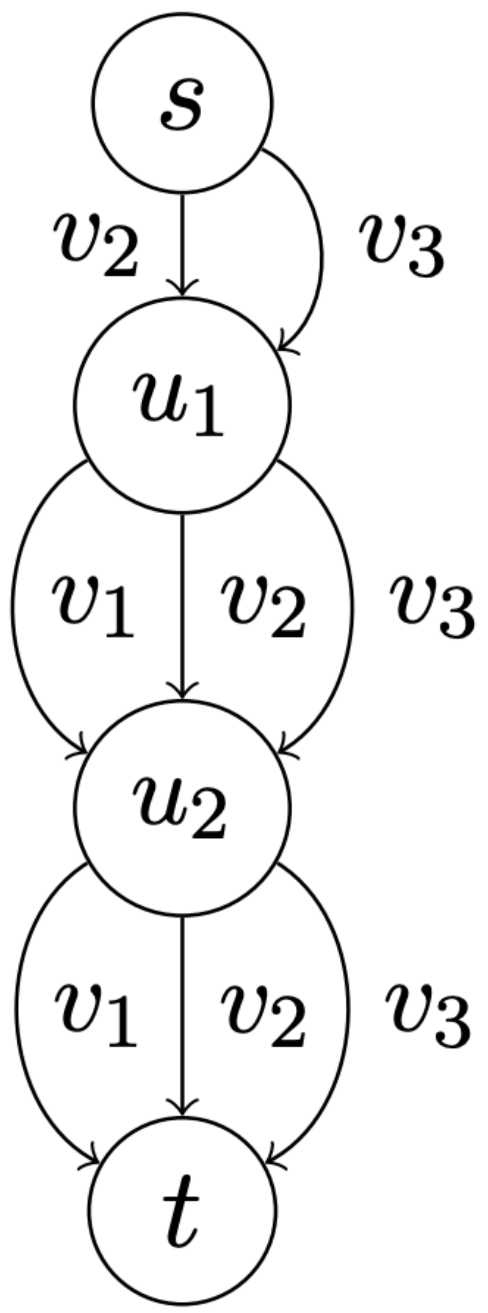
Extra heuristics: merge paths to reduce memory space.



Example decision diagram

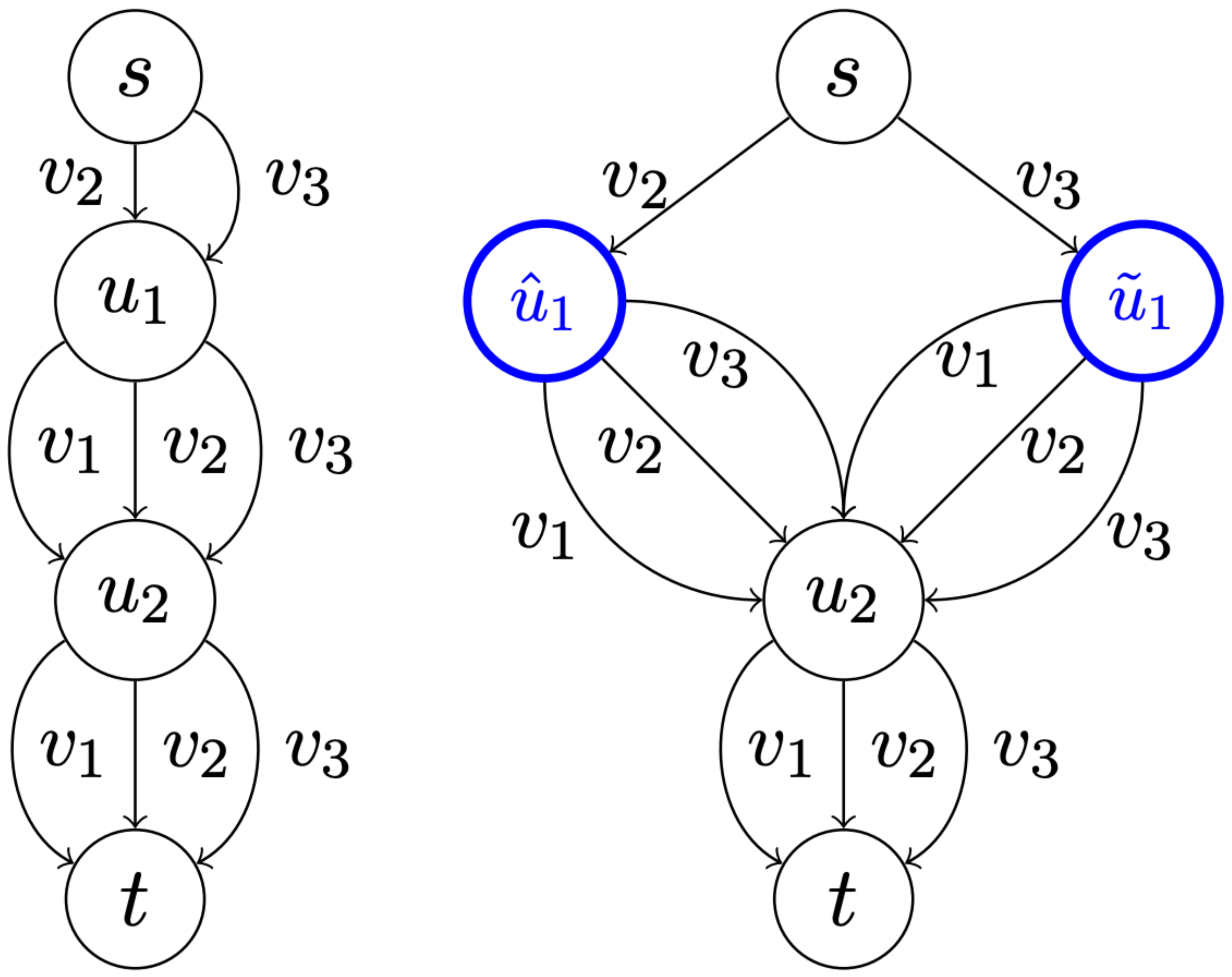
Nan Jiang et al., JMLR, 2022.

Example Compilation: delete paths that violate constraints



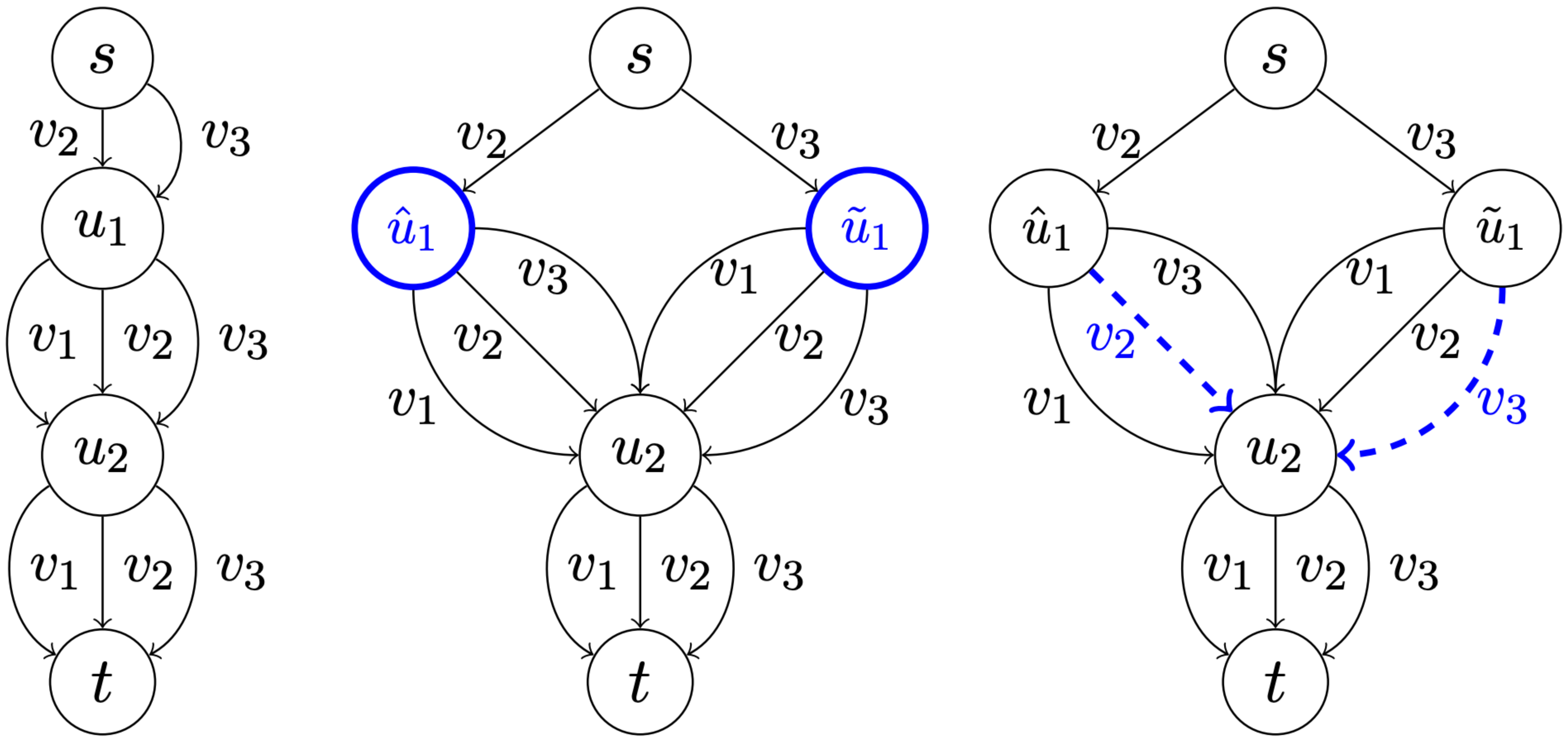
(a) width-1 MDD

Example Compilation: delete paths that violate constraints



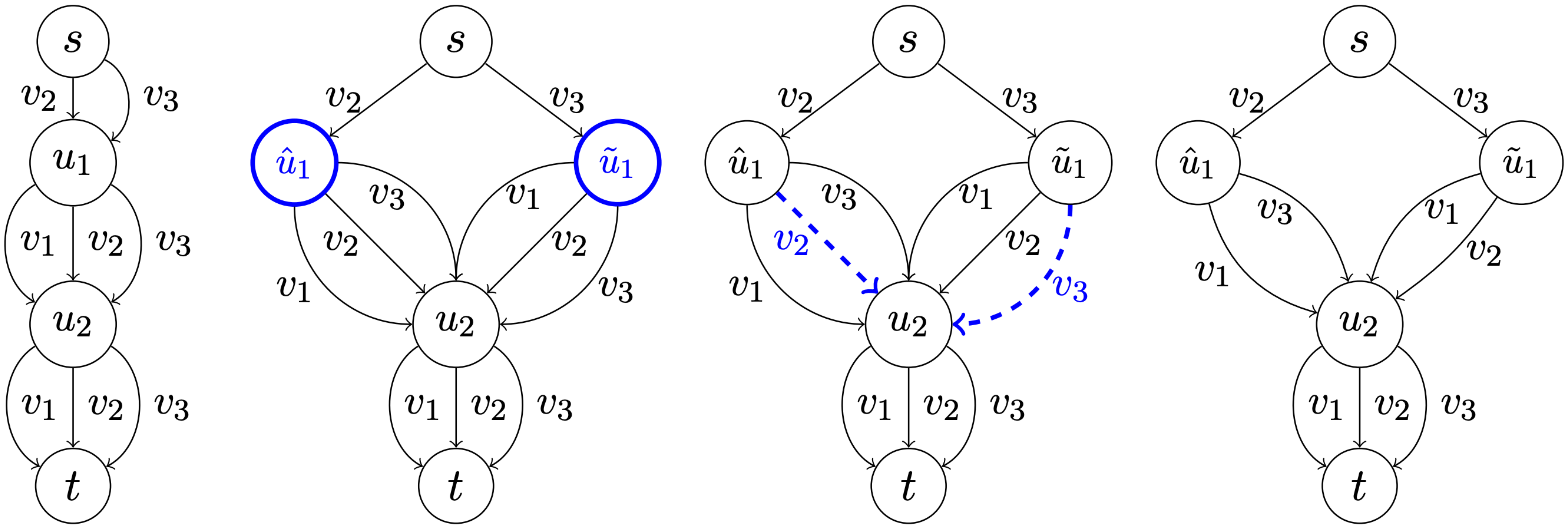
(a) width-1 MDD (b) split node

Example Compilation: delete paths that violate constraints



(a) width-1 MDD (b) split node (c) filter edges

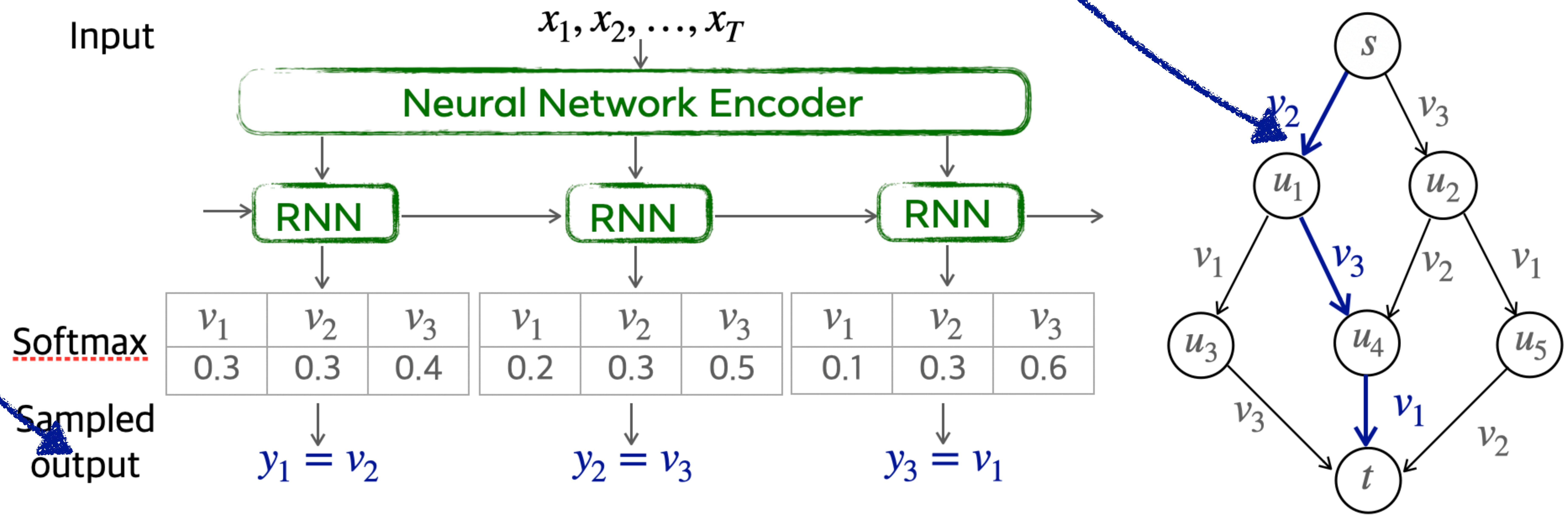
Example Compilation: delete paths that violate constraints



(a) width-1 MDD (b) split node (c) filter edges (d) width-2 MDD

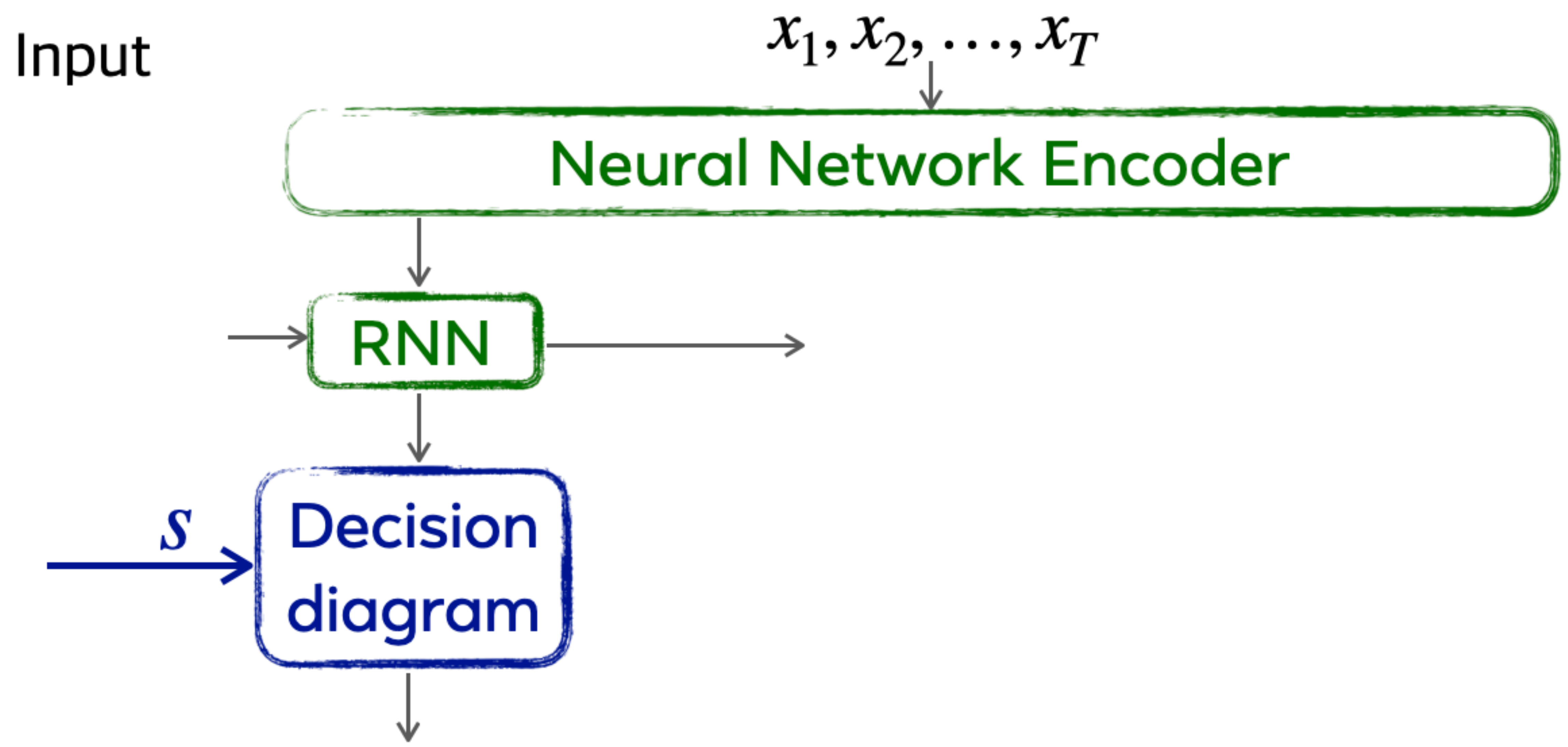
Neural net encodes data distribution; Decision diagram filters invalid predictions

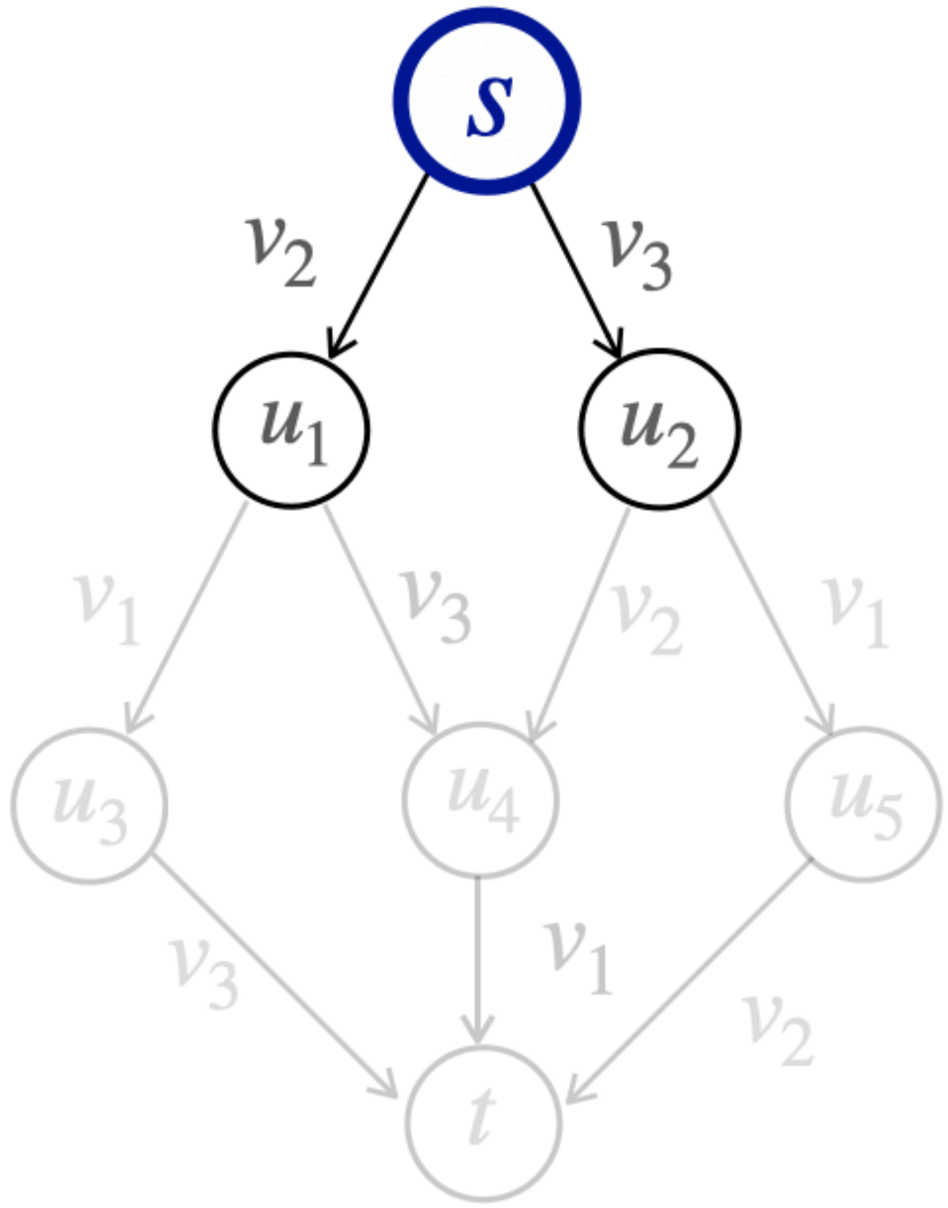
An output from the sequential decoder corresponds to **a path** in the decision diagram.



Nan Jiang et al., JMLR, 2022.

