The 39th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, PENNSYLVANIA, USA

CPML Bridge Program

Integrating Automated Reasoning and Machine Learning for Structured Prediction

On the job market!

Nan Jiang
Purdue University
jiang631@purdue.edu

Two Pillars in AI: Machine Learning and Automated Reasoning

Machine Learning

Automated Reasoning

Two Pillars in AI: Machine Learning and Automated Reasoning

Machine Learning

Automated Reasoning

Bottom-up and Inductive: Fit data distributions well.

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

ChatGPT

Midjourney

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

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

Two Pillars in AI: Machine Learning and Automated Reasoning

Machine Learning

Automated Reasoning

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

Two Pillars in AI: Machine Learning and Automated Reasoning

Machine Learning

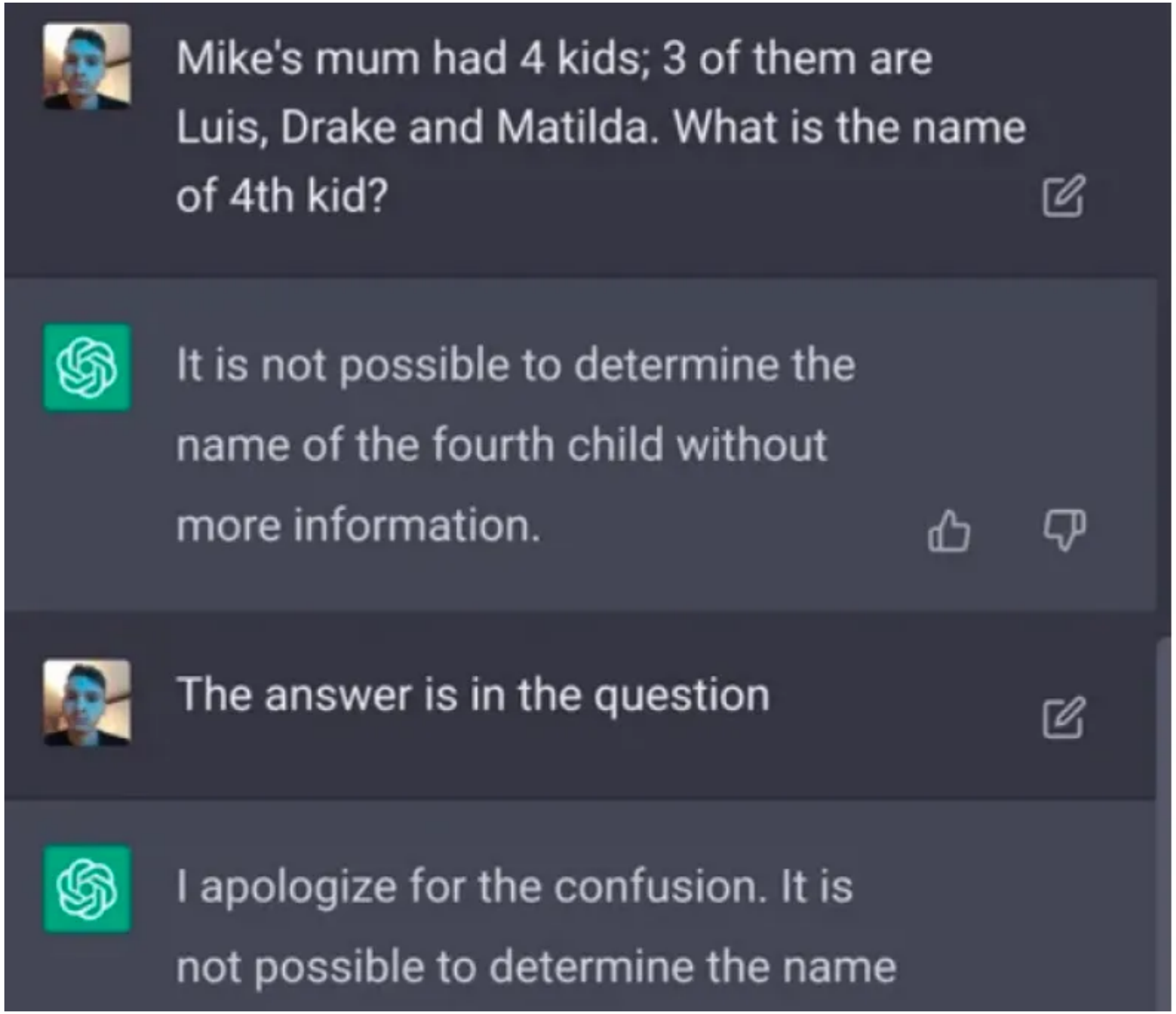
- Challenging in providing formal guarantees.
- Hallucination: generated outputs are false or fabricated.
- May violate constraints in rare and unseen situations.