Execution step 1

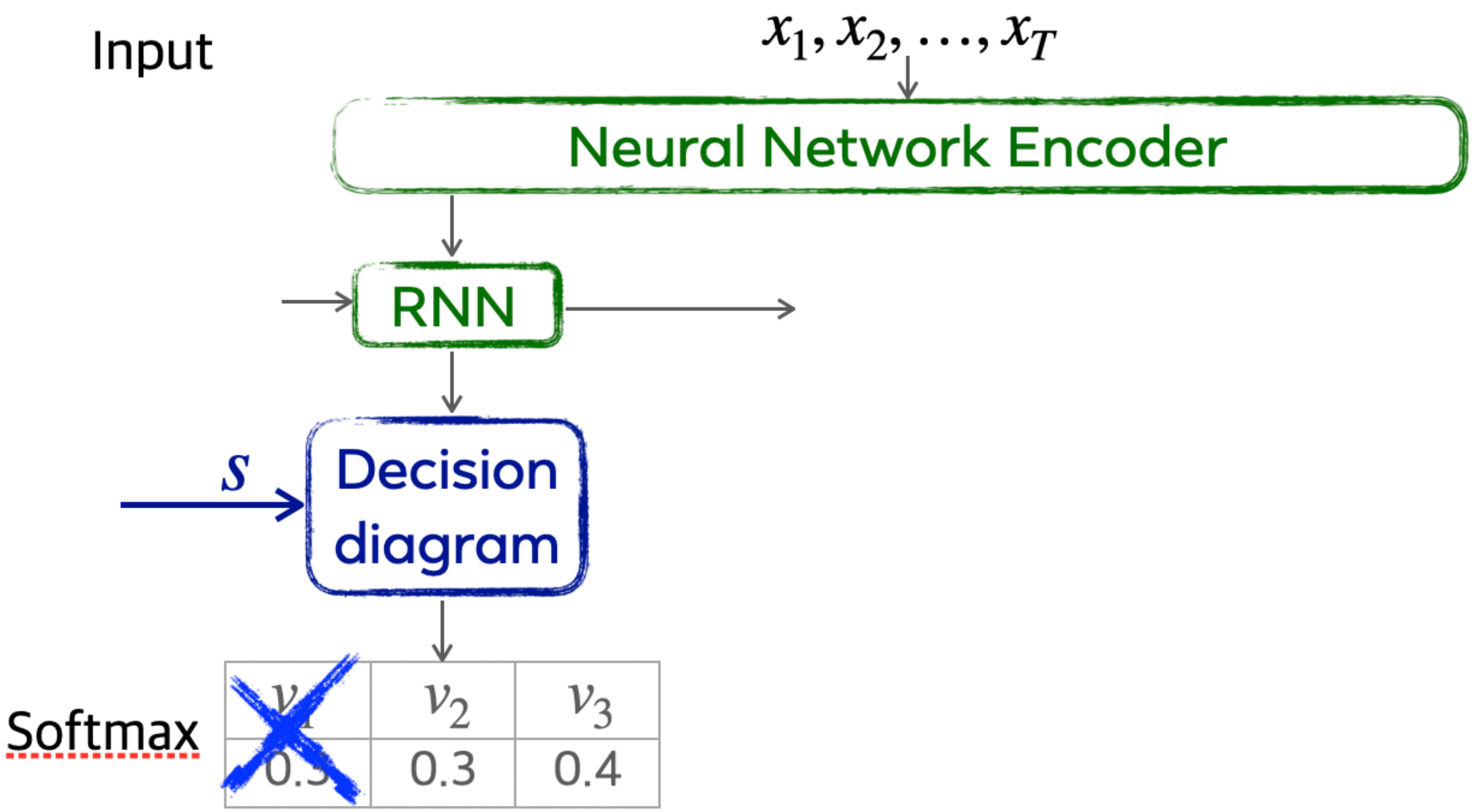


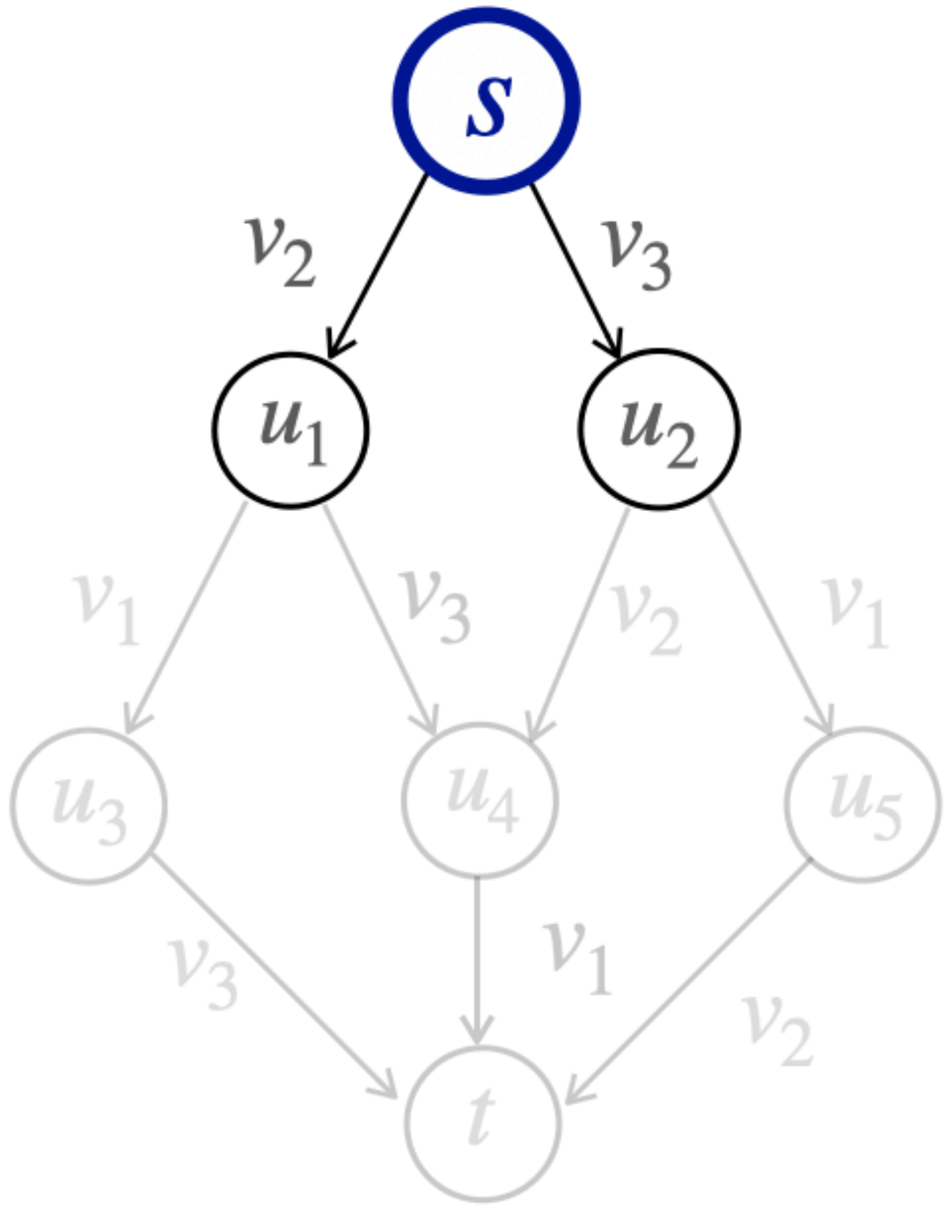


At node s

Nan Jiang et al., JMLR, 2022.