Automated Reasoning

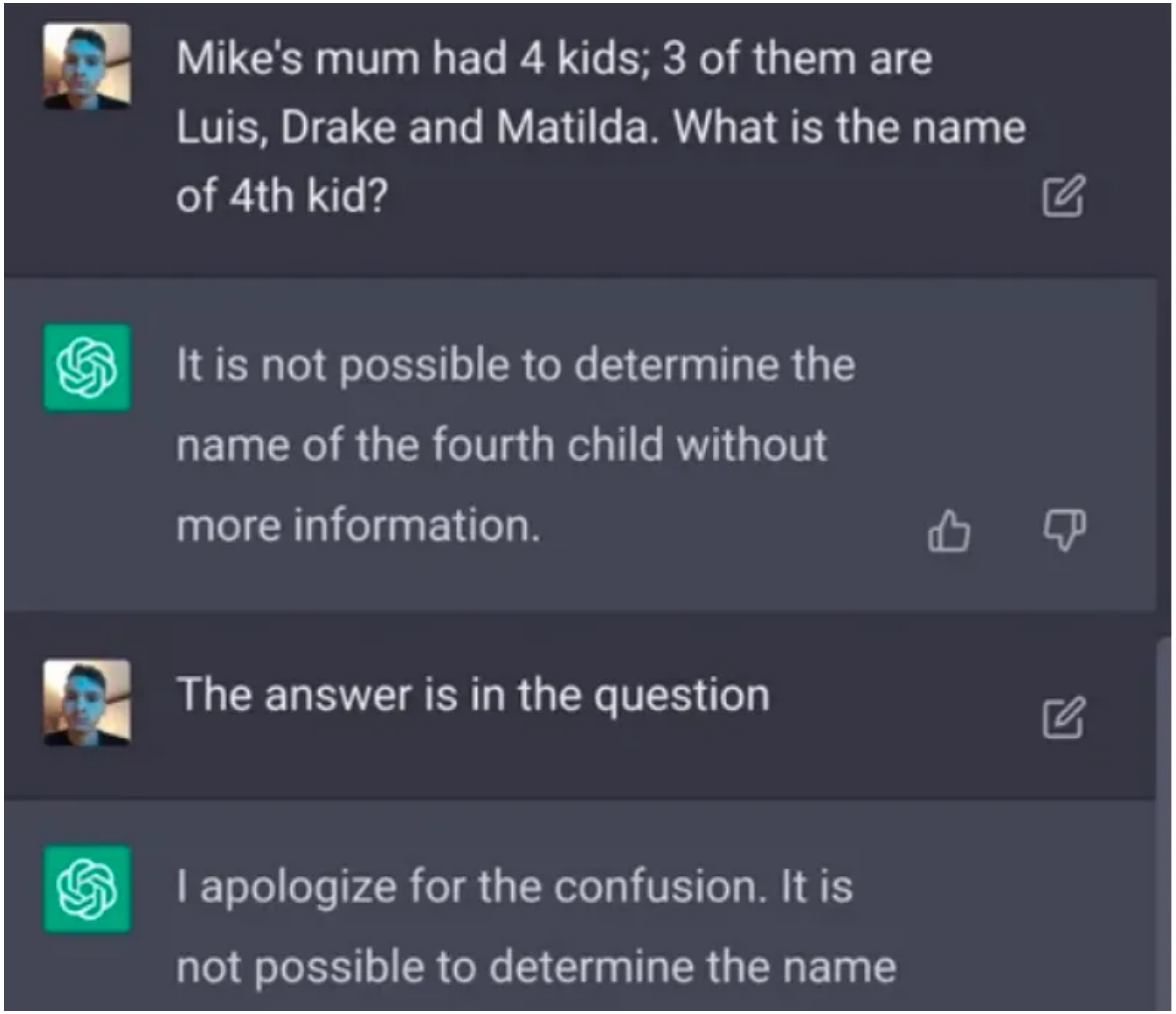
- Rigid models: problem formulation must be agreed a-priori.
- Difficult to adapt to evolving data distributions.
- Cannot understand data like text and images.

Machine Learning has intrinsic difficulty

Machine Learning has intrinsic difficulty

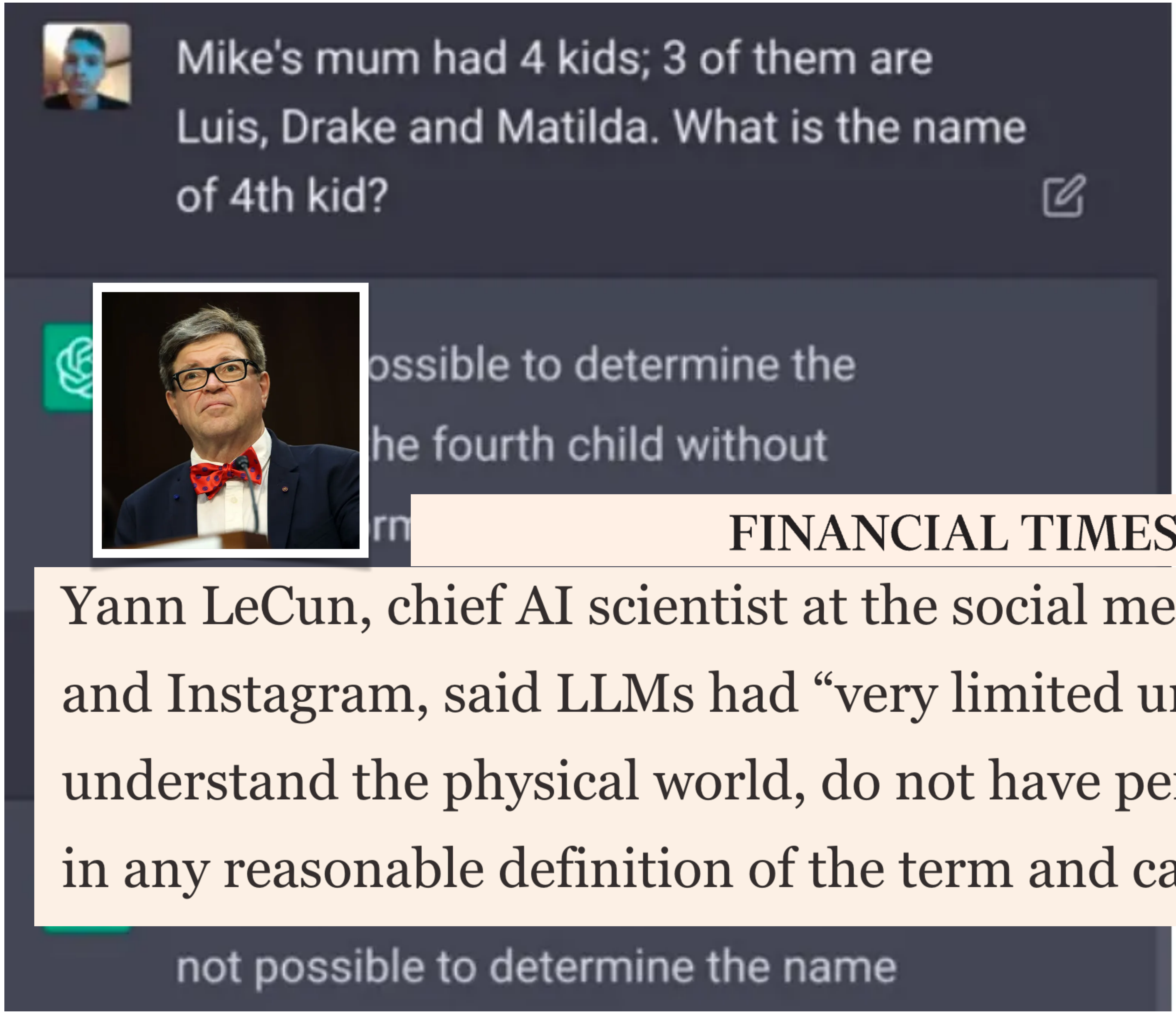


Machine Learning has intrinsic difficulty



ChatGPT struggle with questions in logical reasoning and context comprehension.

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

Automated Reasoning has intrinsic difficulties

Input $(x_1 \lor x_2) \land (\neg x_1 \lor x_3)$

↓

SATisfiability Solvers

↓

Feasible variable assignment

- hard to encode data distribution.

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 language and image

Bridging Machine Learning and Automated Reasoning is Crucial!

Machine Learning	Automated Reasoning

Bridging Machine Learning and Automated Reasoning is Crucial!

	Machine Learning	Automated Reasoning
+	Learn data distribution	

Bridging Machine Learning and Automated Reasoning is Crucial!

	Machine Learning	Automated Reasoning
+	Learn data distribution	
−	Provide formal guarantees	

Bridging Machine Learning and Automated Reasoning is Crucial!

	Machine Learning	Automated Reasoning
+	Learn data distribution	Feasible output
−	Provide formal guarantees	

Bridging Machine Learning and Automated Reasoning is Crucial!