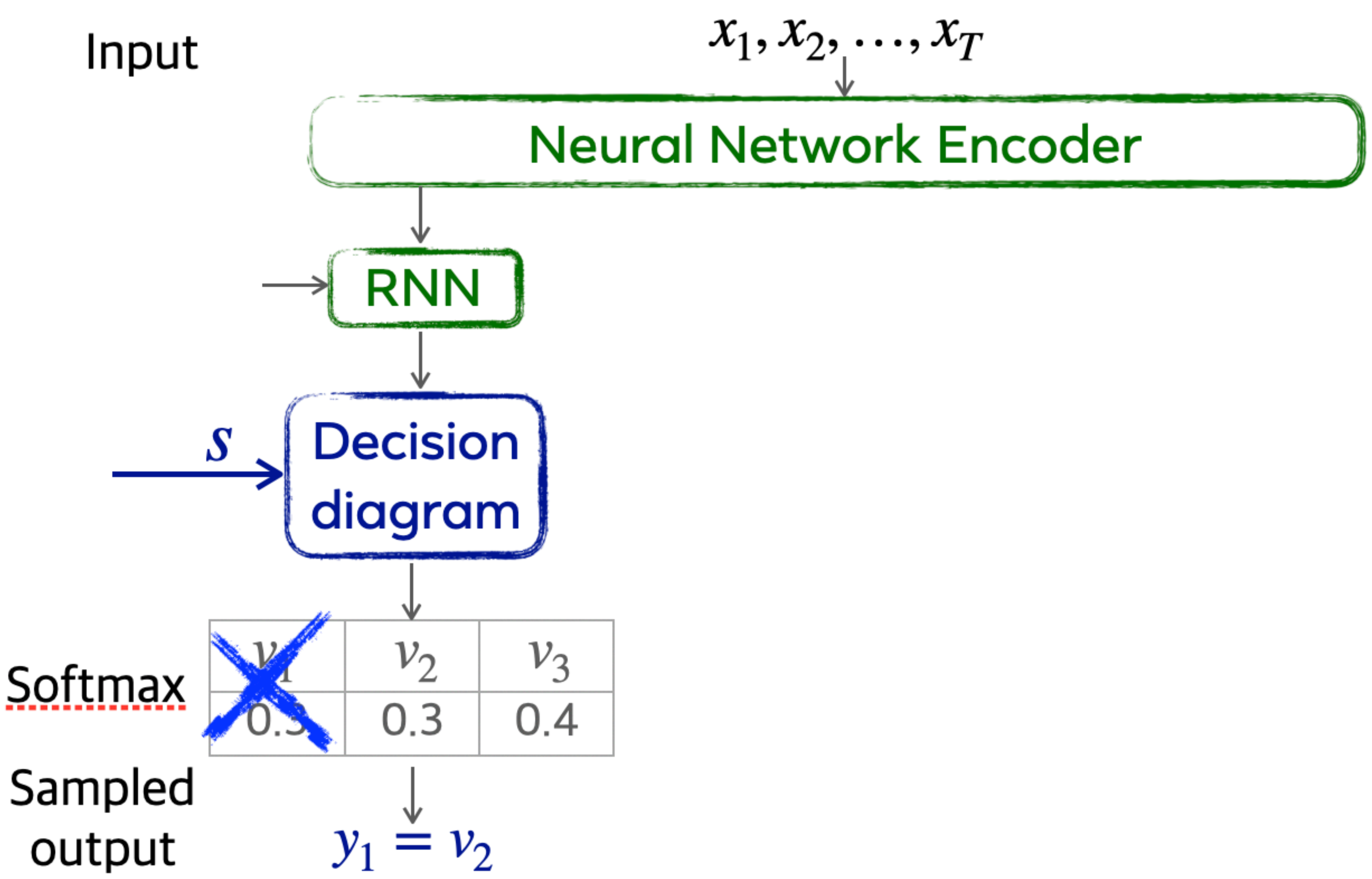
Execution step 1

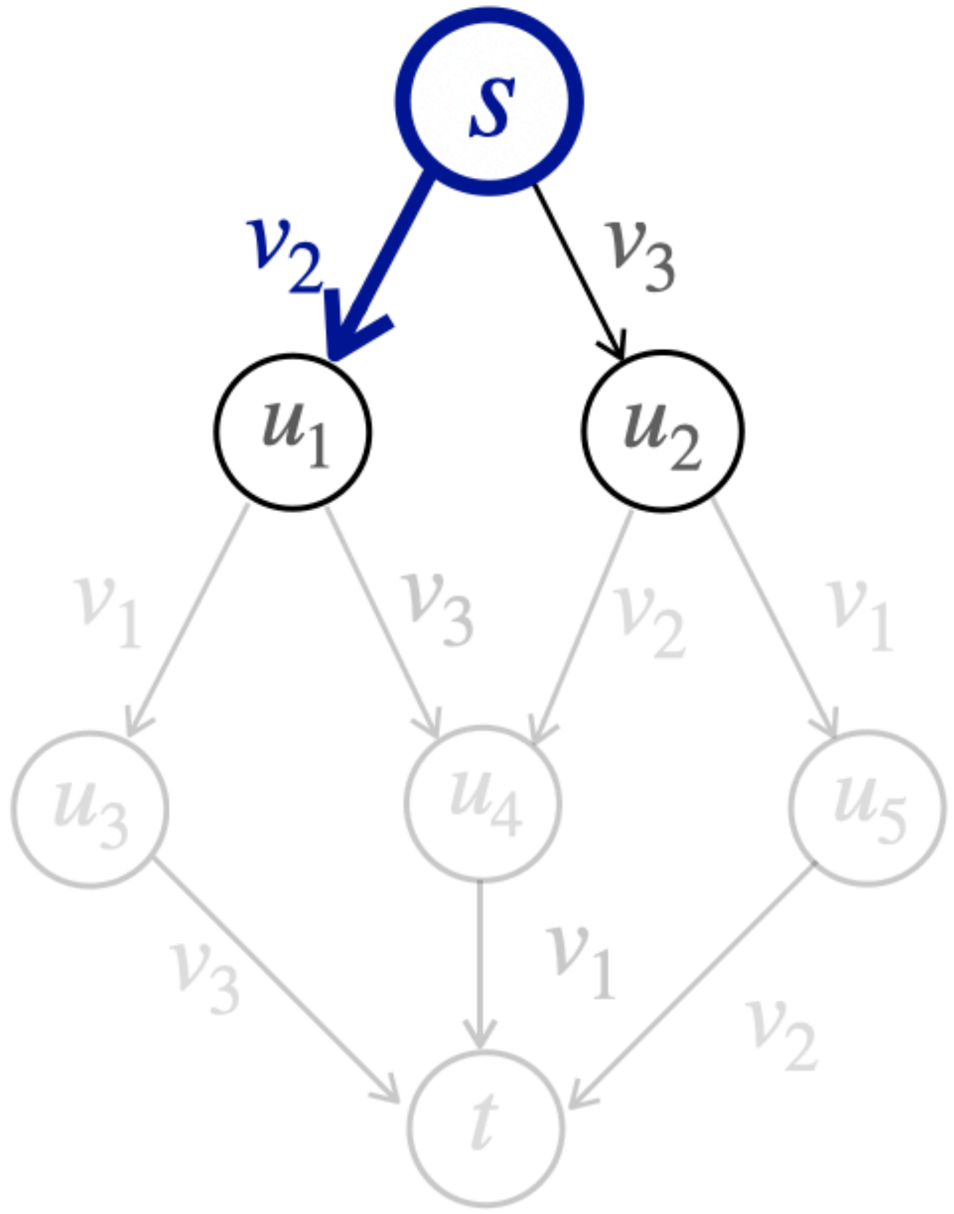




At node s

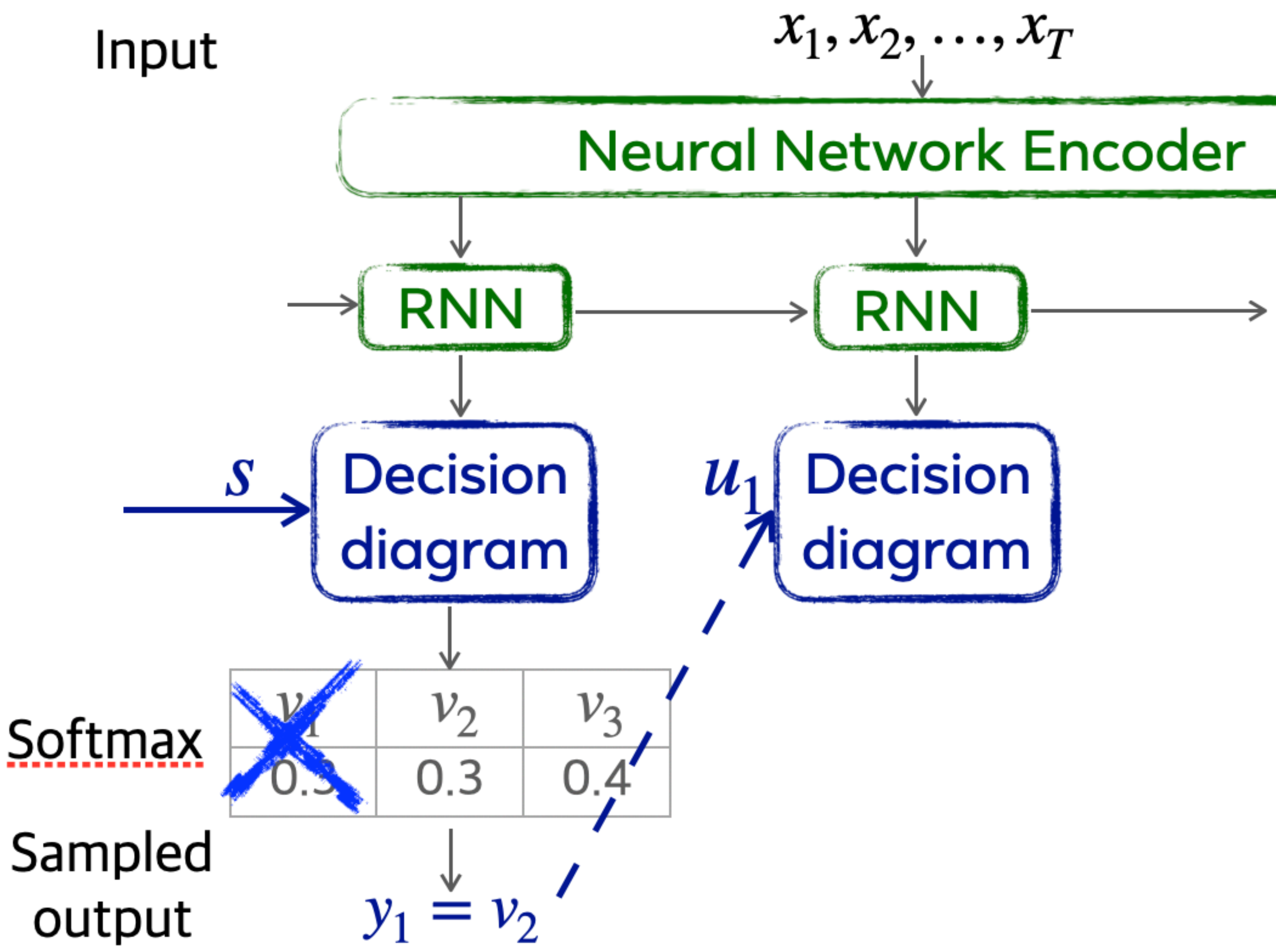
Execution step 1

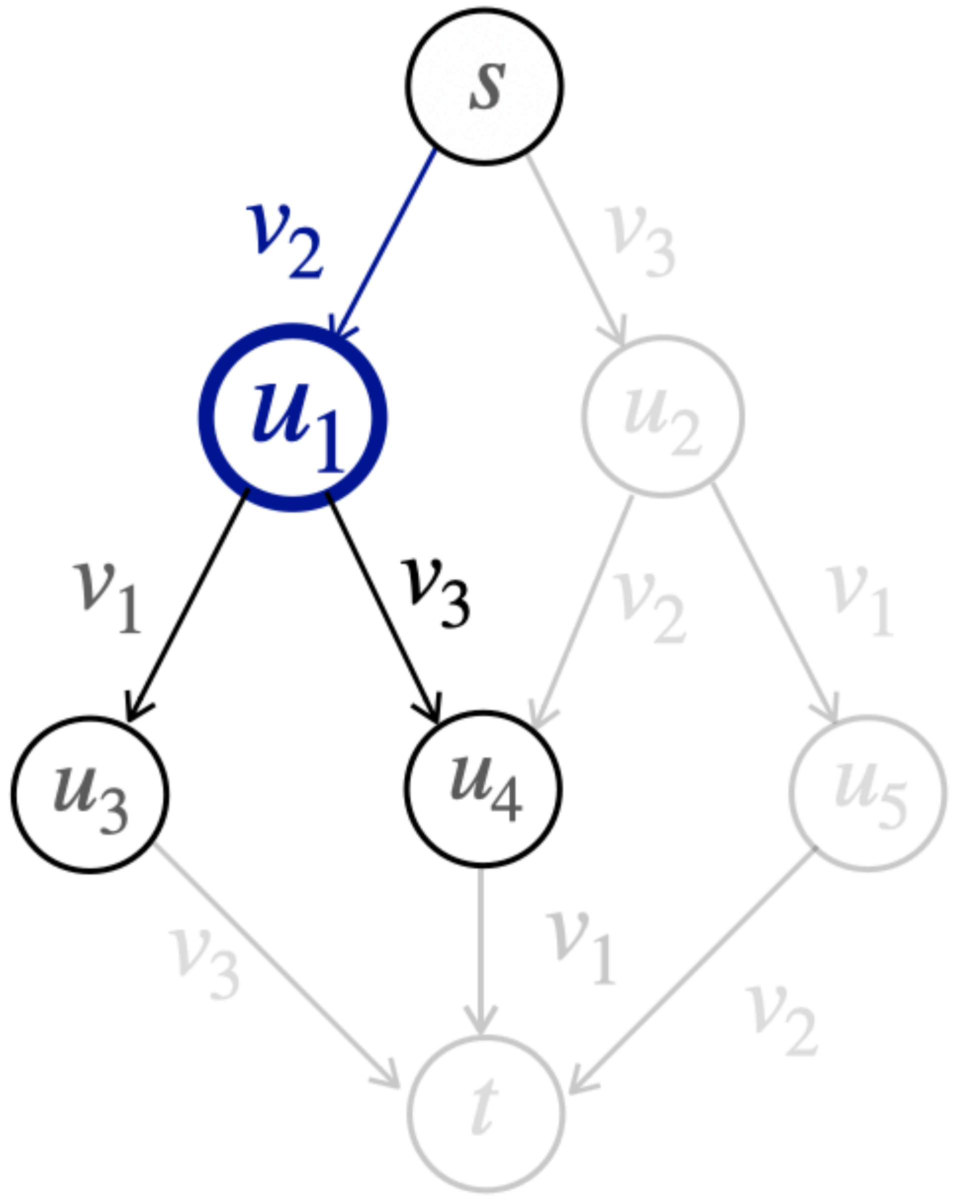




1st step output ⟺ Pick an edge $e(s, u_1) = v_2$

Execution step 2

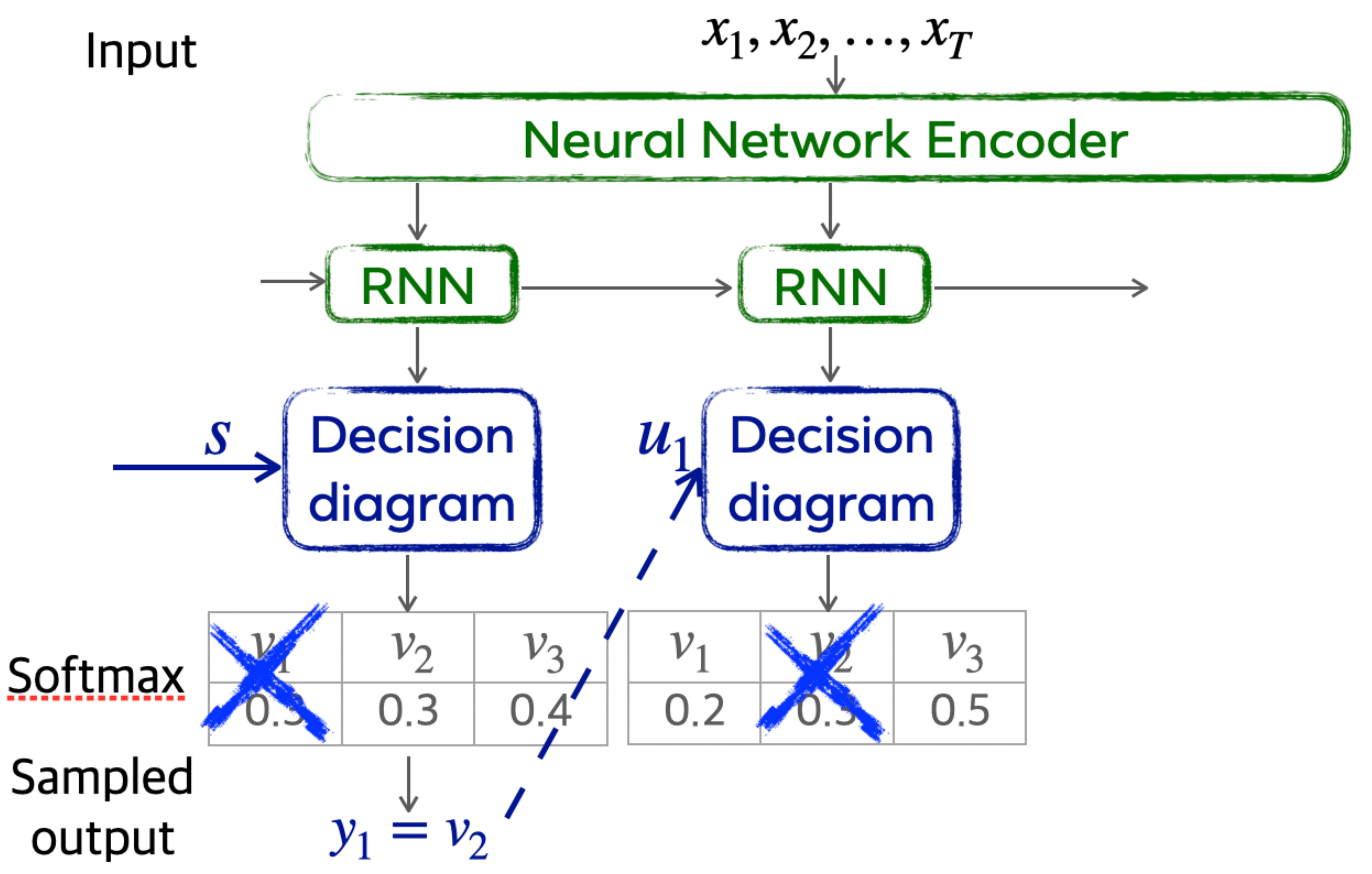


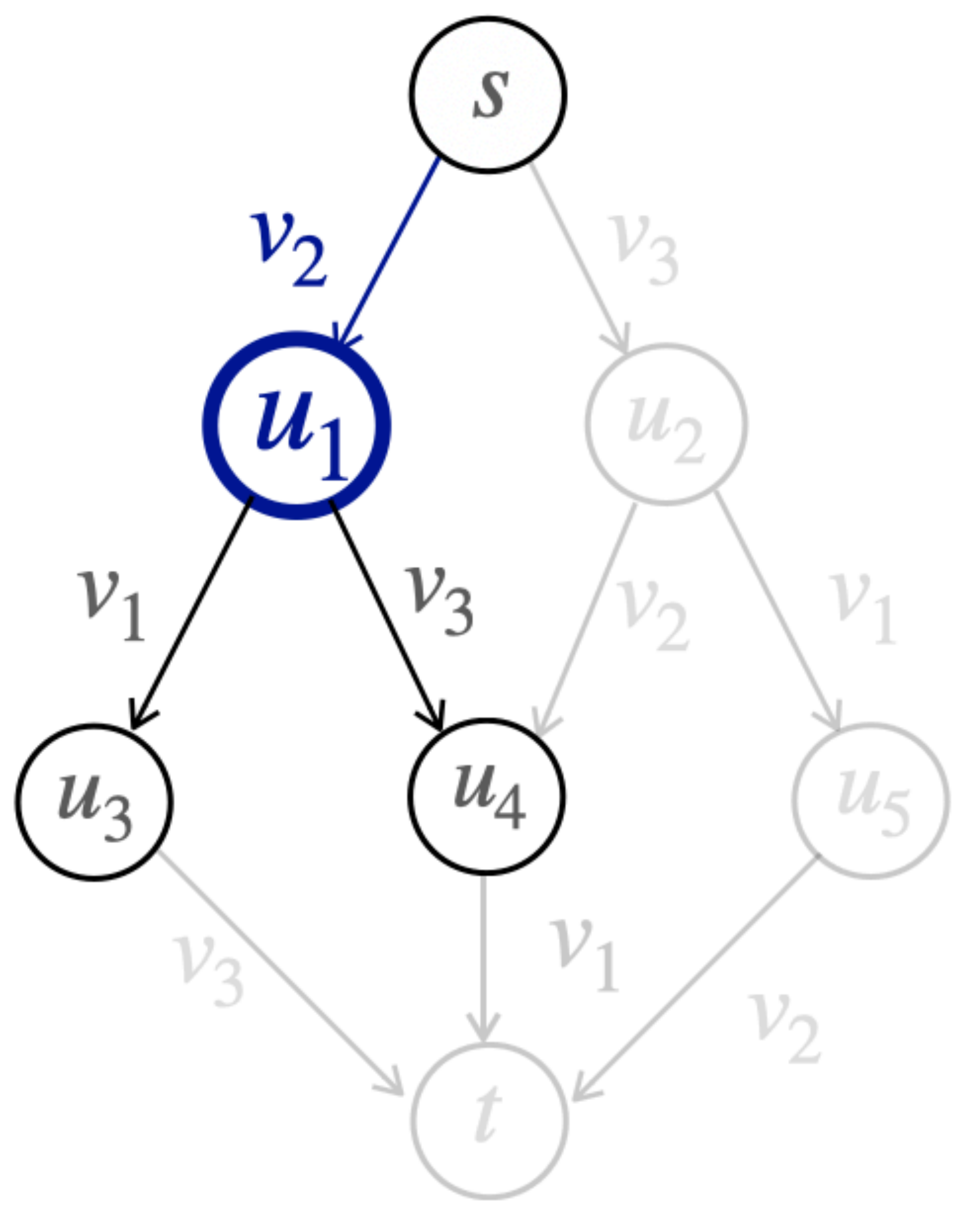


At node u_1

Nan Jiang et al., JMLR, 2022.

Execution step 2

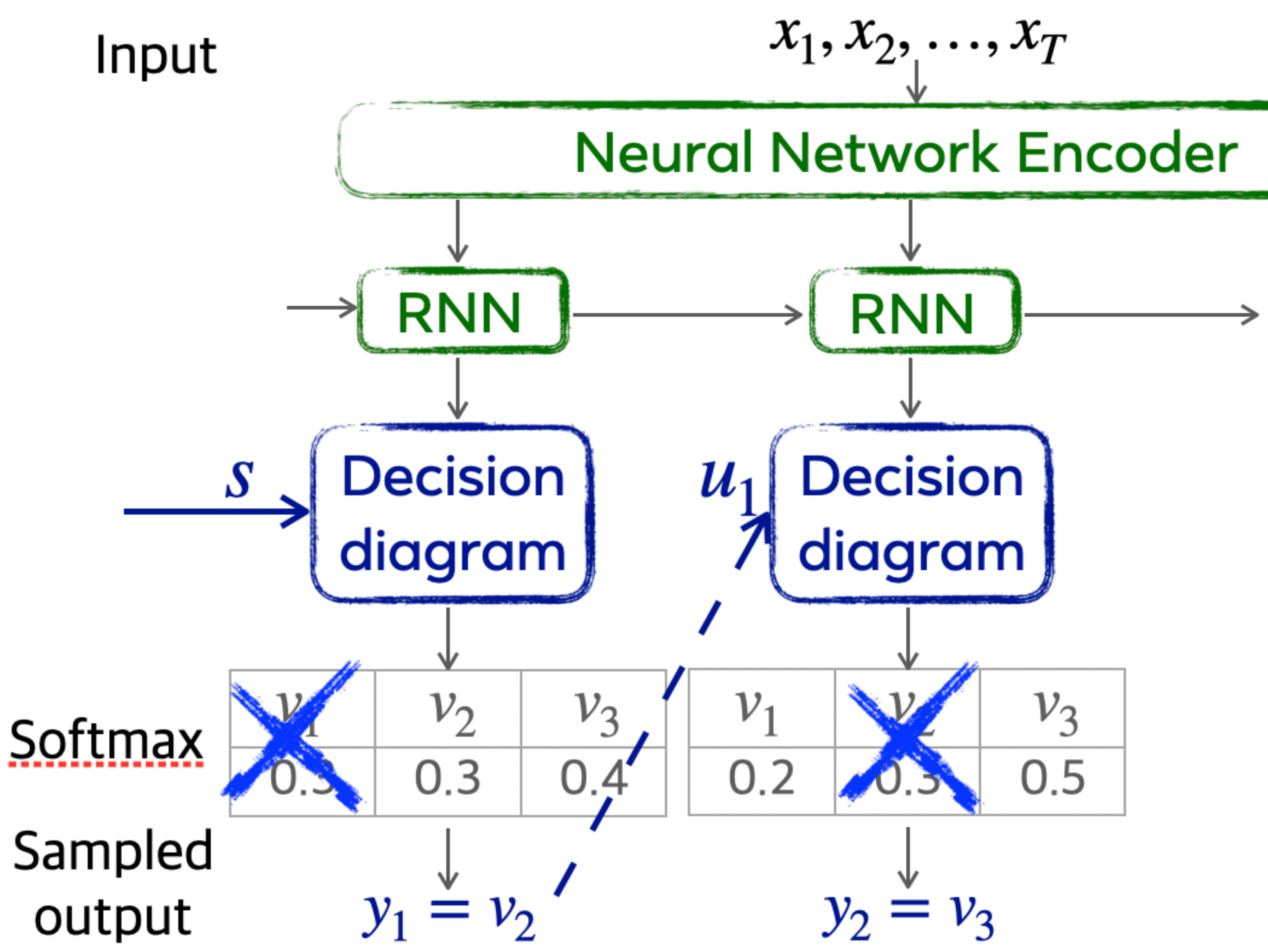


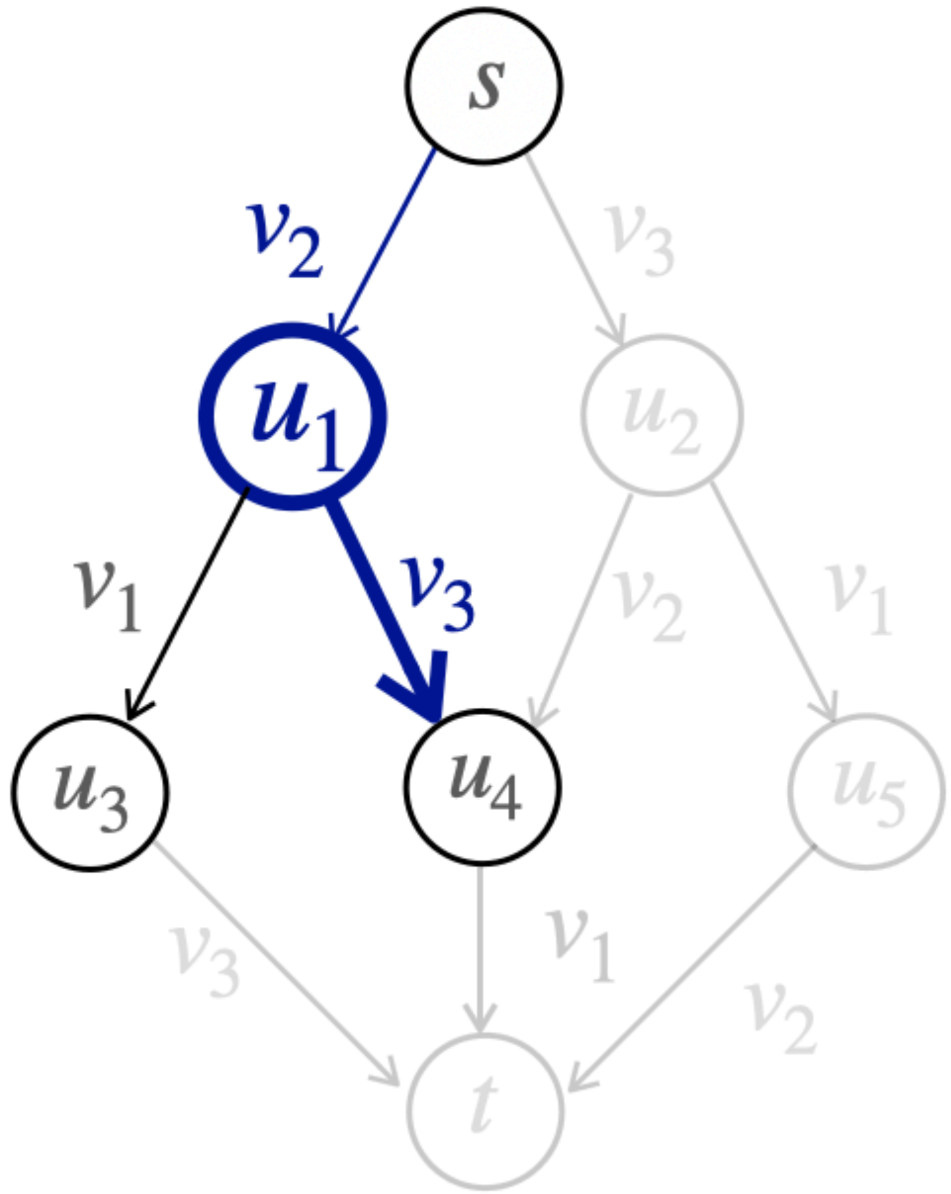


At node u_1

Nan Jiang et al., JMLR, 2022.

Execution step 2

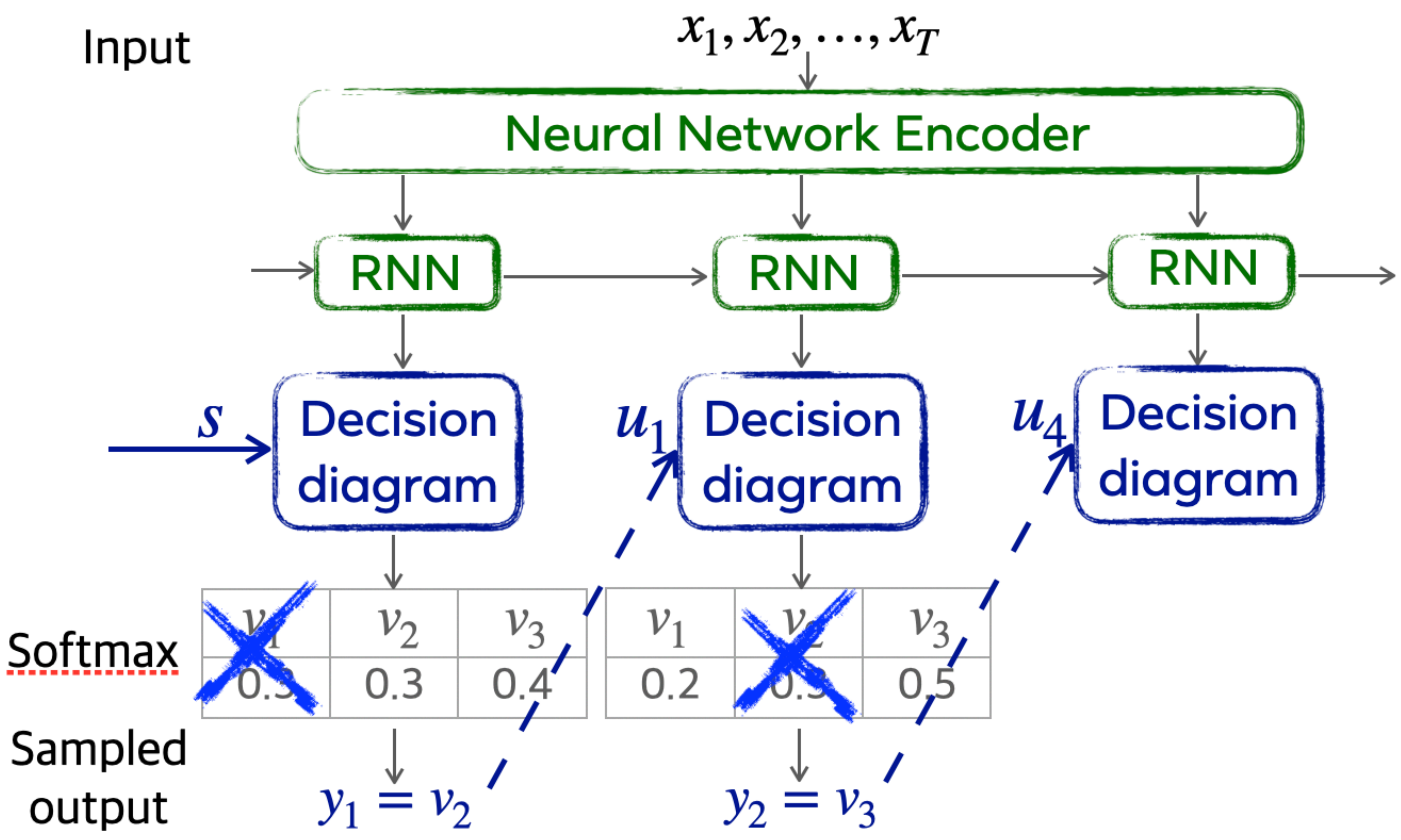


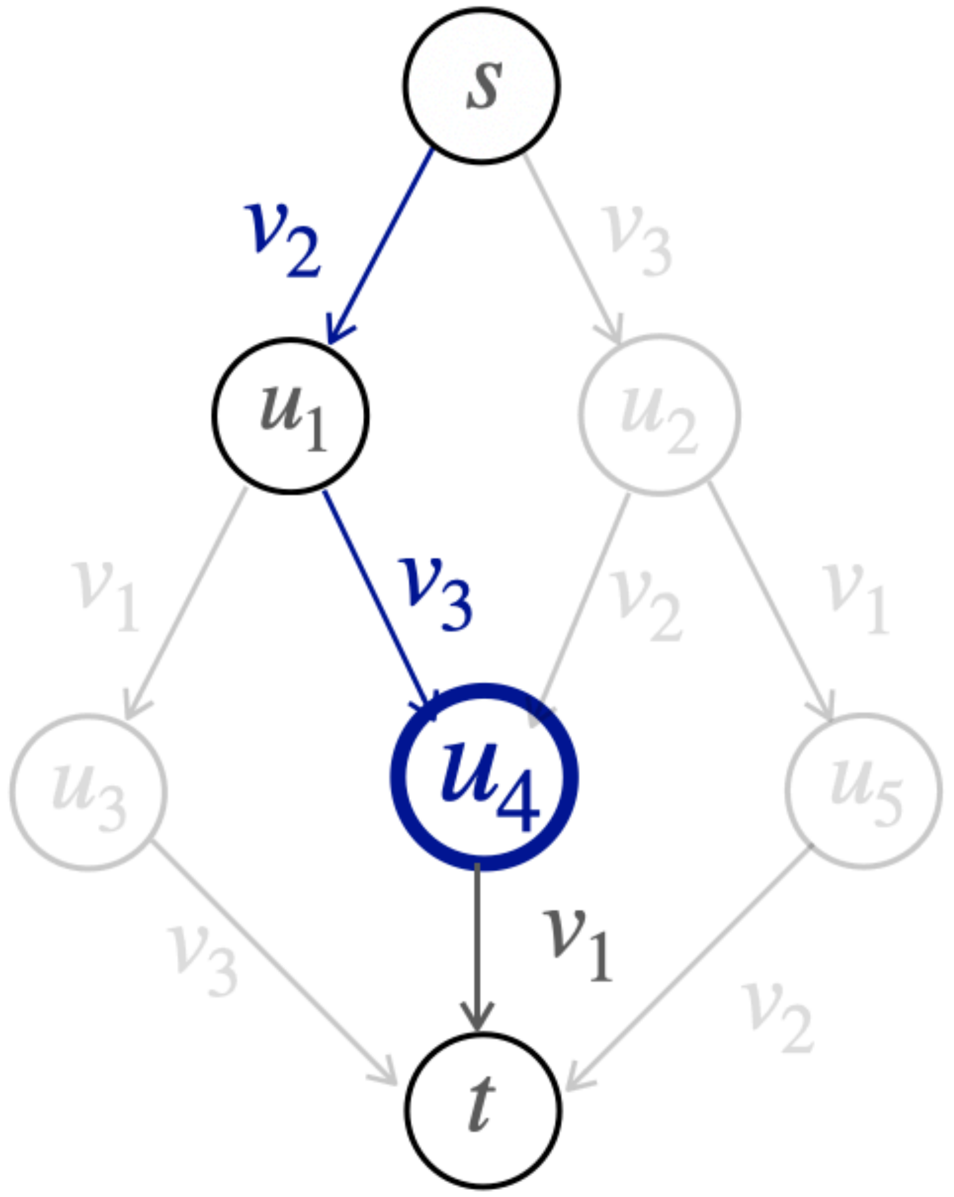


2nd step output $y_2 = v_3$ ⟺ Pick an edge $e(u_1, u_4) = v_3$

Nan Jiang et al., JMLR, 2022.