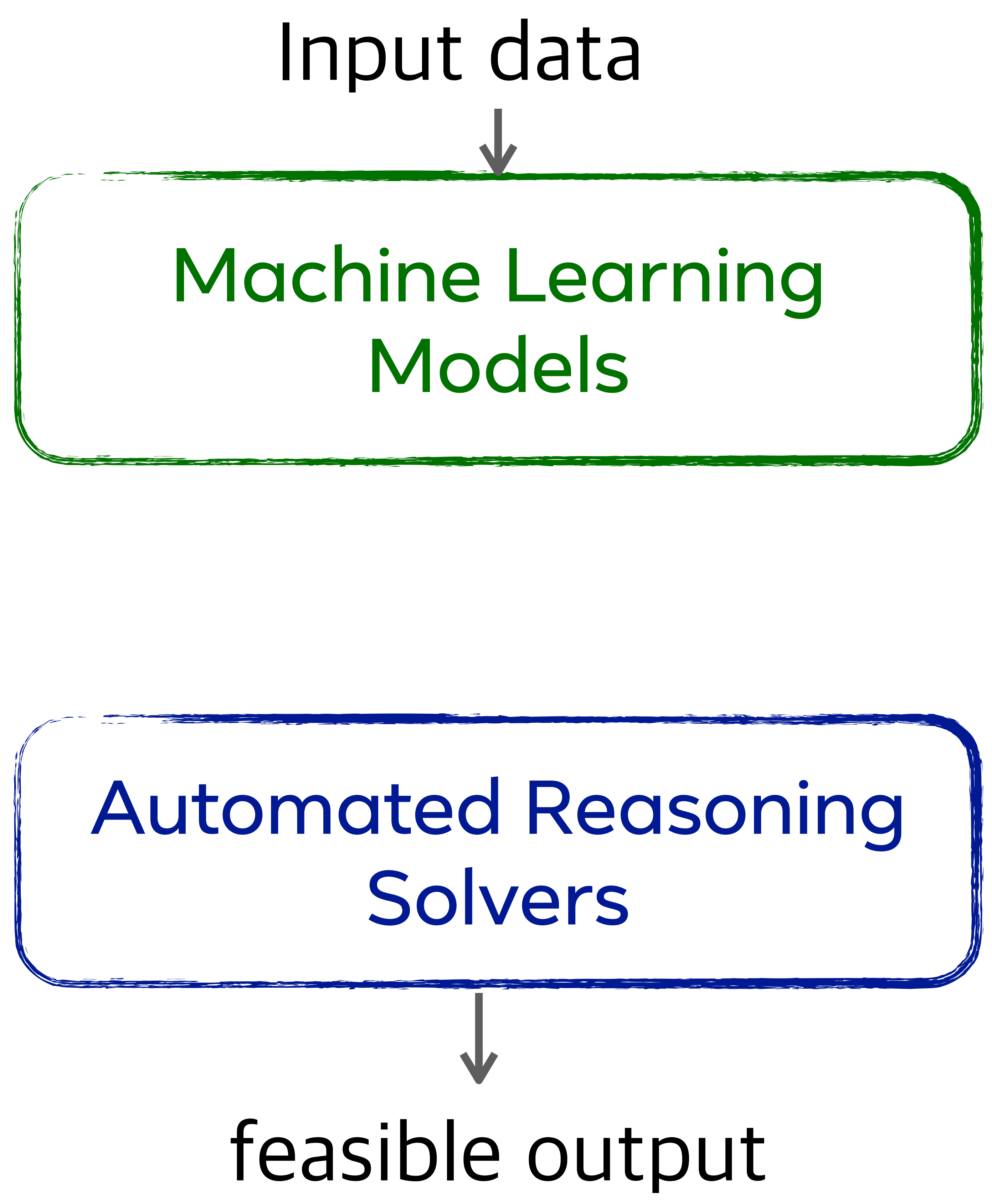
	Machine Learning	Automated Reasoning
+	Learn data distribution	Feasible output
−	Provide formal guarantees	Encode evolving data distribution

Bridging Machine Learning and Automated Reasoning is Crucial!

	Machine Learning	Automated Reasoning
+	Learn data distribution	Feasible output
−	Provide formal guarantees	Encode evolving data distribution

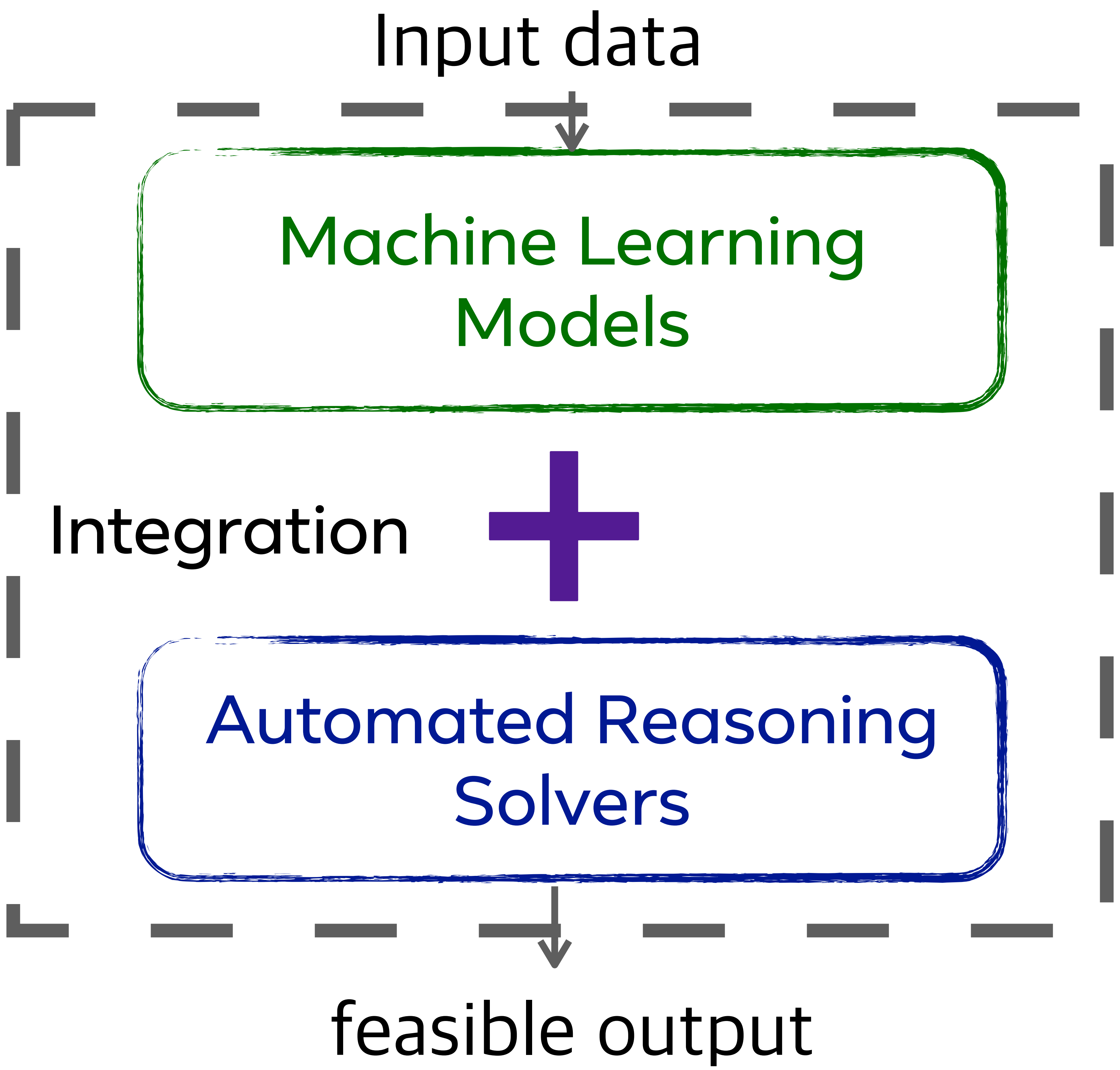
A class of Structured prediction problems are beyond the reach of machine learning and automated reasoning, when they are applied in isolation.

My Research: Integrate Learning with Reasoning



My Research: Integrate Learning with Reasoning

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

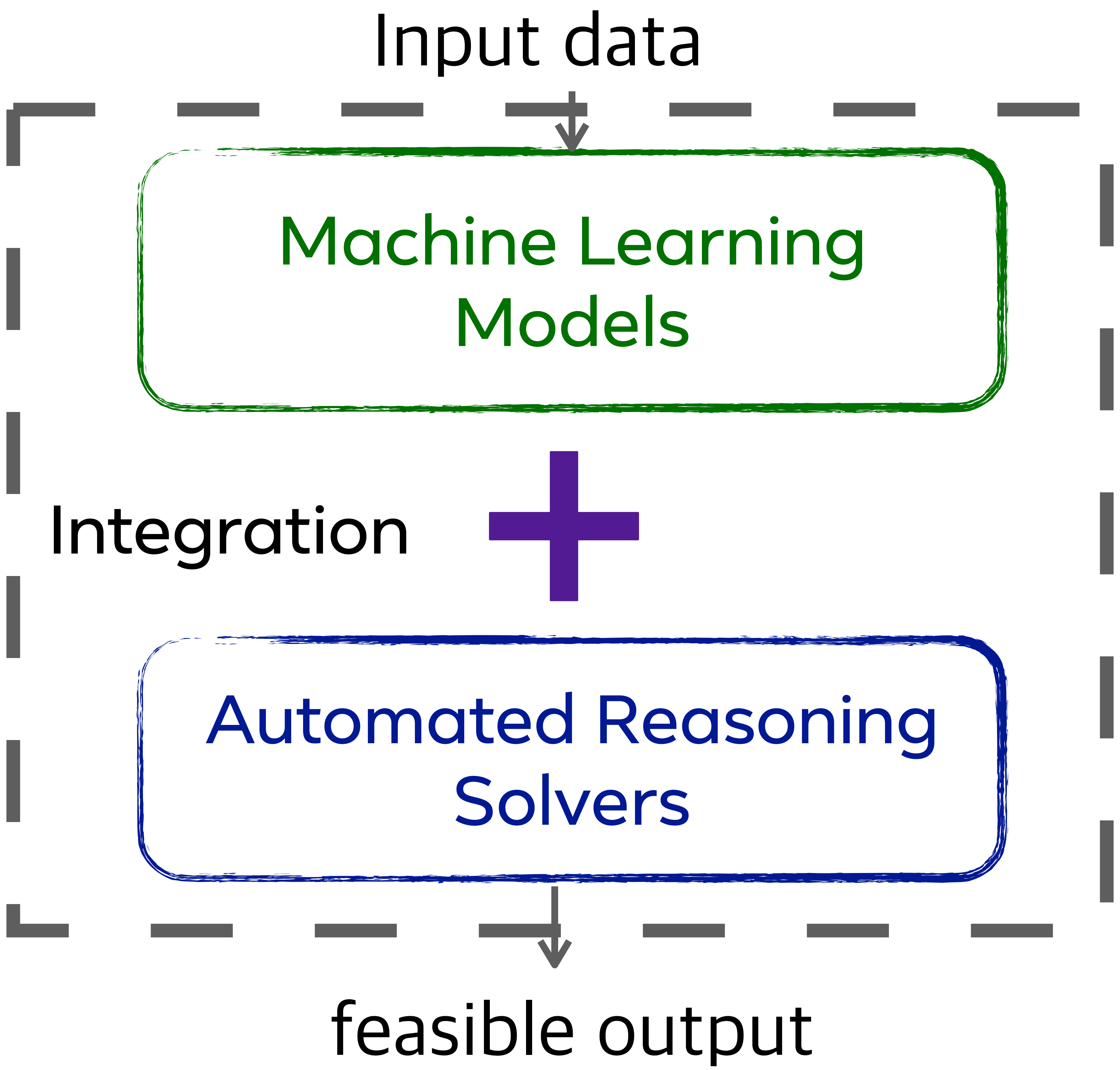


My Research: Integrate Learning with Reasoning

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

The benefits are:

- Formal guarantee of constraint satisfaction.