Execution step 3

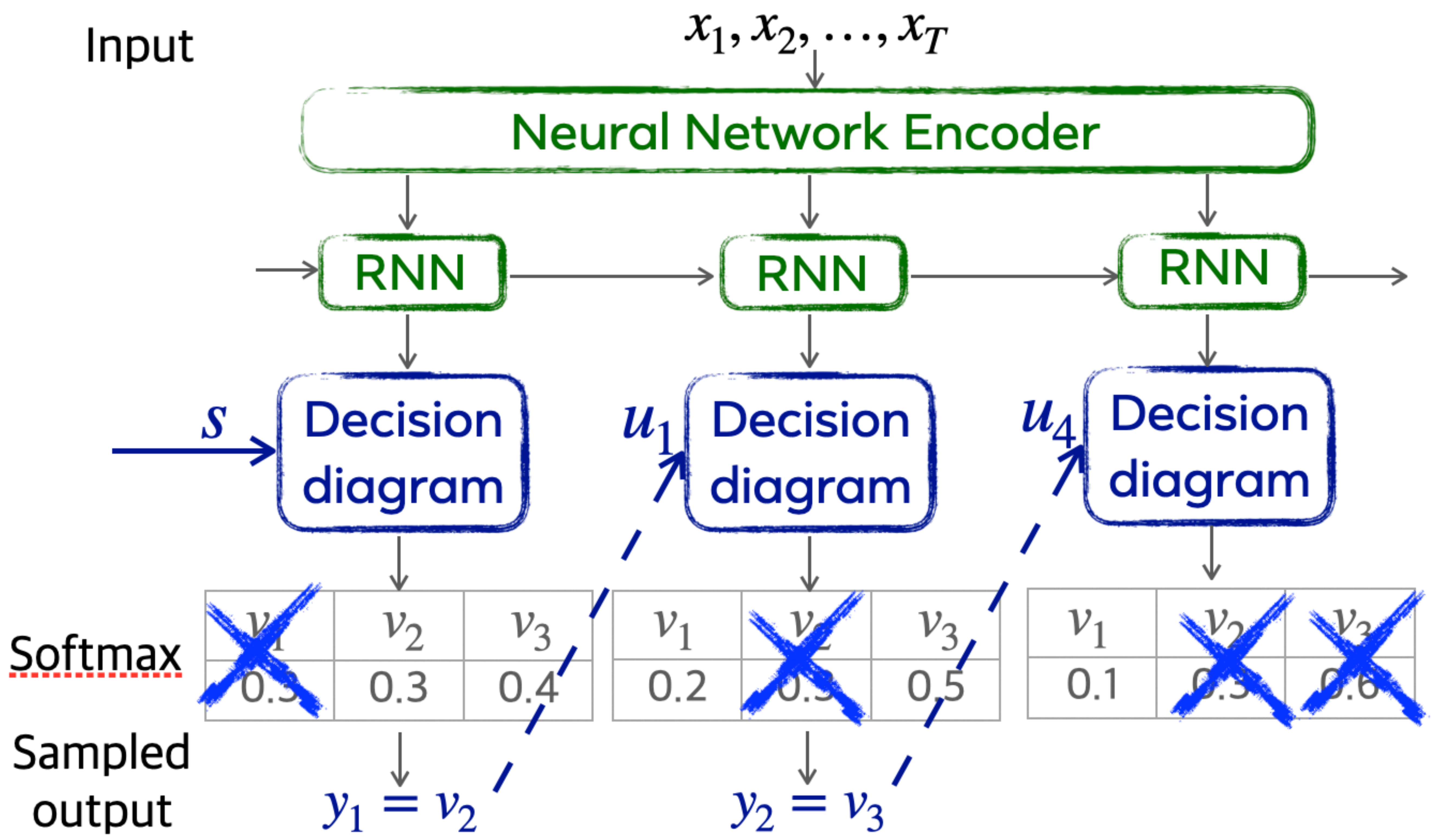


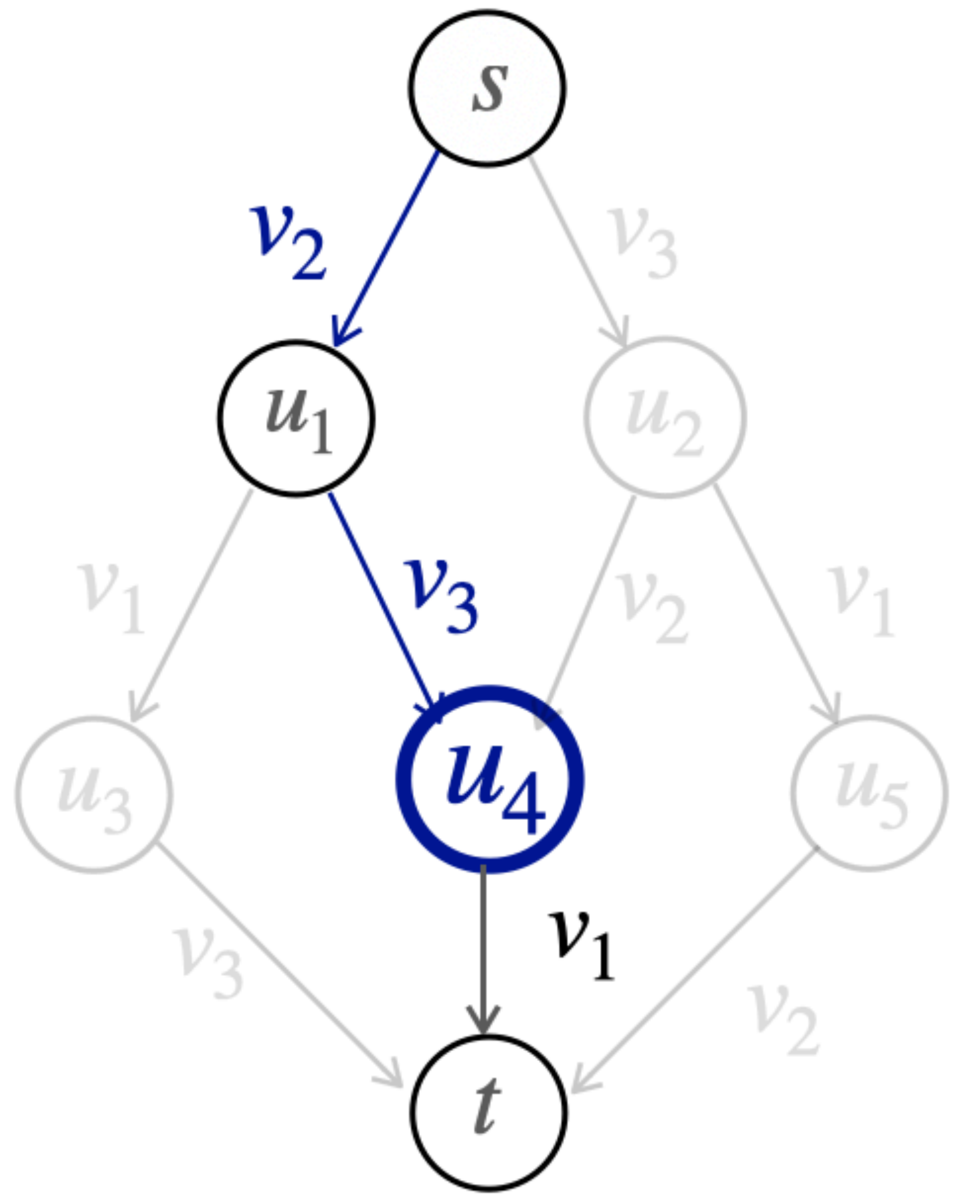


At node u_4

Nan Jiang et al., JMLR, 2022.

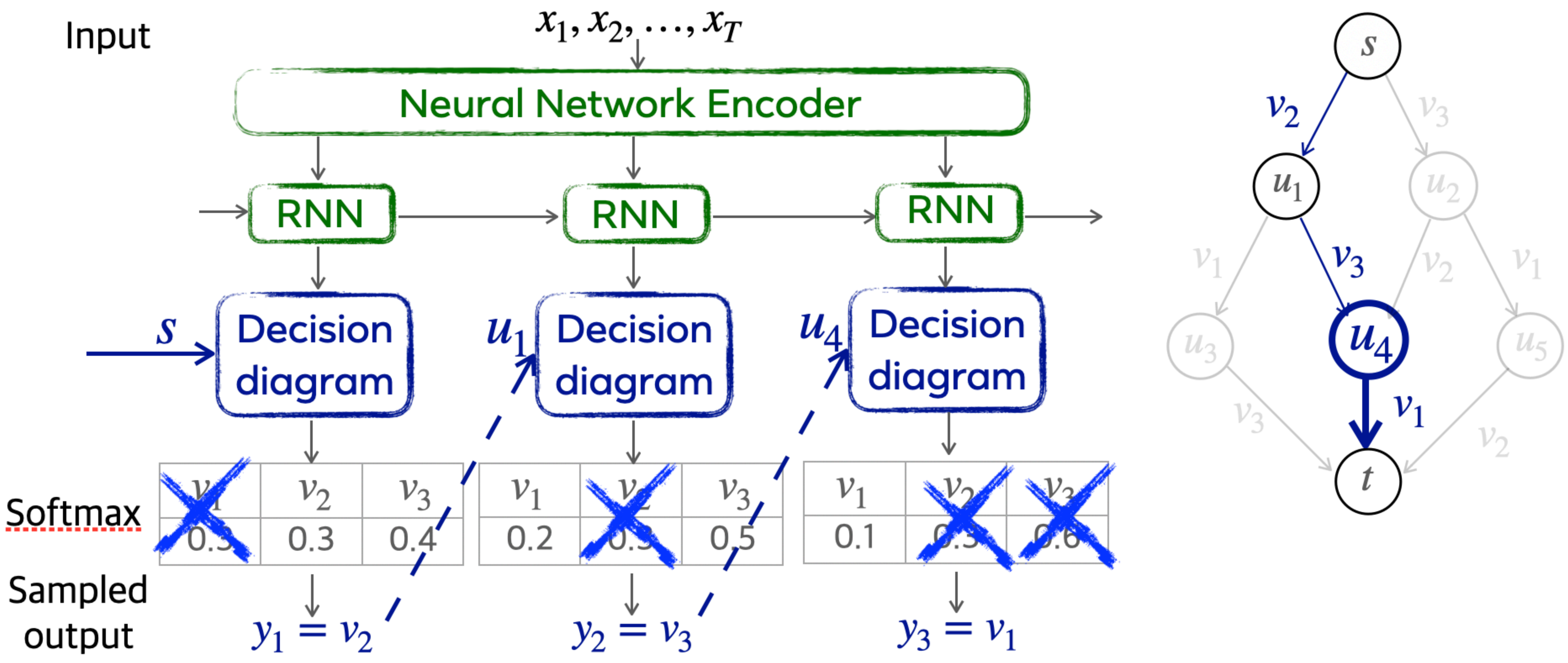
Execution step 3





At node u_4

Execution step 3



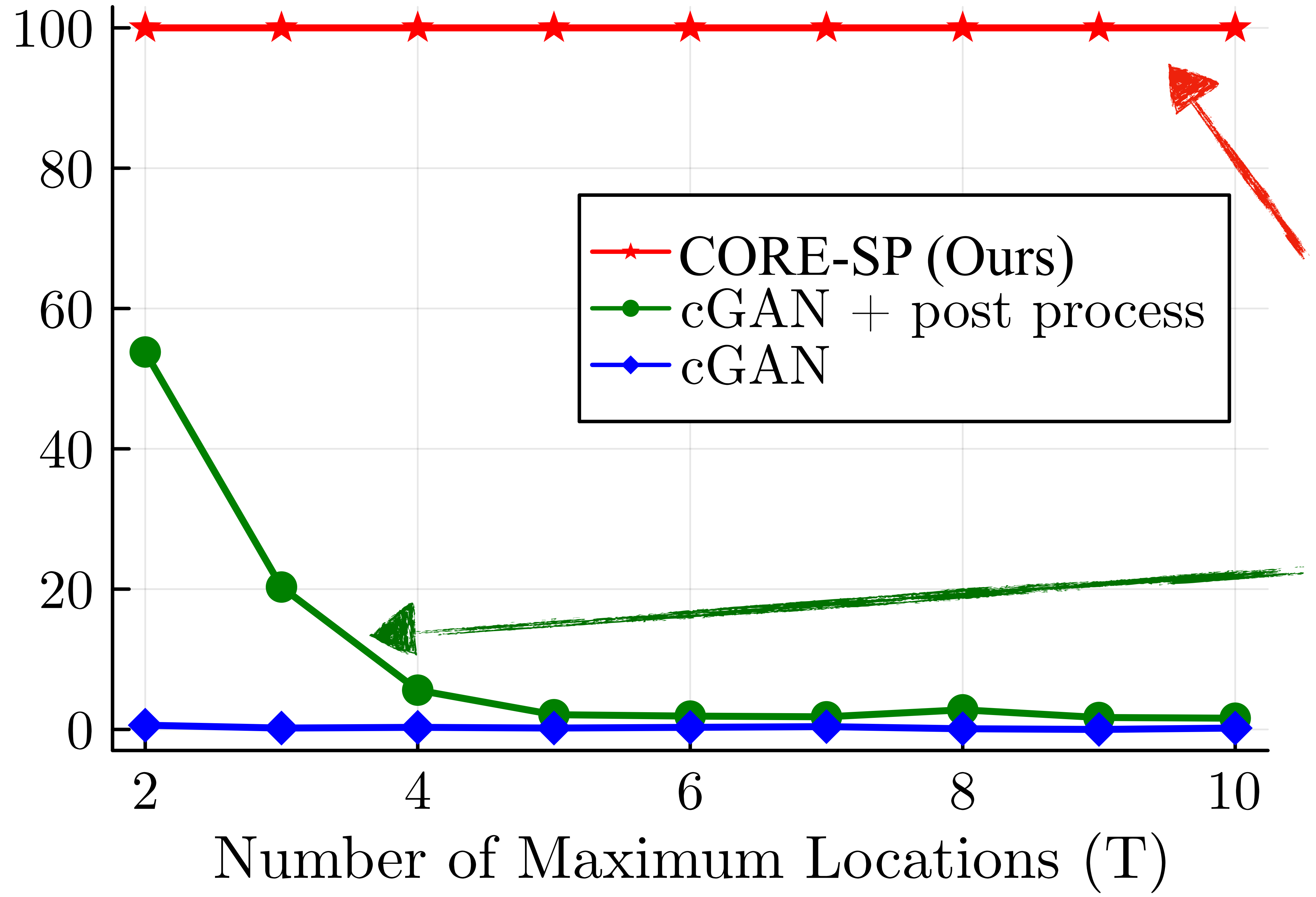
3rd step output $y_3 = v_1$ ⟺ Pick an edge $e(u_4, t) = v_1$

Nan Jiang et al., JMLR, 2022.

Experimental evaluations

Experiments: Delivery Route Planning

Valid Route (%)



Task: Recommend routes that

- satisfy delivery requests;
- meet agent' implicit preferences.

The outputs generated from integrated method satisfy 100% of the constraints.

The outputs generated from pure neural networks scale poorly to problem size.

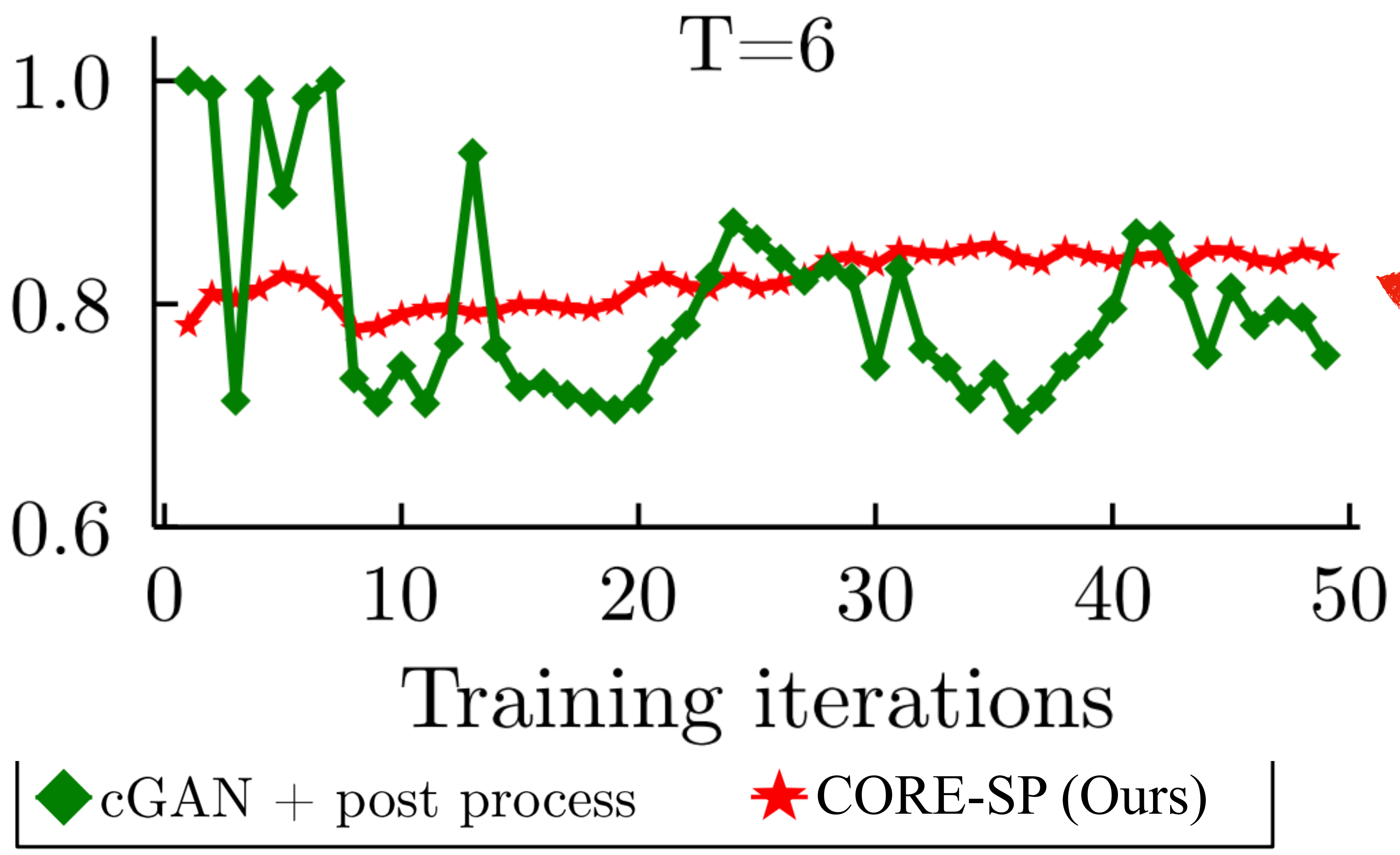
Nan Jiang et al., JMLR, 2022.

Experiments: Delivery Route Planning

Task: Recommend routes that

- satisfy delivery requests;
- meet agent' implicit preferences.

Reward-based Objective

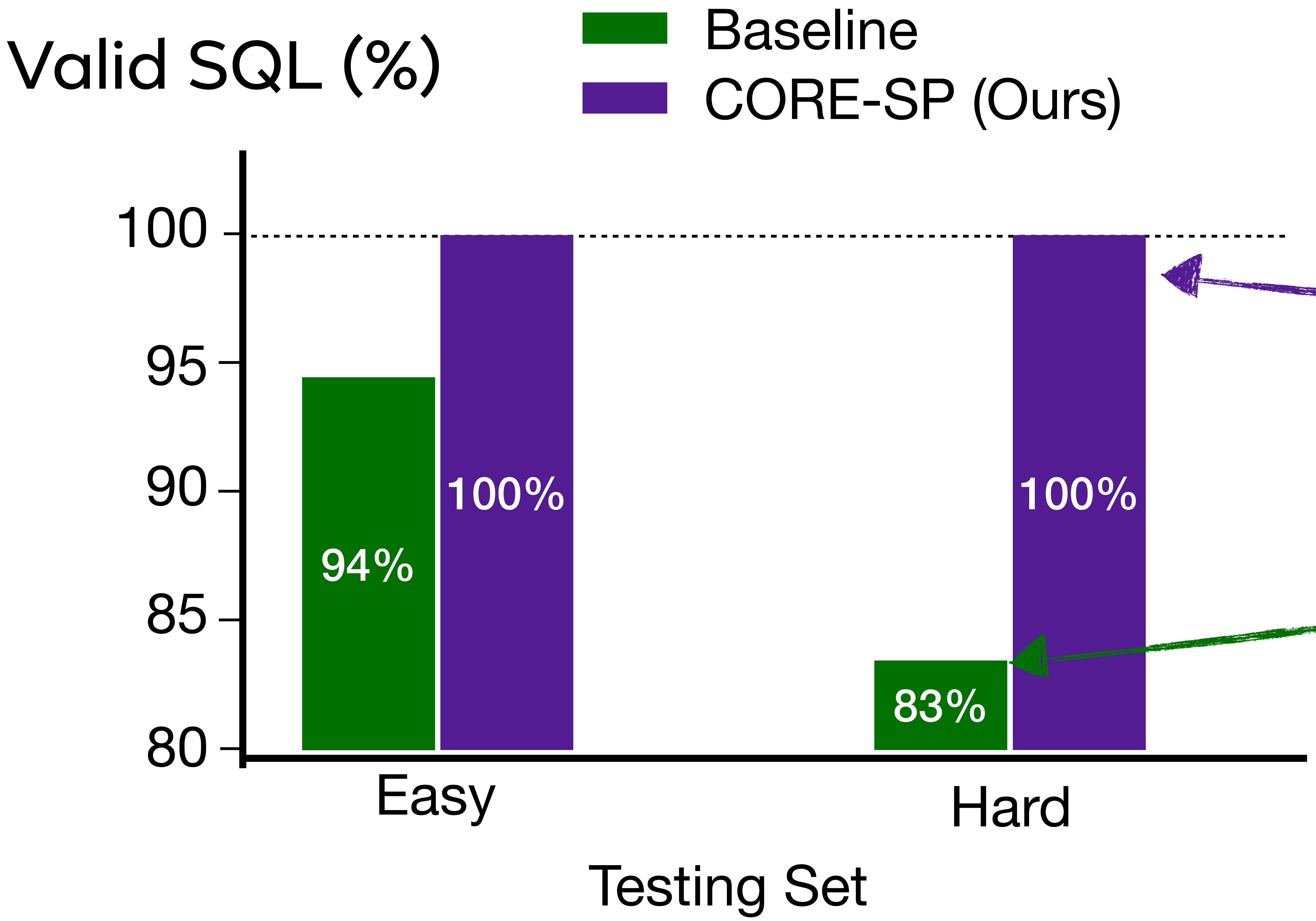


The training objective of our CORE-SP is more stable.

Nan Jiang et al., JMLR, 2022.

Experiments: code generation from language

Task: predict a SQL program that
- Understand user query in natural language.
- The program is executable.



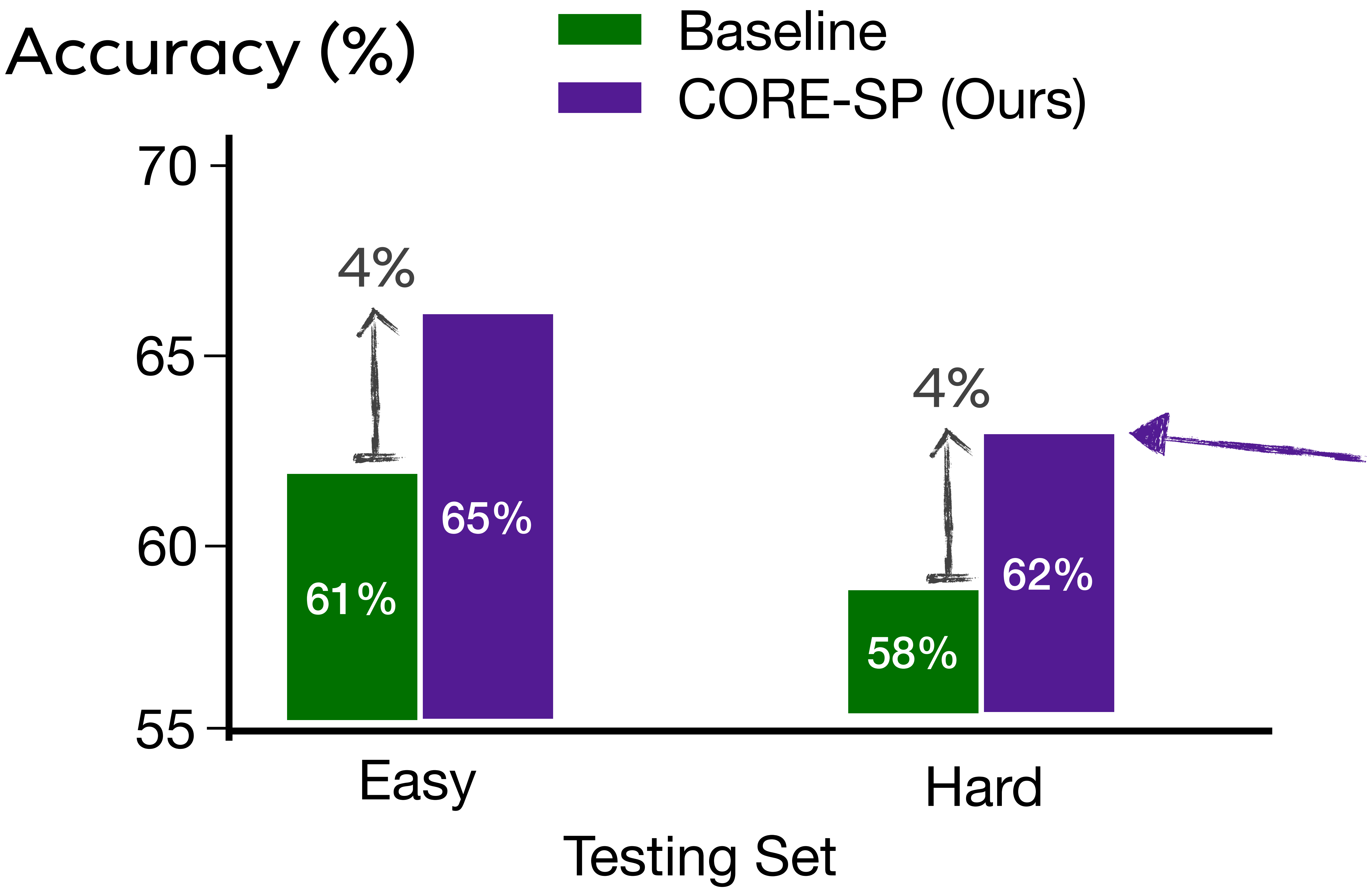
Our CORE-SP output SQL satisfying 100% of the constraints.