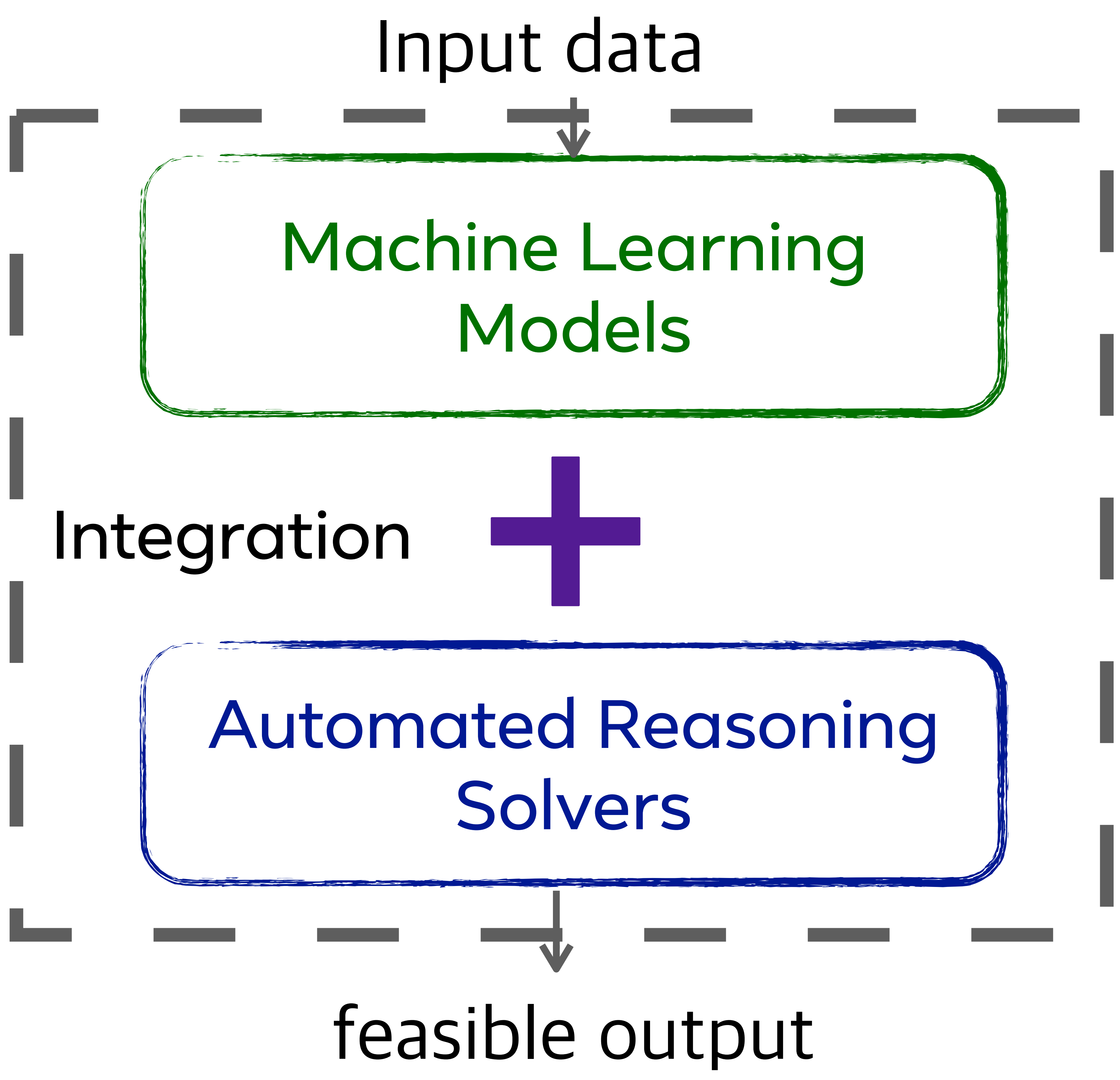


My Research: Integrate Learning with Reasoning

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

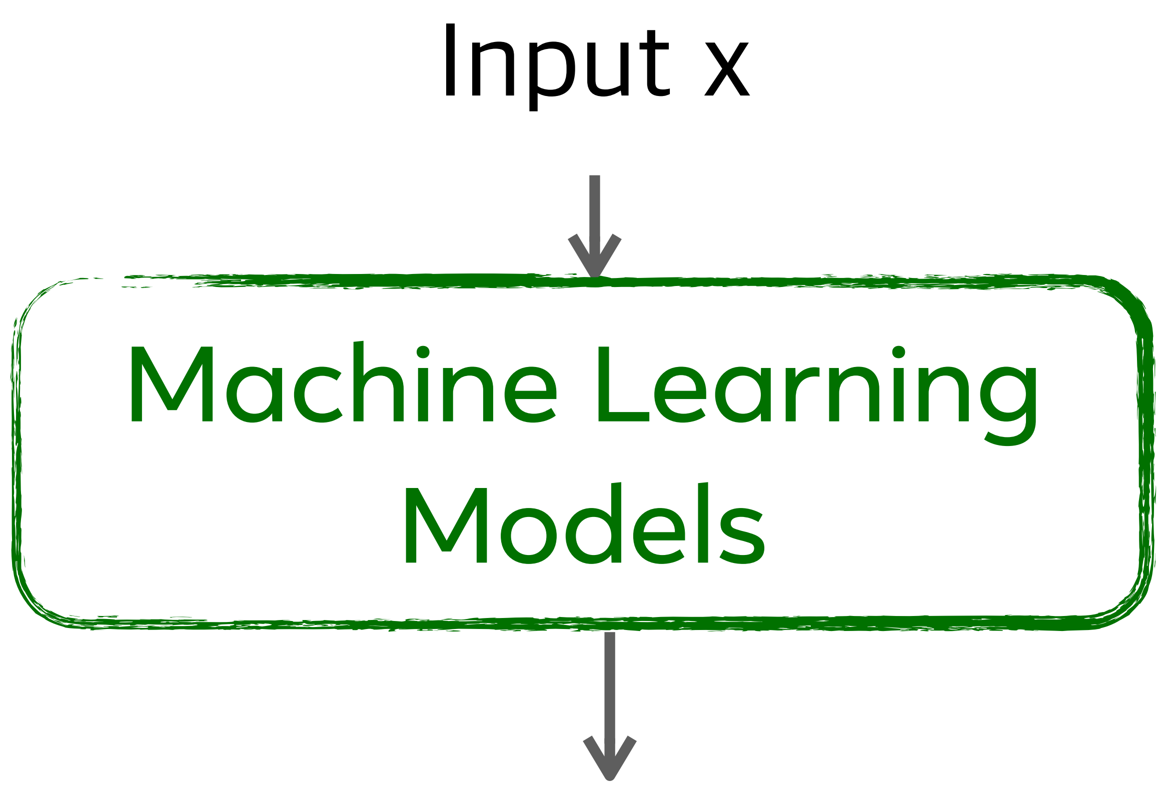
The benefits are:

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



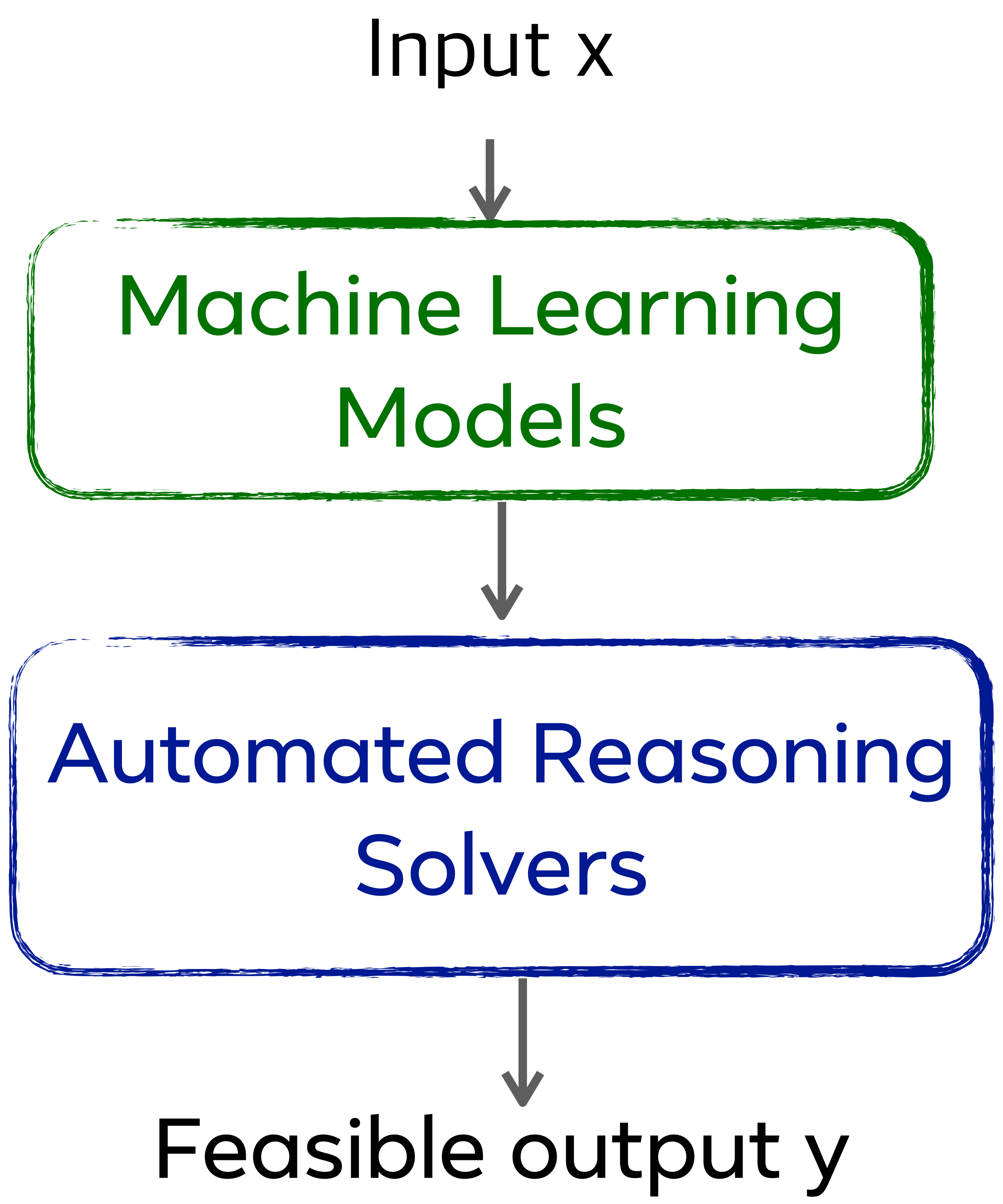
Design principle of the integrated system