The baseline predict more invalid SQL when evaluated on harder dataset.

Nan Jiang et al., JMLR, 2022.

Experiments: code generation from language

Task: predict a SQL program that
- Understand user query in natural language.
- The program is executable.

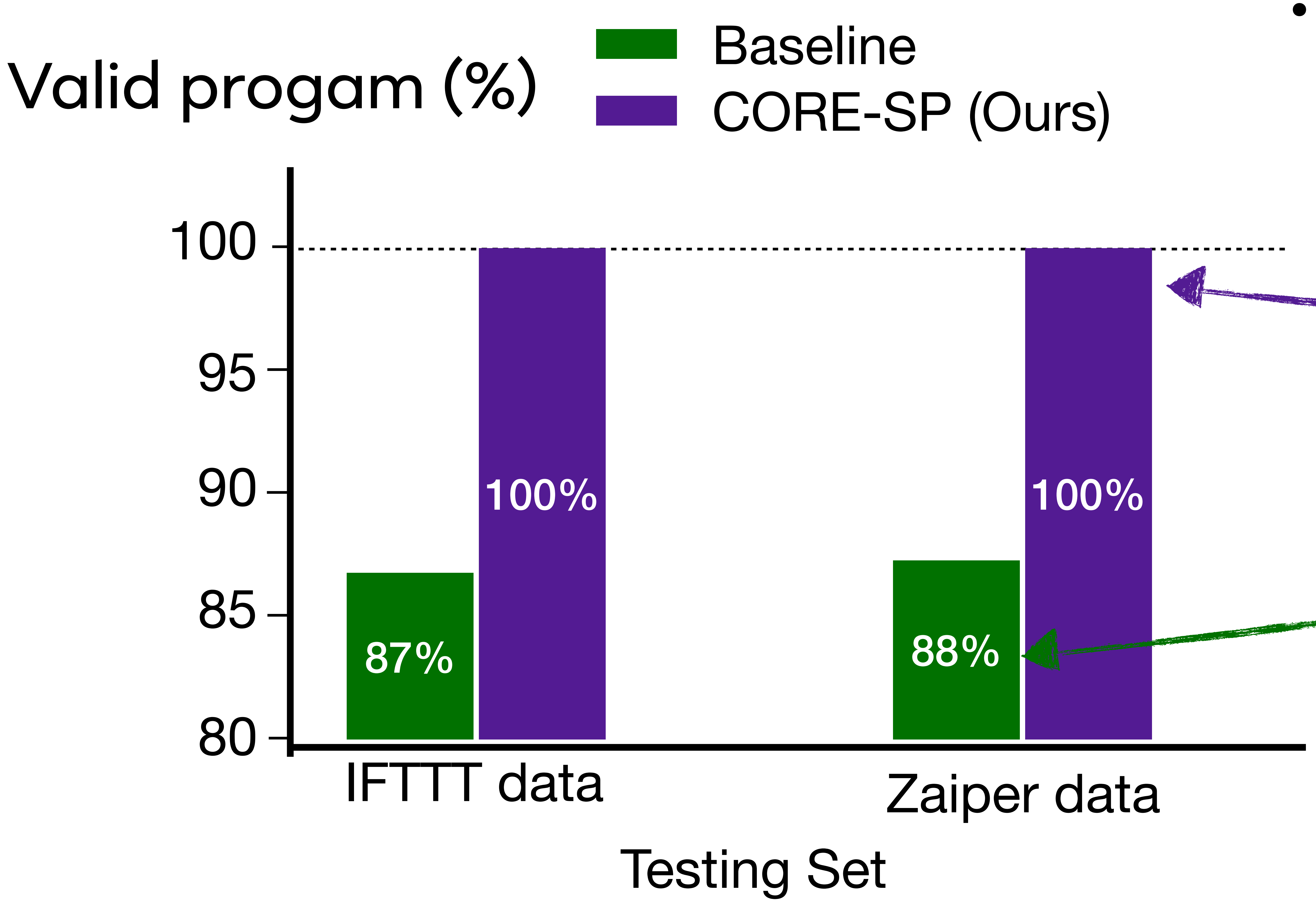


Model with reasoning attains a higher accuracy than the model without.

Nan Jiang et al., JMLR, 2022.

Experiments: code generation from language

Task: predict a web-service program that
- Understand user query in natural language.
- The program is executable.



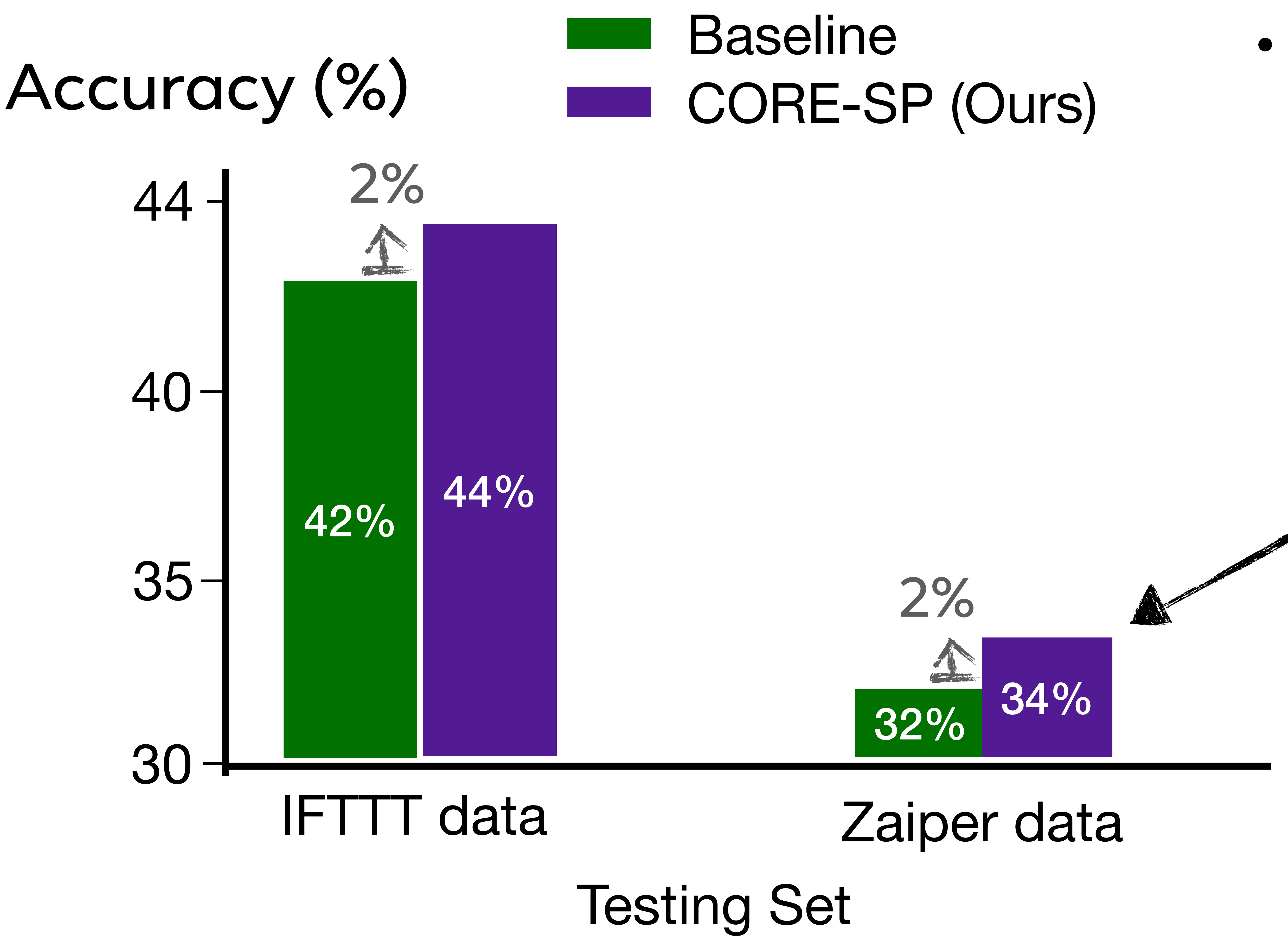
The structures generated from our CORE-SP satisfy 100% of the constraints.

~12% of the predictions from the baseline violate the constraints.

Nan Jiang et al., JMLR, 2022.

Experiments: code generation from language

Task: predict a web-service program that
- Understand user query in natural language.
- The program is executable.



Model with reasoning attains a higher accuracy than the model without.

Nan Jiang et al., JMLR, 2022.

Design principle of the integrated system
For logical constraints satisfying "extreme conditions"

Input x

Neural Network Models

A highly-efficient theory-guided sampler

Enforce constraints via Lovasz Local Lemma Theory

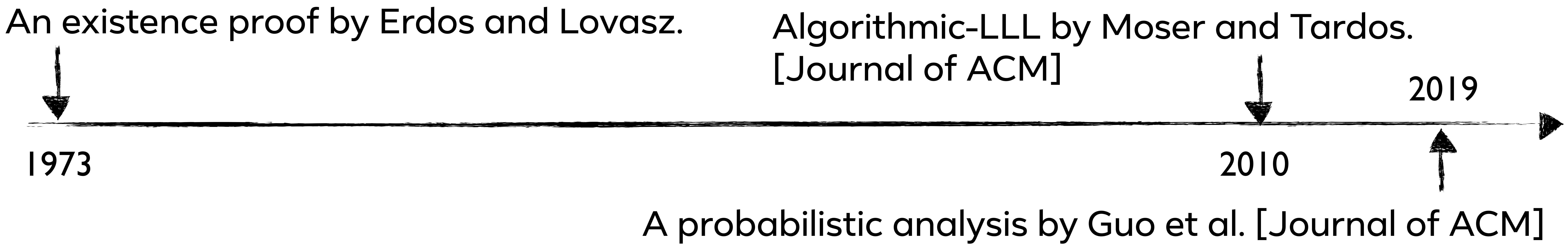
CNF-SAT Logical constraint, i.e.,

$$C = \overbrace{(x_1 \lor x_2)}^{c_1} \land \overbrace{(\neg x_1 \lor x_3)}^{c_2}$$

Feasible output y

Nan Jiang et al., Learning Markov Random Fields for Combinatorial Structures via Sampling through Lovász Local Lemma. AAAI, 2023.

The Background on Lovasz Local Lemma



Given:

- Boolean variables $X = (x_1, x_2, \ldots, x_n)$, with $x_i \in \{0,1\}$.
- CNF-SAT logical constraints, i,e., $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$.

Output:

A valid sample from distribution $P(X_1)\ldots P(X_n)$ subject to constraints C.

1. Transform into matrix computation.
2. embed into neural network.

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

c_2 is violated

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

Resample X_1, X_3 from $P(X_1), P(X_3)$ →

X_1	X_2	X_3
1	0	0
0	0	1

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^n$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

X_1	X_2	X_3
1	0	0
0	0	1

c_1 is violated

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

X_1	X_2	X_3
1	0	0
0	0	1
0	1	1

Resample X_1, X_2 from $P(X_1), P(X_2)$ $\rightarrow$

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = \{X_i\}_{i=1}^{n}$, with $X_i \in \{0,1\}$.

Marginal distribution: $P(X_1), P(X_2), P(X_3)$;

Constraints: $C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$

Output: A valid sample from distribution $P(X_1)P(X_2)P(X_3)$ subject to constraints C.

All constraints are satisfied!

X_1	X_2	X_3
1	0	0
0	0	1
0	1	1

Our contribution: we formulate a fully-differentiable and efficient neural network modules that simulates sampling through Lovasz Local Lemma.

Design principle of the integrated system
For logical constraints satisfying "extreme conditions"

Input x_1, x_2, x_3

Markov random field

Neural Network Models

Lovasz Local Lemma-based Sampler

constraint reasoning sampler

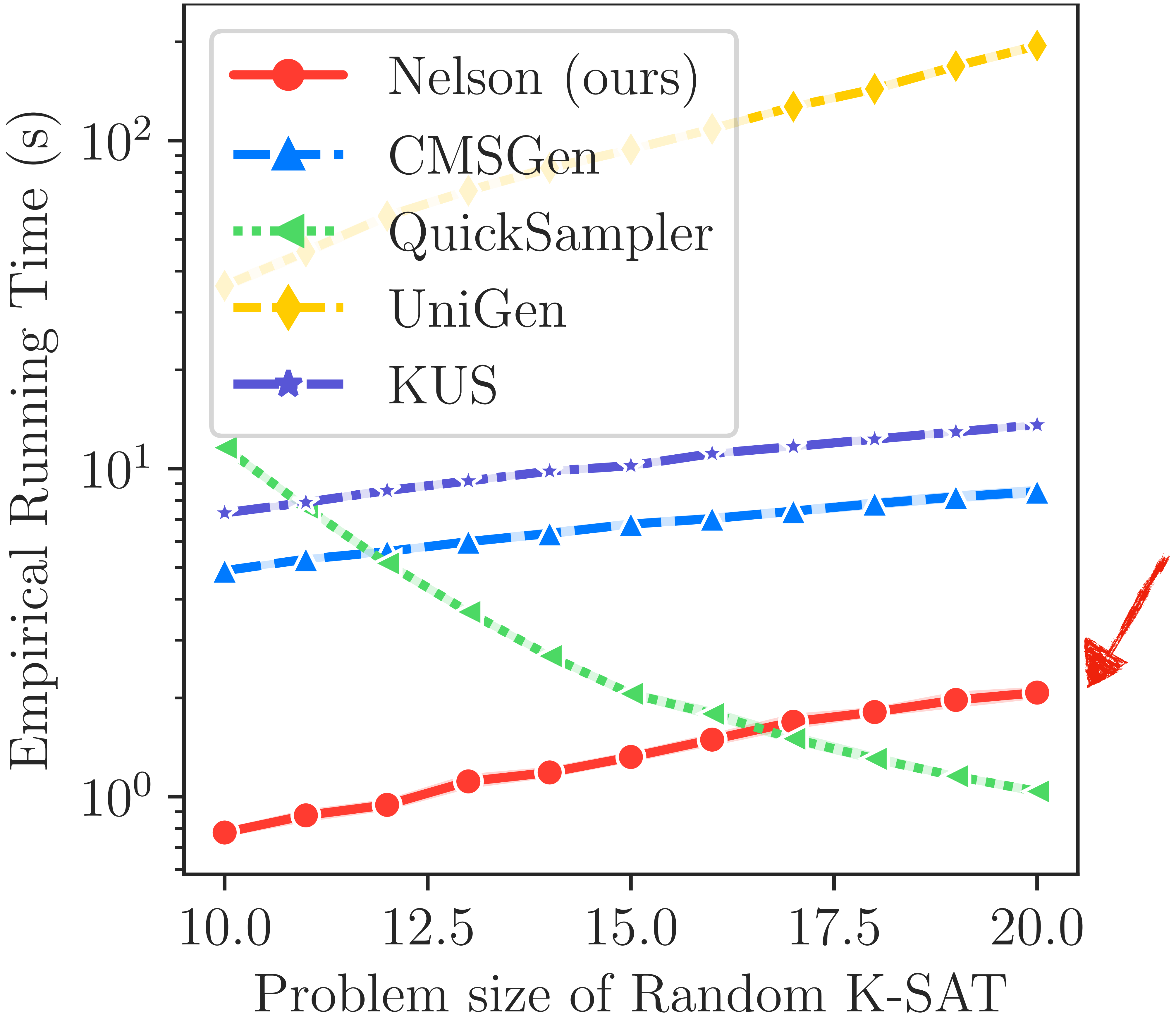
Logical constraint, i.e.,

$$C = \overbrace{(x_1 \vee x_2)}^{c_1} \wedge \overbrace{(\neg x_1 \vee x_3)}^{c_2}$$

Output: feasible variable assignment

$x_1 = \text{True}, x_2 = \text{False}, x_3 = \text{True}$

Experiments: Random K-SAT Solutions with Implicit Preference



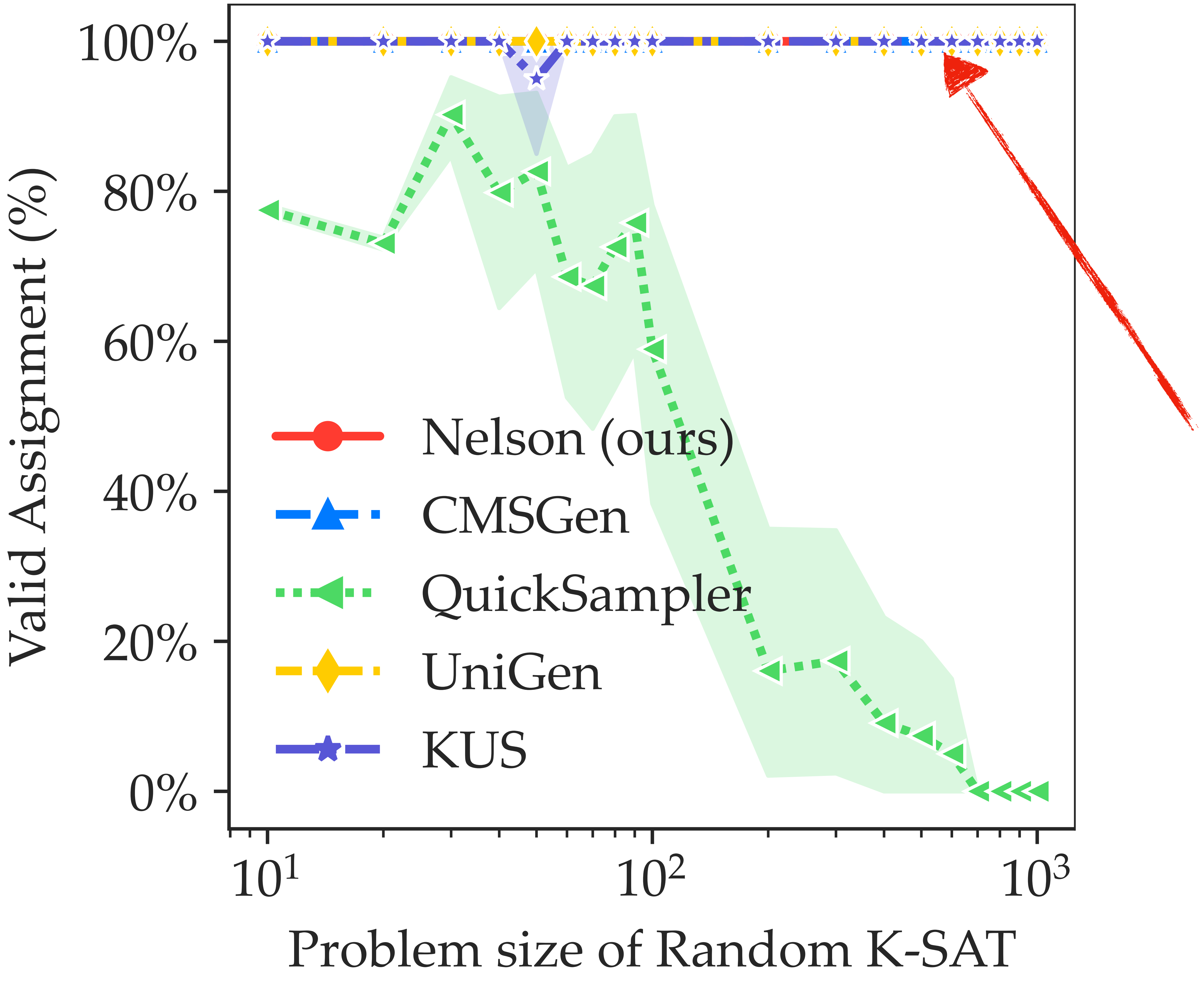
Task: sample feasible output from the model.

Our method is much faster than existing methods.

Nan Jiang et al., AAAI, 2023.

Experiments: Random K-SAT Solutions with Implicit Preference

Task: sample feasible output from the model.



Our method always sample feasible output from the model.

Nan Jiang et al., AAAI, 2023.

Design principle of the integrated system:
For a mixture of binary- and real-valued constraints

Input x

Neural Network Models

GPT-3

Suited for Large Language Model, i.e., GPT model.

Enforce constraints via pruned tree search

The constraints can be binary-valued or real-valued.

Feasible output y

Maosen Zhang, **Nan Jiang**, et al. Language Generation via Combinatorial Constraint Satisfaction: A Tree Search Enhanced Monte-Carlo Approach. EMNLP 2020.

(3) Controllable Text Generation with Constrained Tree Search

"NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics"
Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah Smith, Yejin Choi

Best new method paper

Notes from the Best Paper Committee: Language generation is, in its simplest form, a search problem in very high dimensional space. This paper makes that connection clear by incorporating the classic search algorithm A* into the language generation process. A* allows for a heuristic search that incorporates "lookahead" signals of future performance into token selection. The authors perform a very thorough evaluation of their model across many tasks including question generation, machine translation, and story generation. They show large performance improvements over the typical beam search approach, and over their original NeuroLogic algorithm. This paper is an inspiring mixture of old and new.

Our method

Decode Method	Automatic Evaluation						Human Evaluation			
	ROUGE	BLEU	METEOR	CIDEr	SPICE	Coverage	Grammar	Fluency	Meaningfulness	Overall
CGMH (Miao et al., 2019)	28.8	2.0	18.0	5.5	21.5	18.3	2.28	2.34	2.11	2.02
TSMH (Zhang et al., 2020)	42.0	4.3	25.9	10.4	37.7	92.7	2.35	2.28	2.37	2.22
NeuroLogic (Lu et al., 2021)	38.8	11.2	24.5	18.0	41.7	90.6	2.78	2.71	2.49	2.51
NeuroLogic★ (greedy)	**43.7**	**14.7**	28.0	**20.9**	47.7	**100.0**	**2.83**	2.77	2.74	**2.76**
NeuroLogic★ (beam)	42.9	14.4	27.8	20.3	46.9	**100.0**	2.81	**2.86**	**2.76**	2.75
NeuroLogic★ (sample)	43.5	14.6	**28.2**	20.8	**47.8**	**100.0**	**2.83**	2.75	**2.76**	2.73

Table 8: Performance of different unsupervised decoding algorithms on interrogative question generation.

Outline

1 Formal guarantee: Integrate reasoning with learning to ensure constraint satisfaction for structured prediction.

2 Scalability: Integrate reasoning with learning to accelerate scientific discovery.

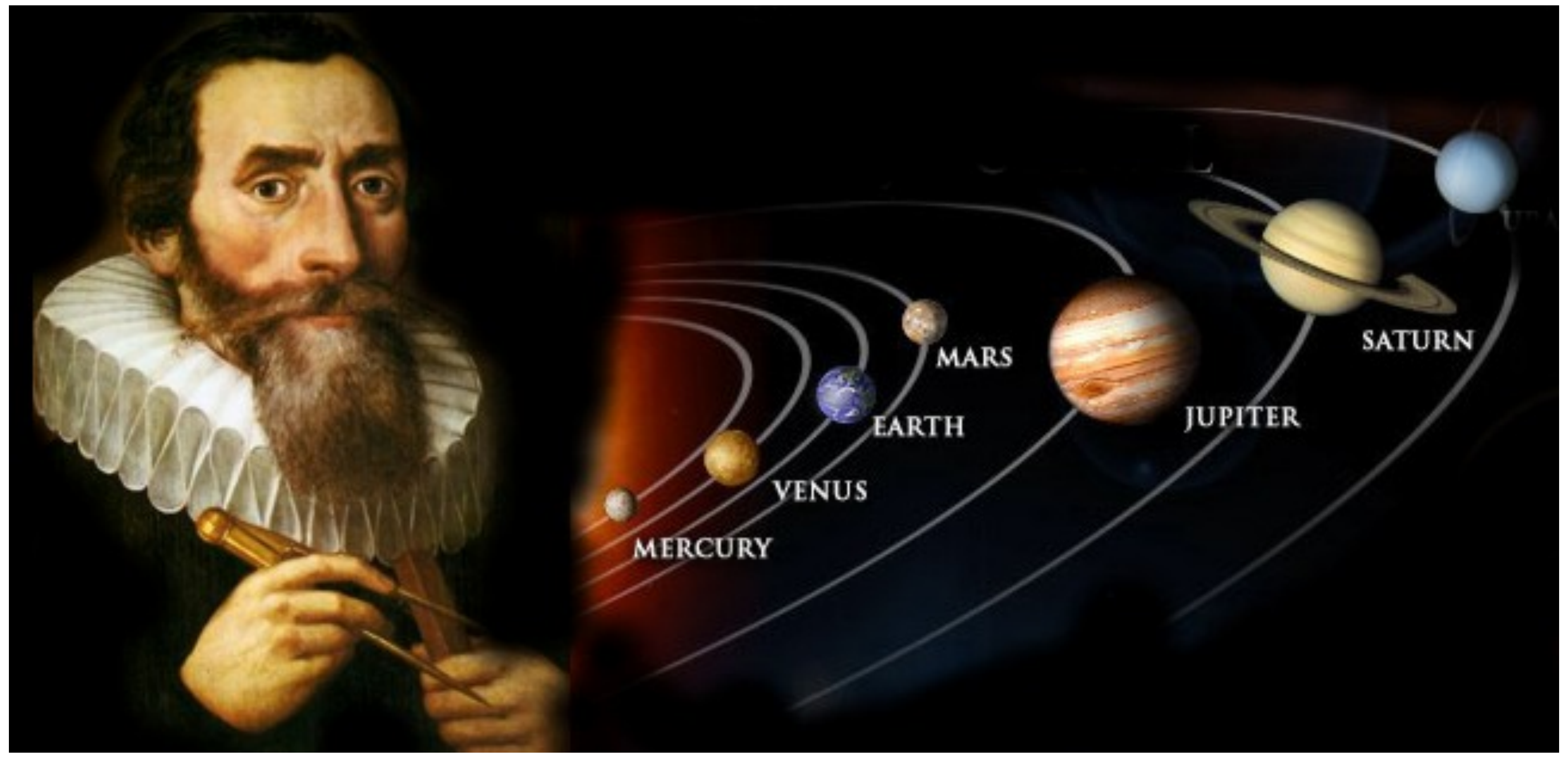
Nan Jiang et al. AAAI 2025; **Nan Jiang et al.** AAAI 2024; **Nan Jiang et al.** IJCAI 2024. **Nan Jiang et al.** RLJ, 2024; **Nan Jiang et al.,** ECML, 2022; **Nan Jiang et al.,** WWW 2022.

3 Future work

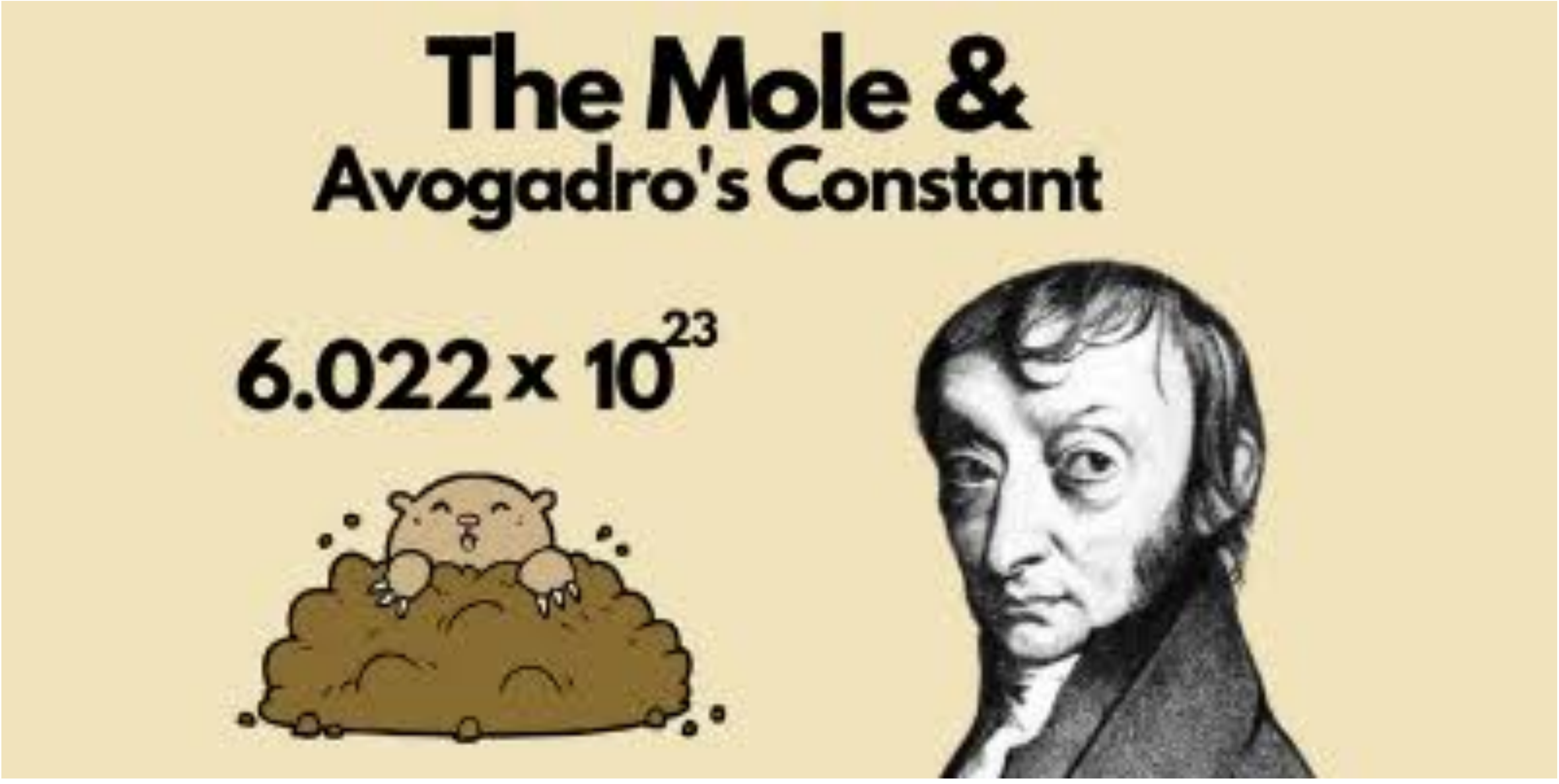
Symbolic Regression: An Important Task in Scientific Discovery

Goal: discover new physical knowledge from data.

Scientist-based discovery is slow.



Kepler discovered laws of planetary motion



Avogadro found the idea gas law

Machine is much faster!

Symbolic regression uses machine learning to advance the discovery of more complex physical phenomena.

Background on Symbolic Regression in Scientific Discovery

Goal: discover new physical knowledge from data.

Given:

- Experimental data
- a set of math operators, i.e.,$\{+, -, \times, \div, \sin, \exp\}$.

Goal:

Find a closed-form equation that best fits the data.

Experimental Data

x_1	x_2	x_3	x_4	y
0.2	0.4	0.2	0.7	-0.24
0.9	0.3	0.5	0.5	0.30
0.5	0.4	0.8	0.1	0.36
0.1	0.8	0.7	0.6	-0.41

Symbolic regression method

Best symbolic equation is

$x_1 \times x_2 - x_3/x_4$.

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Existing works on scientific discovery and current challenges on scalability

1979 — BACON (Pat Langley & Herbert Simon)

....

2009 — Generic programming [science 2009]

2019 — Deep Reinforcement Learning [2019]
Monte Carlo Tree Search [2020]

2024 — Pre-trained Transformer [2021]
Large-language models [2024]

Current challenges on scalability

- Struggle to solve equations with a few (≤ 4) variables.
- The space of expressions grows $\propto \exp(\#\text{input variables})$.

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

A running example of our idea

X_1	X_2	X_3	Y
2.5	1.0	9.5	12
3.0	-1.0	4.0	1
1.6	3.5	5.2	10.8
1.8	1.0	3.2	5
7.1	8.6	3.8	64.9
1.7	1.0	2.3	4
2.5	2.6	3.1	9.6
8.9	1.1	2.0	11.8
4.2	-1.0	2.2	-2
5.8	1.0	7.2	13
1.6	5.7	1.2	10.3
9.7	-1.0	1.7	-8

Can you guess which equation $y = f(x_1, x_2, x_3)$ generates the data shown in the left table?

A running example of our idea

X_1	X_2	X_3	Y
3.0	-1.0	4.0	1
4.2	-1.0	2.2	-2
9.7	-1.0	1.7	-8

How about if I only ask you to look into these rows?

It could be $y = x_1 + x_3$

A running example

X_1	X_2	X_3	Y
2.5	1.0	9.5	12
1.8	1.0	3.2	5
1.7	1.0	2.3	4
5.8	1.0	7.2	13

How about these rows?

It could be $y = -x_1 + x_3$

A running example

Red and blue data are from control variable experiments that X_2 is controlled.

X_1	X_2	X_3	Y
2.5	1.0	9.5	12
3.0	-1.0	4.0	1
1.8	1.0	3.2	5
1.7	1.0	2.3	4
4.2	-1.0	2.2	-2
5.8	1.0	7.2	13
9.7	-1.0	1.7	-8

Based on the discovered expressions:

$$y = x_1 + x_3$$

$$y = -x_1 + x_3$$

The true expression could be:

It could be $y = x_2 x_1 + x_3$

Idea: Inspired from idea gas law

- In 1663, Robert Boyle found:

$$PV = \texttt{const}$$

where n and T are fixed.

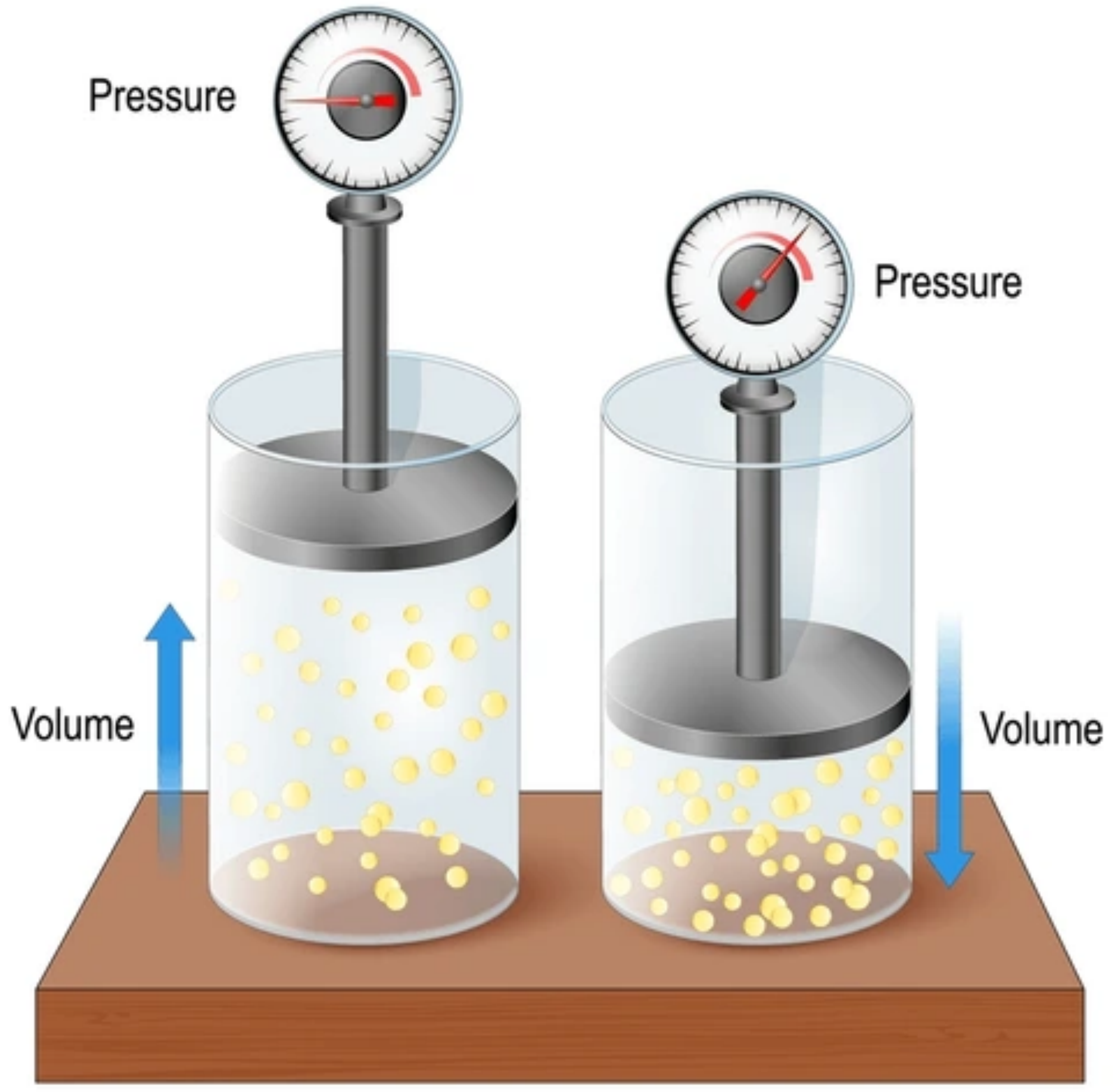
- In 1787, Jacques Charles demonstrated

$$\frac{PV}{T} = \text{const}$$

where only n is fixed.

- In 1811, Amedeo Avagadro demonstrated

$$\frac{PV}{nT} = \text{const}$$



Relevant variables:

- The amount of gas (n, moles),
- Temperature (T),
- Pressure (P),
- Volume of gas (V).

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Image source: https://www.energy.gov/

Idea: Inspired from idea gas law

- In 1663, Robert Boyle found:

$$PV = \texttt{const}$$

where n and T are fixed.

- In 1787, Jacques Charles demonstrated

$$\frac{PV}{T} = \text{const}$$

where only n is fixed.

- In 1811, Amedeo Avagadro demonstrated

$$\frac{PV}{nT} = \text{const}$$

We call this iterative process of doing control variable experiments as Scientific Reasoning:

Step 1. n and T are fixed.

Step 2. only n is fixed.

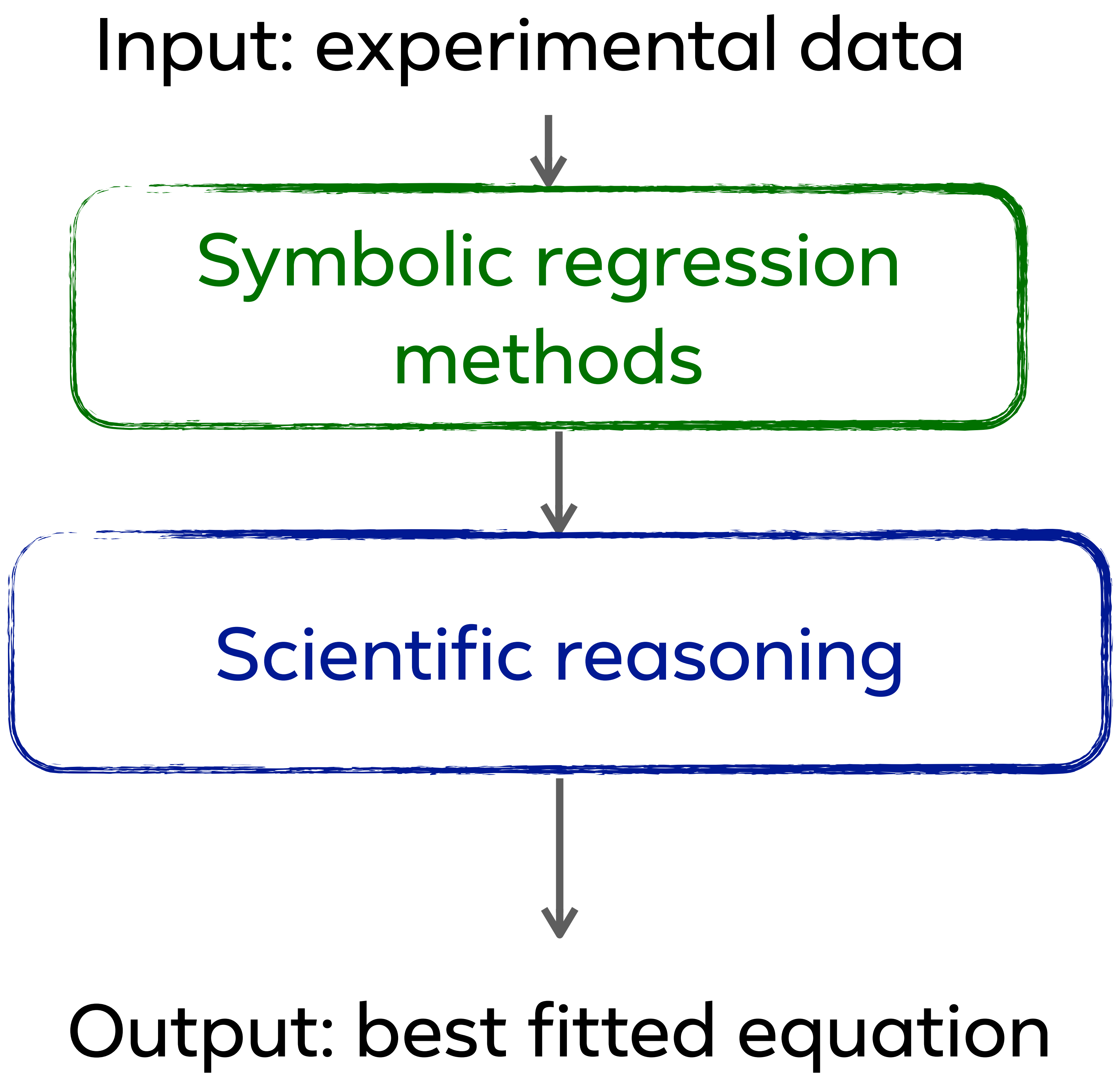
Step 3. No variable is fixed.

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Design principle of the integrated system

- Search for optimal equation that matches the data

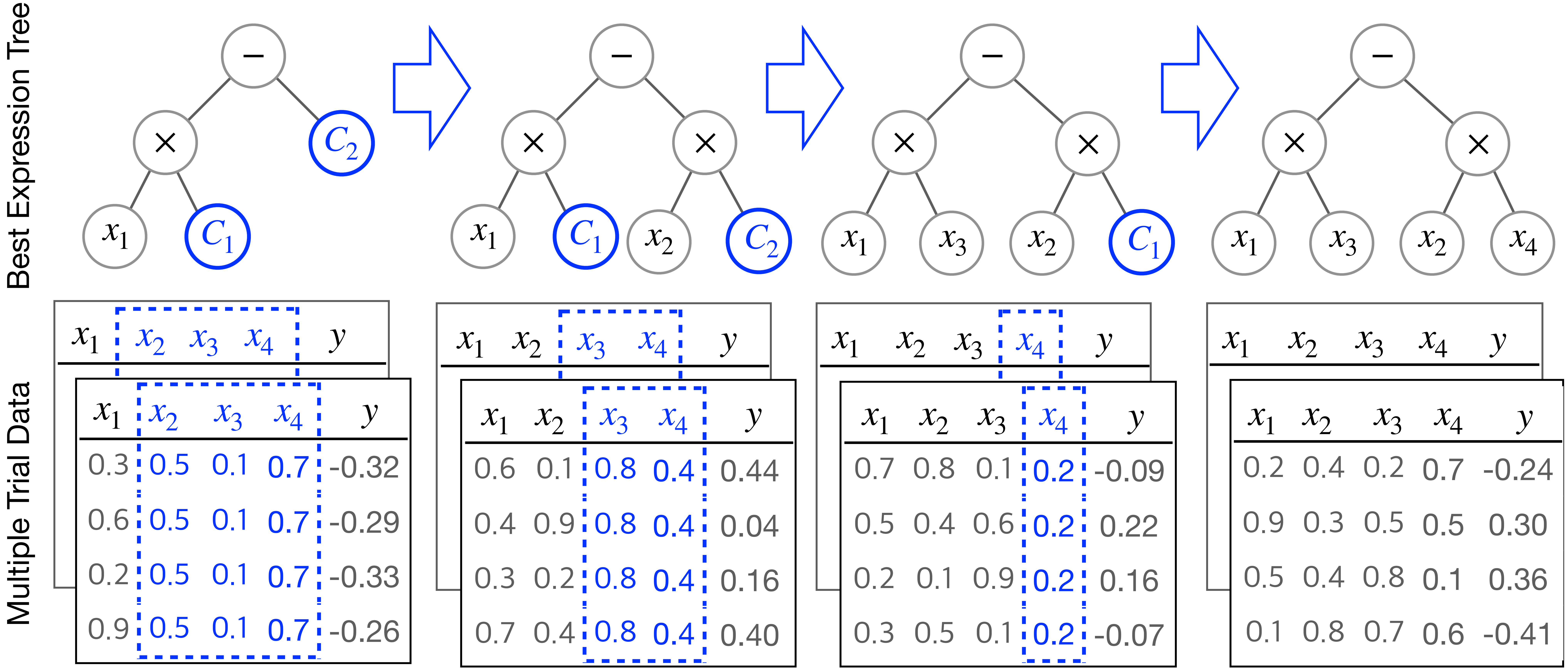
- determine the hypothesis and conduct controlled experiments



Design Principle Scientific Reasoning embedded Symbolic regression

Build the expression from simple to complex, using scientific reasoning.

- **Assumption**: need a data oracle that can return the controlled variables data.



(a) control x_2, x_3, x_4 **(b)** control x_3, x_4 **(c)** control x_4 **(d)** no control

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Execution step 1

classic symbolic regression method

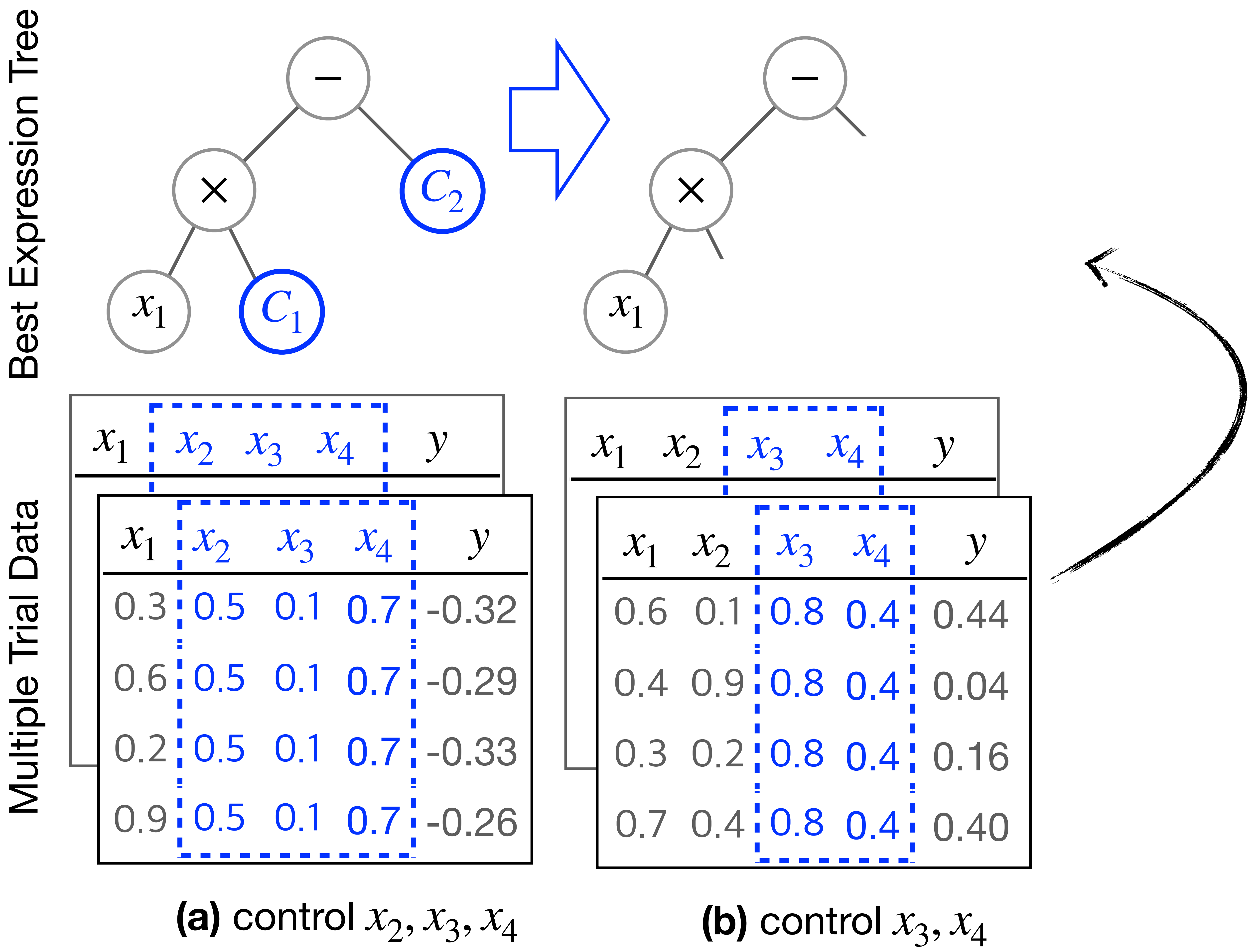
Multiple Trial Data

x_1	x_2	x_3	x_4	y
0.3	0.5	0.1	0.7	-0.32
0.6	0.5	0.1	0.7	-0.29
0.2	0.5	0.1	0.7	-0.33
0.9	0.5	0.1	0.7	-0.26

(a) control x_2, x_3, x_4

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

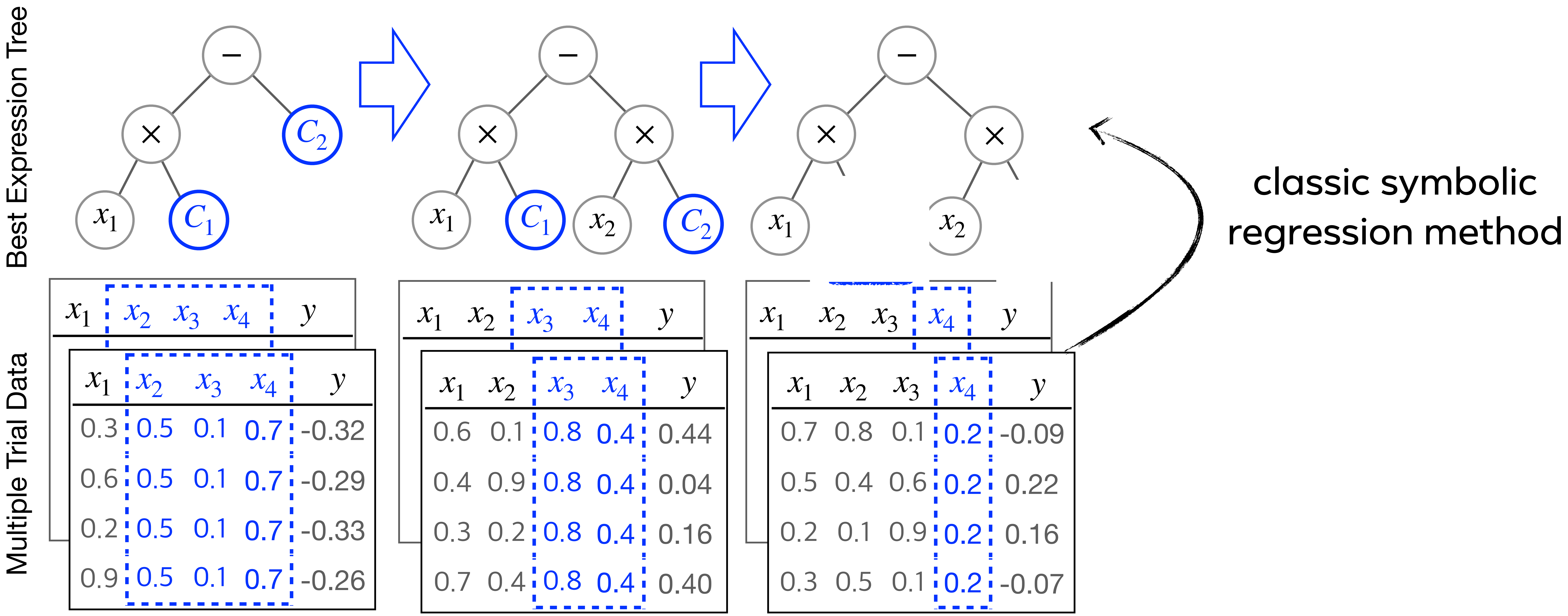
Execution step 2



classic symbolic regression method

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Execution step 3

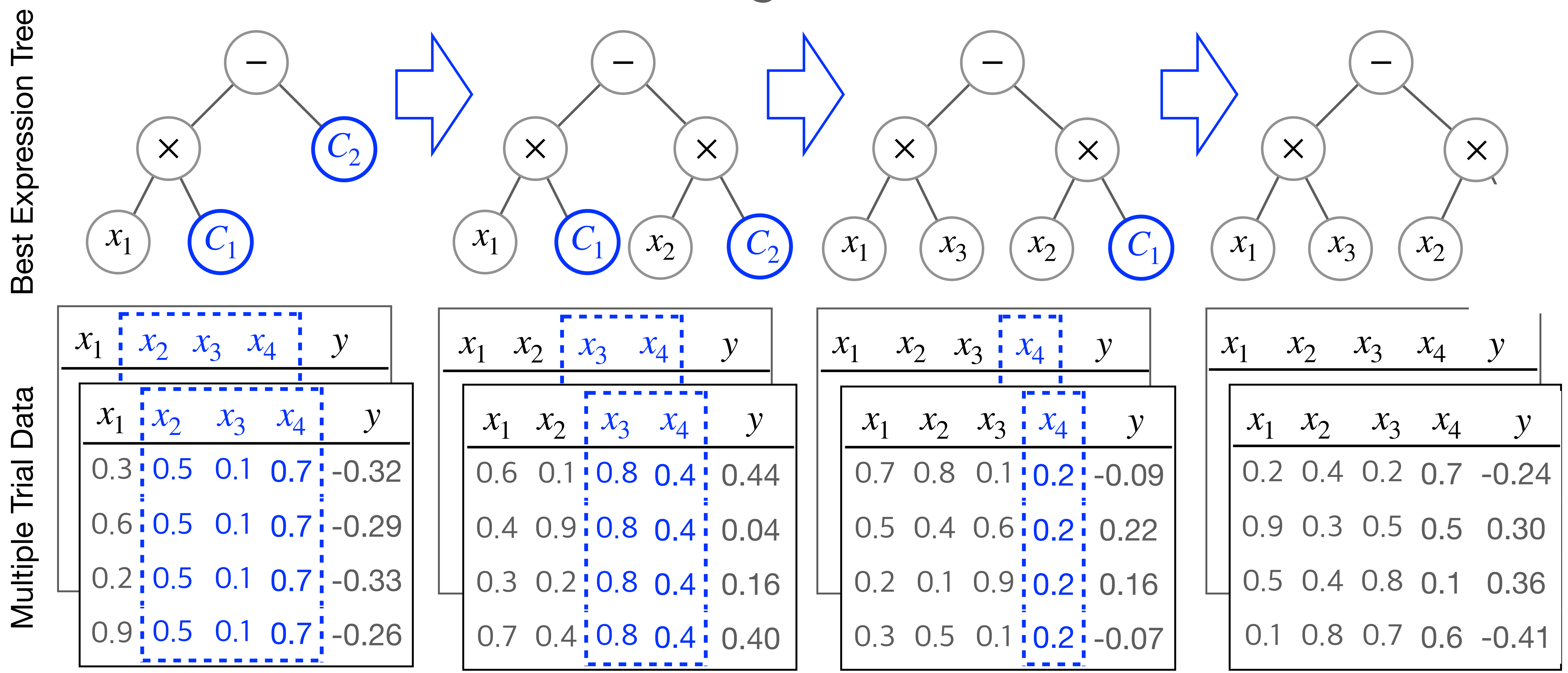


(a) control x_2, x_3, x_4 **(b)** control x_3, x_4 **(c)** control x_4

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Execution step 4

Our method works with many existing symbolic regression methods: like GP, DRL, MCTS, LLM!



x_1	x_2	x_3	x_4	y
0.3	0.5	0.1	0.7	-0.32
0.6	0.5	0.1	0.7	-0.29
0.2	0.5	0.1	0.7	-0.33
0.9	0.5	0.1	0.7	-0.26

(a) control x_2, x_3, x_4

x_1	x_2	x_3	x_4	y
0.6	0.1	0.8	0.4	0.44
0.4	0.9	0.8	0.4	0.04
0.3	0.2	0.8	0.4	0.16
0.7	0.4	0.8	0.4	0.40

(b) control x_3, x_4

x_1	x_2	x_3	x_4	y
0.7	0.8	0.1	0.2	-0.09
0.5	0.4	0.6	0.2	0.22
0.2	0.1	0.9	0.2	0.16
0.3	0.5	0.1	0.2	-0.07

(c) control x_4

x_1	x_2	x_3	x_4	y
0.2	0.4	0.2	0.7	-0.24
0.9	0.3	0.5	0.5	0.30
0.5	0.4	0.8	0.1	0.36
0.1	0.8	0.7	0.6	-0.41

(d) no control

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Scientific Reasoning brings an exponential reduction of the search space for a class of equations

There exists a family of symbolic expression ϕ of $(4m-1)$ nodes,

One example: $(x_1+x_2)(x_3+x_4)\ldots(x_{2m-1}+x_{2m})$.

- Classic symbolic regression following the simple to complex search order has to explore a search space whose size is $O(e^m)$ to find the expression.
- Our scientific reasoning following the simple to complex order expands $O(m)$ search spaces.

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Experiments on large-scale algebraic equation dataset

Benchmark on Normalized Mean-squared error metric.

Methods	Total variables				
	10	20	30	40	50
Monte Carlo Tree Search	0.386	0.554	0.554	0.714	0.815
Genetic Programming	0.159	0.172	0.218	0.229	0.517
Deep RL with risk-seeking policy gradient	0.284	0.521	0.522	0.66	0.719
Deep RL with vanilla policy gradient	0.415	0.695	0.726	0.726	0.779
Deep RL with priority queue training	0.384	0.488	0.615	0.62	0.594
Our method	1E-06	1E-06	1E-06	0.002	0.021

Our method successfully scales up to dataset with 50 variable due to scientific reasoning.

Nan Jiang et al., ECML2022; IJCAI 2024; AAAI2024; AAAI2025.

Use scientific reasoning to find the governing PDE for nano voids

Nanovoid evolution video

↓

Computer Vision Annotation

↓

Annotated voids moving over time

↓

Reasoning the governing PDE

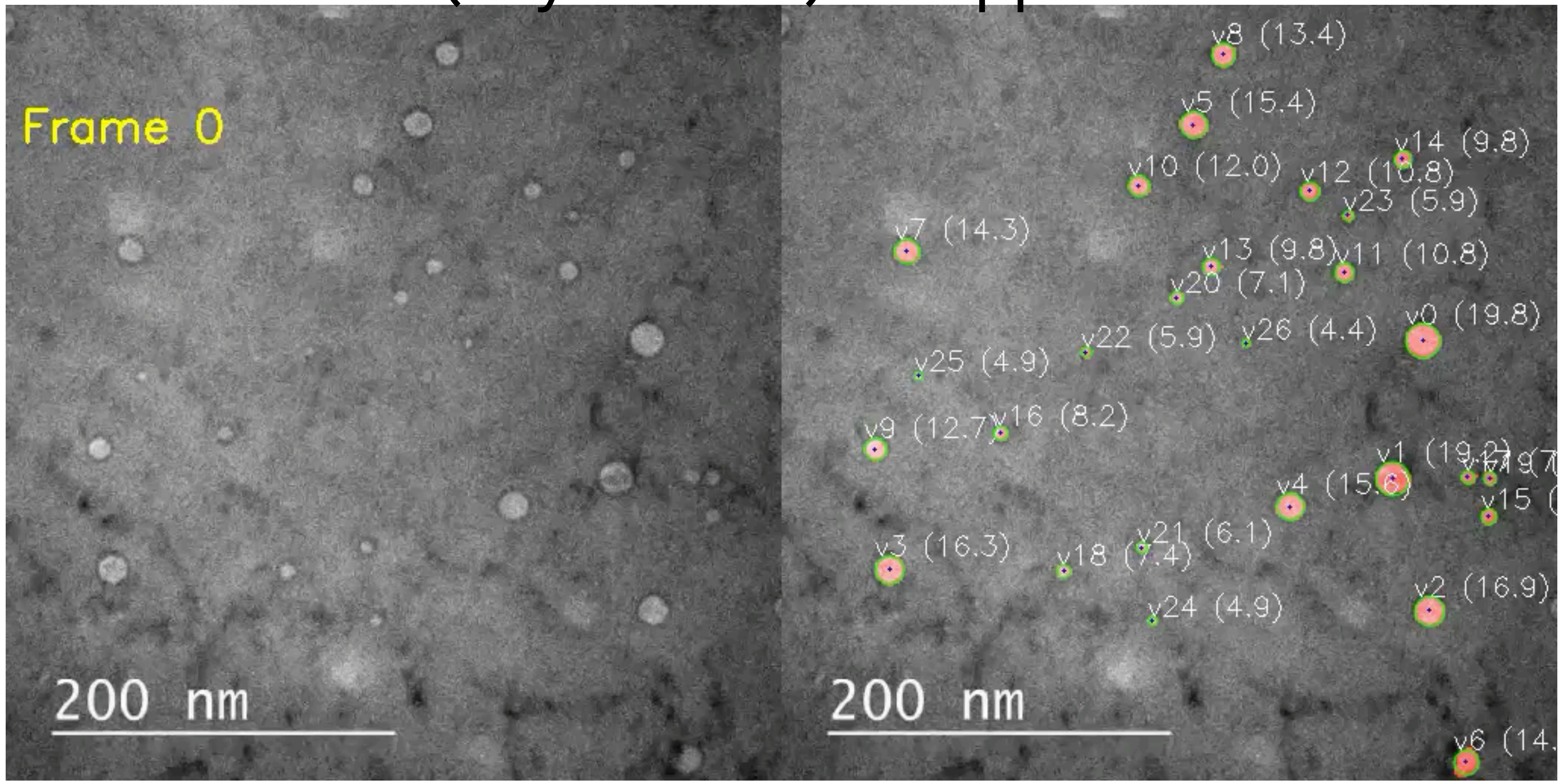
↓

$$\frac{\partial c_v}{\partial t} = \nabla \cdot (M_v \nabla \frac{1}{N} \frac{\delta F}{\delta c_v}) + \xi(\mathbf{r},t) + P_v(\mathbf{r},t) - R_{iv}(\mathbf{r},t),$$

$$\frac{\partial c_i}{\partial t} = \nabla \cdot (M_i \nabla \frac{1}{N} \frac{\delta F}{\delta c_i}) + \zeta(\mathbf{r},t) + P_i(\mathbf{r},t) - R_{iv}(\mathbf{r},t),$$

$$\frac{\partial \eta}{\partial t} = - L \frac{\delta F}{\delta \eta} + \iota(\mathbf{r},t) + P_v(\mathbf{r},t).$$

The growth and movement of nanoscale voids (tiny cavities) in copper.

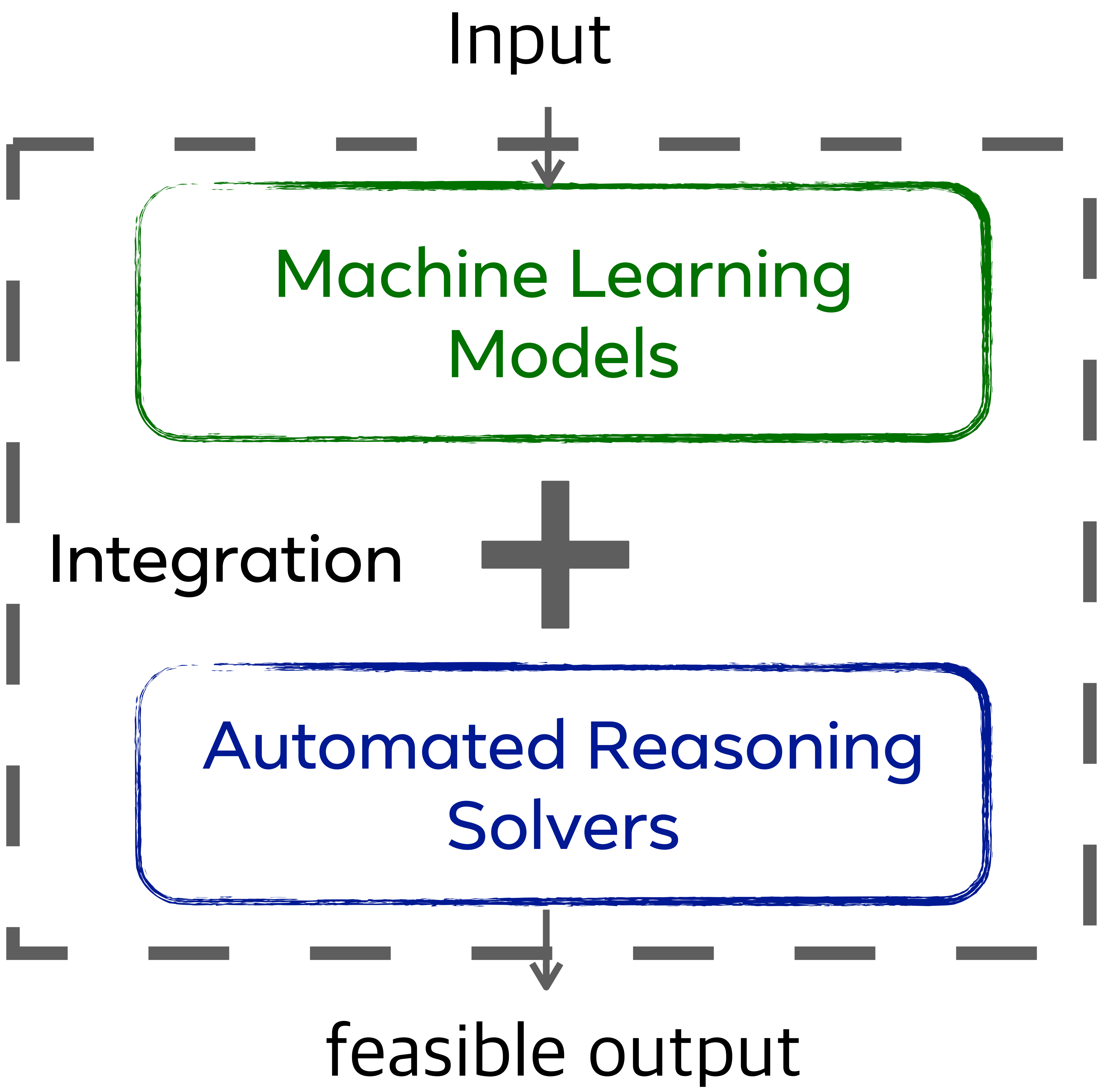


It enables scientific discovery on nano voids size fluctuation (nasim et al., Journal of Nuclear Materials 2022).

Takeaway

The benefits are:

- Formal guarantee on Constraint satisfaction.
- Scalablilty: Accelerate learning for higher-dimensional data.



Outline

1. Verifiability:
Reasoning with learning to ensure constraint satisfaction for structured prediction.

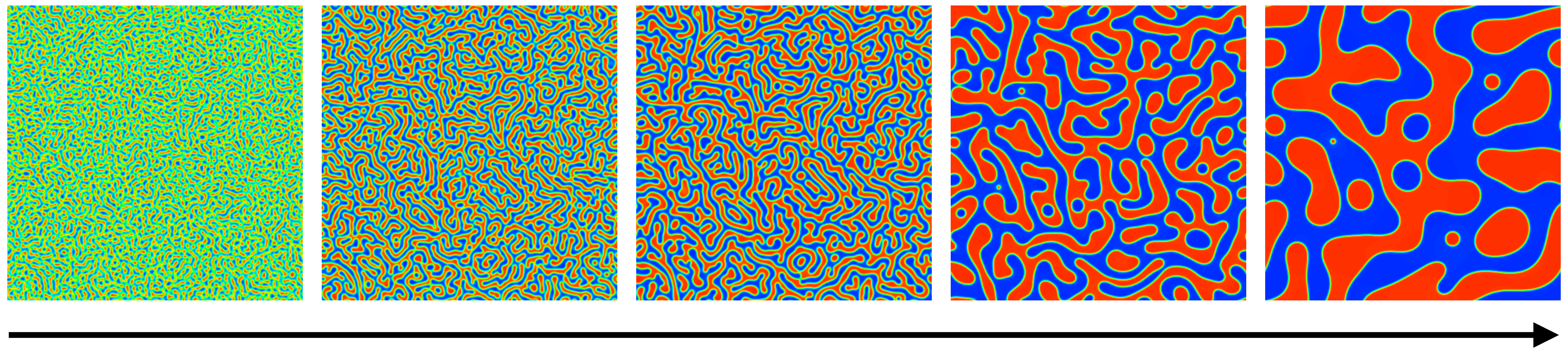
2. Scalability:
Reasoning with learning to accelerate scientific discovery.

3. Future work

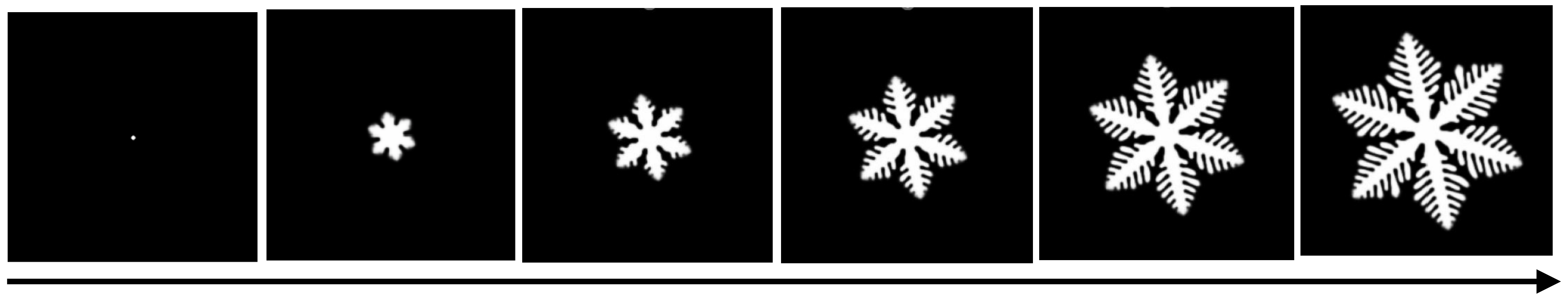
Thrust 1: Embed Reasoning in learning for accelerating scientific discovery

Discover knowledge for extensive scientific problems, like:

- Spinnodal decomposition, like oil and water mixes.



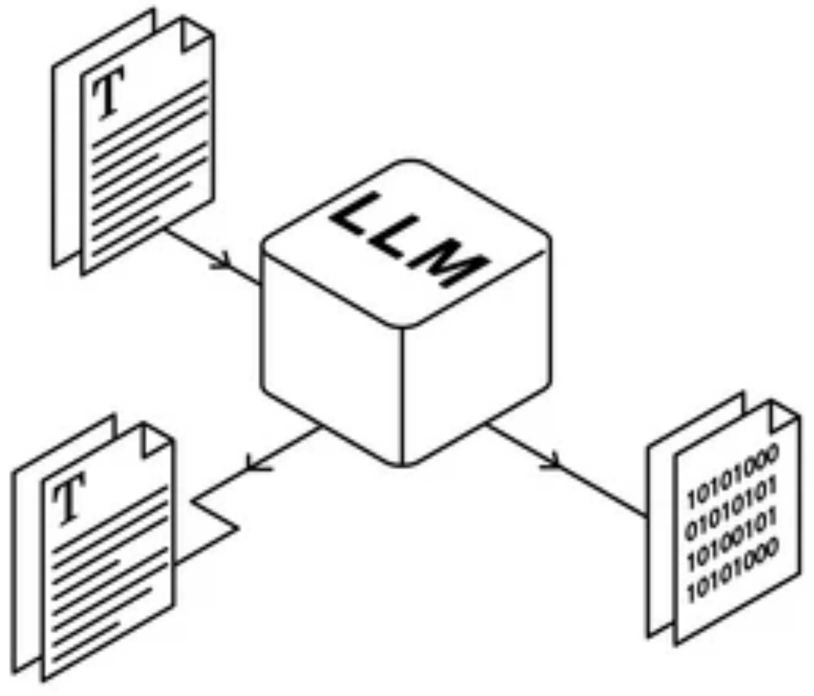
- Dendritic solidification, like the growth of snow flake.

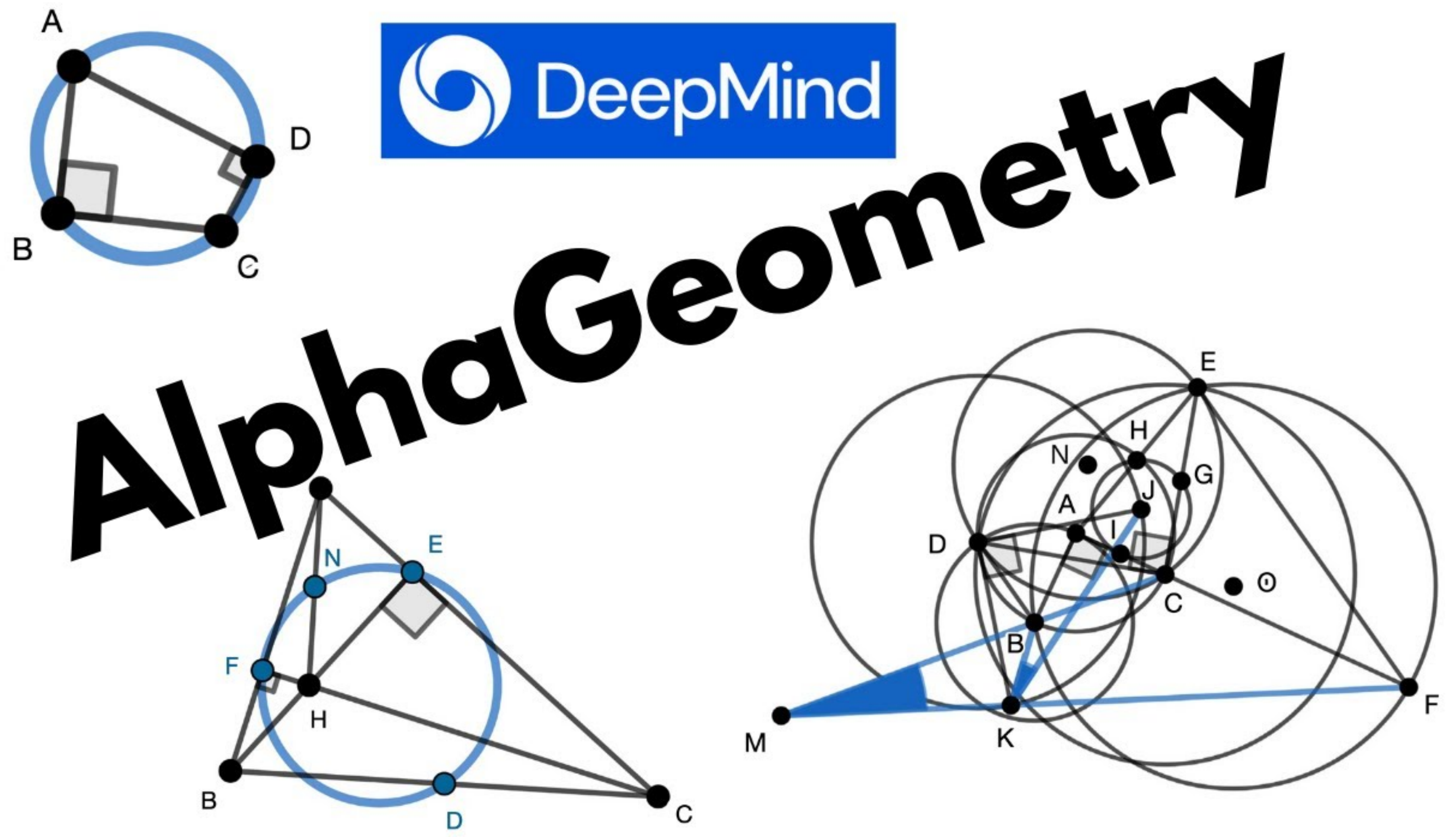


What is the governing dynamics in those physical phenomena?

Thrust 2 & 3: Formal Reasoning for Large Language Model

Combine intuitive and informal language description with formal reasoning (symbolic proofs, rule-based derivations) for verifiable and accurate prediction.



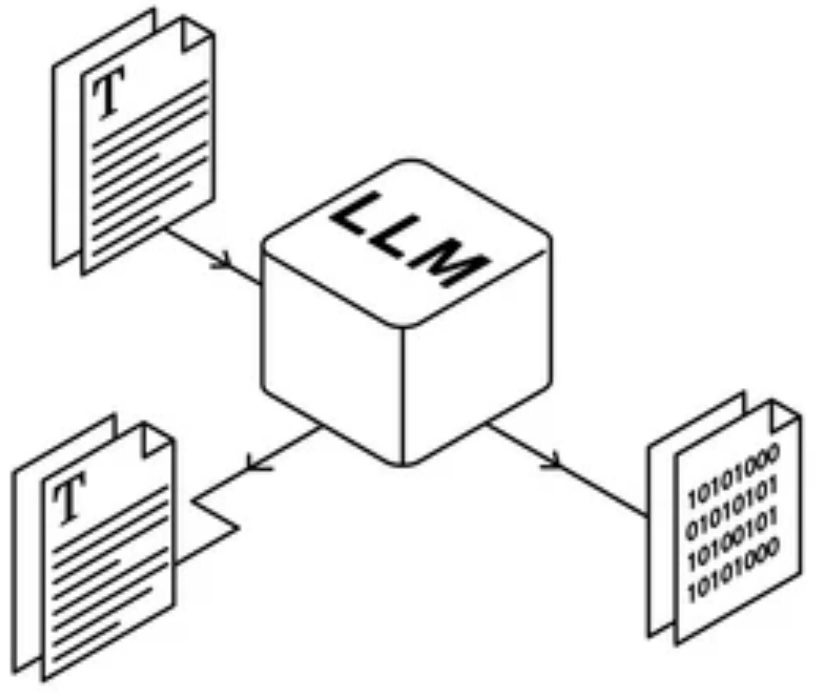


2d geometry problems

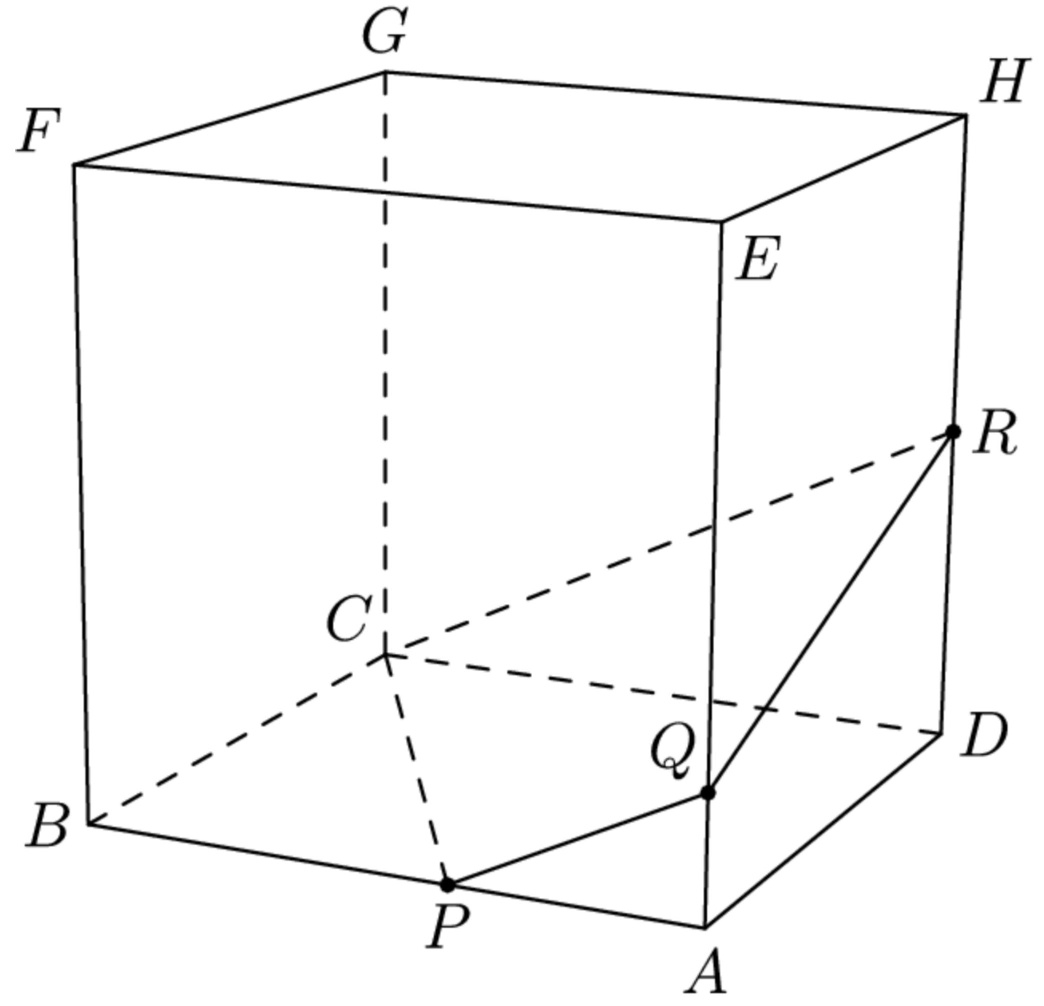
How to solve hard mathematical problems with large language model and symbolic solvers?

Thrust 2 & 3: Formal Reasoning for Large Language Model

Combine intuitive and informal language description with formal reasoning (symbolic proofs, rule-based derivations) for verifiable and accurate prediction.



Let $ABCDEFGH$ be a cube of side length 5, as shown. Let P and Q be points on $\overline{AB}$ and $\overline{AE}$, respectively, such that $AP = 2$ and $AQ = 1$. The plane through C, P, and Q intersects $\overline{DH}$ at R. Find DR.

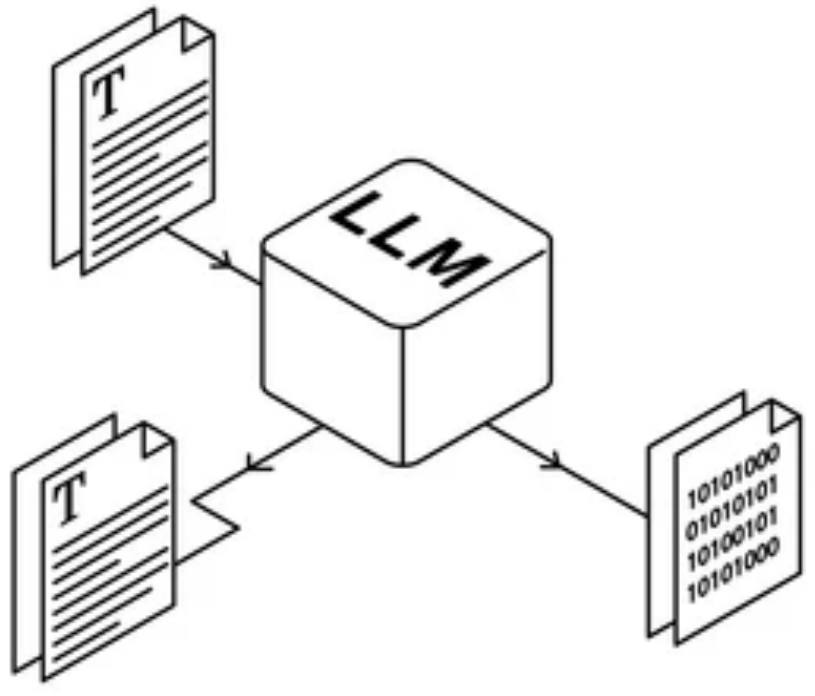


3d geometry problems

How to solve hard mathematical problems with large language model and symbolic solvers?

Thrust 2 & 3: Formal Reasoning for Large Language Model

Combine intuitive and informal language description with formal reasoning (symbolic proofs, rule-based derivations) for verifiable and accurate prediction.




```
theorem and_commutative (p q : Prop) : p ∧ q → q ∧ p :=
assume hpq : p ∧ q,
have hp : p, from and.left hpq,
have hq : q, from and.right hpq,
show q ∧ p, from and.intro hq hp
```

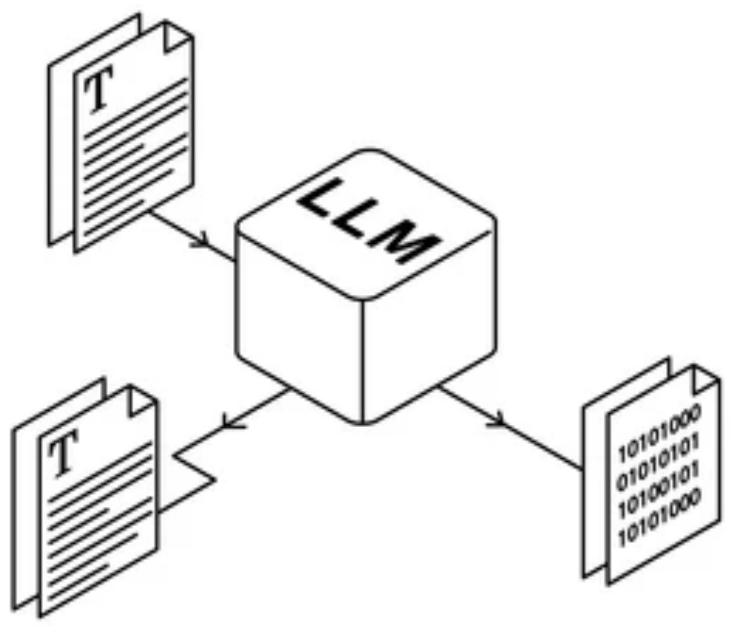
LEAN THEOREM PROVER

Theorem proving problems

How to solve hard mathematical problems with large language model and symbolic solvers?

Thrust 2 & 3: Formal Reasoning for Large Language Model

Combine intuitive and informal language description with formal reasoning (symbolic proofs, rule-based derivations) for verifiable and accurate prediction.



Problem

H-Index

Given a list of citations counts, where each citation is a nonnegative integer, write a function h_index that outputs the h-index. The h-index is the largest number *h* such that *h* papers have each least *h* citations.

Example:
Input: [3,0,6,1,4]
Output: 3

Generated Code

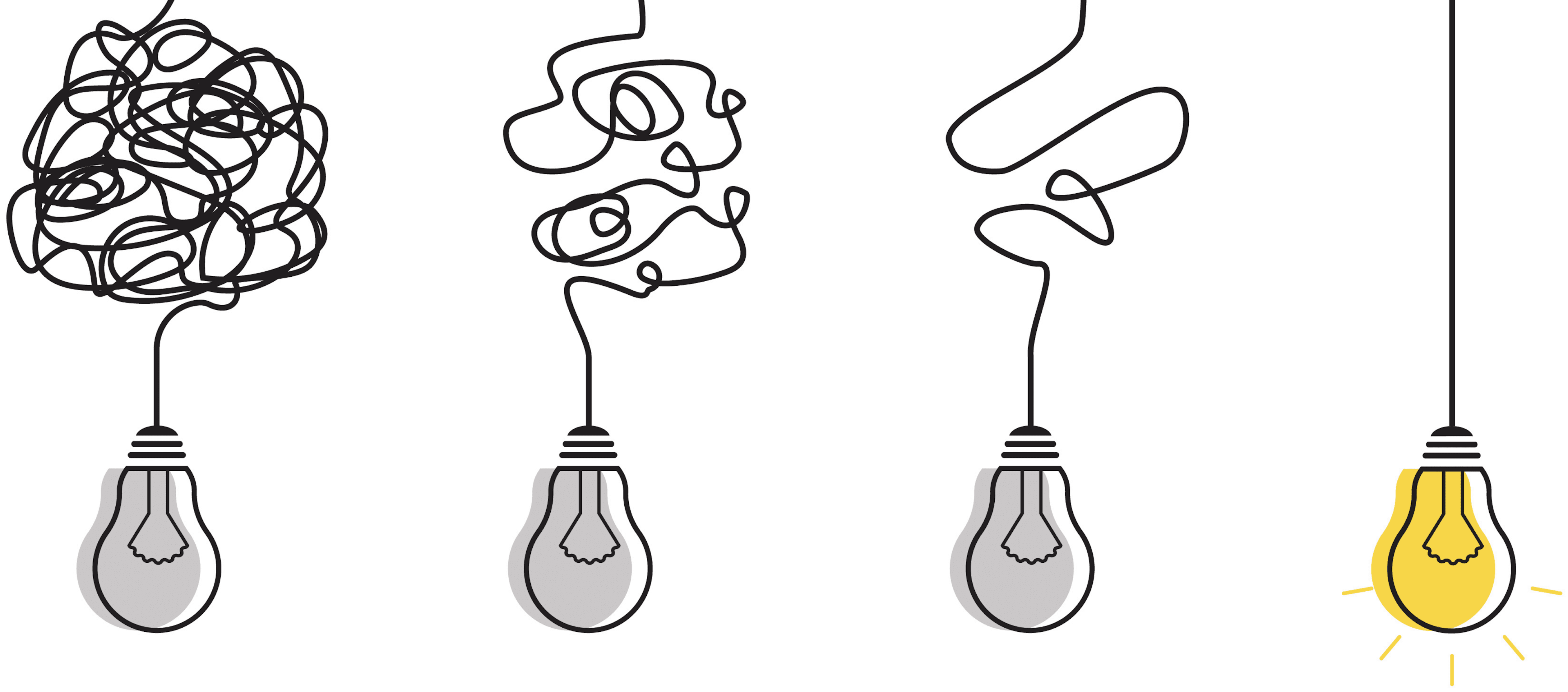
```
def h_index(counts):
  n = len(counts)
  if n > 0:
    counts.sort()
    counts.reverse()
    h = 0
    while (h < n and
        counts[h]-1>=h):
      h += 1
    return h
  else:
    return 0
```

Test Cases

```
Input:
[1,4,1,4,2,1,3,5,6]

Generated Code Output:
4                          ✓

Input:
[1000,500,500,250,100,
100,100,100,100,75,50,
30,20,15,15,10,5,2,1]

Generated Code Output:
15                         ✓
```

What is the code implementation with a given text description of task? like

- competitive programming,
- efficient low-level execution code.
- Automatic program repair
- Automatic testing function generation



Q&A

Nan Jiang, CS@Purdue

https://jiangnanhugo.github.io