- Learn from data



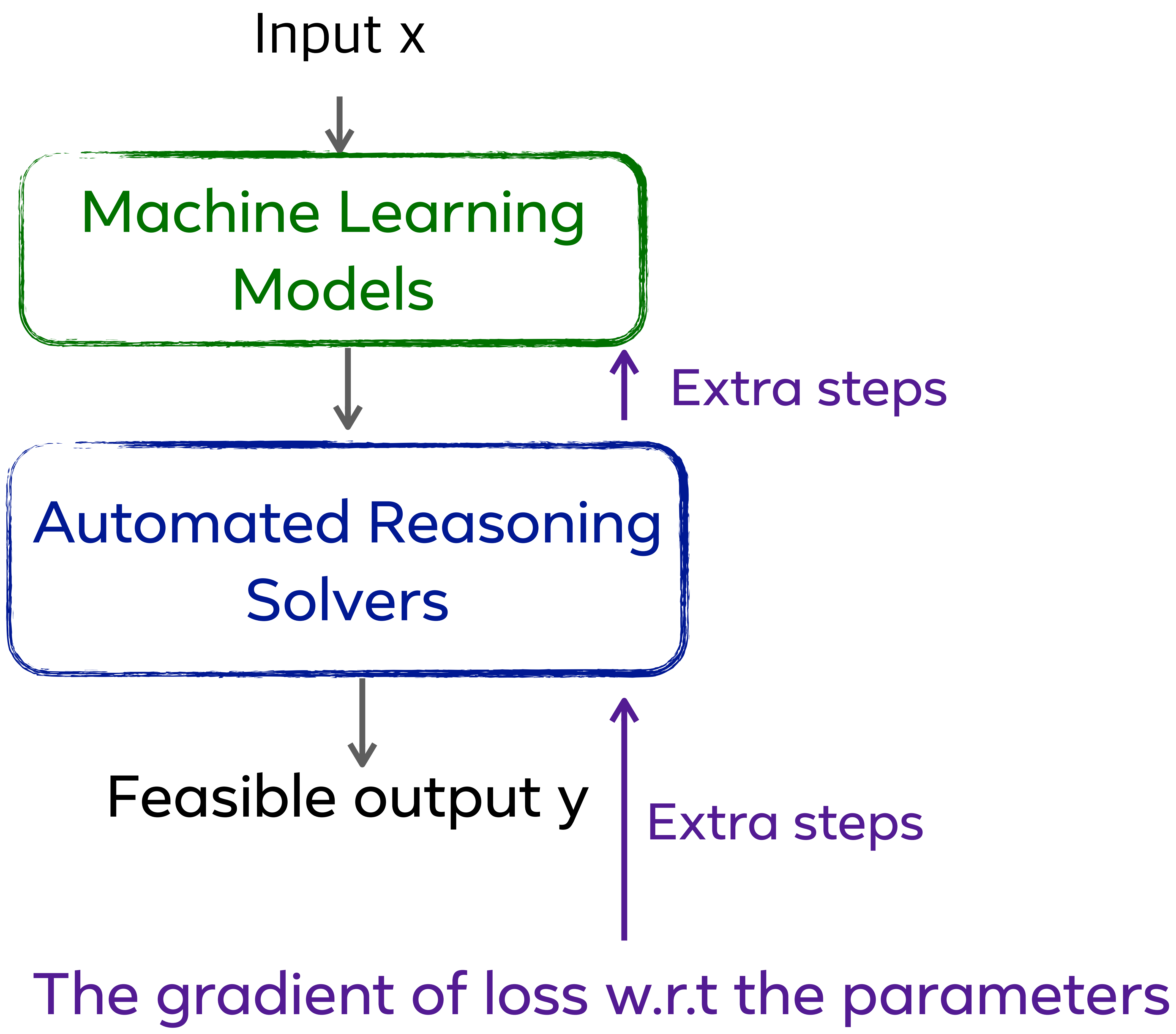
Design principle of the integrated system

- Learn from data
- Satisfy the constraints

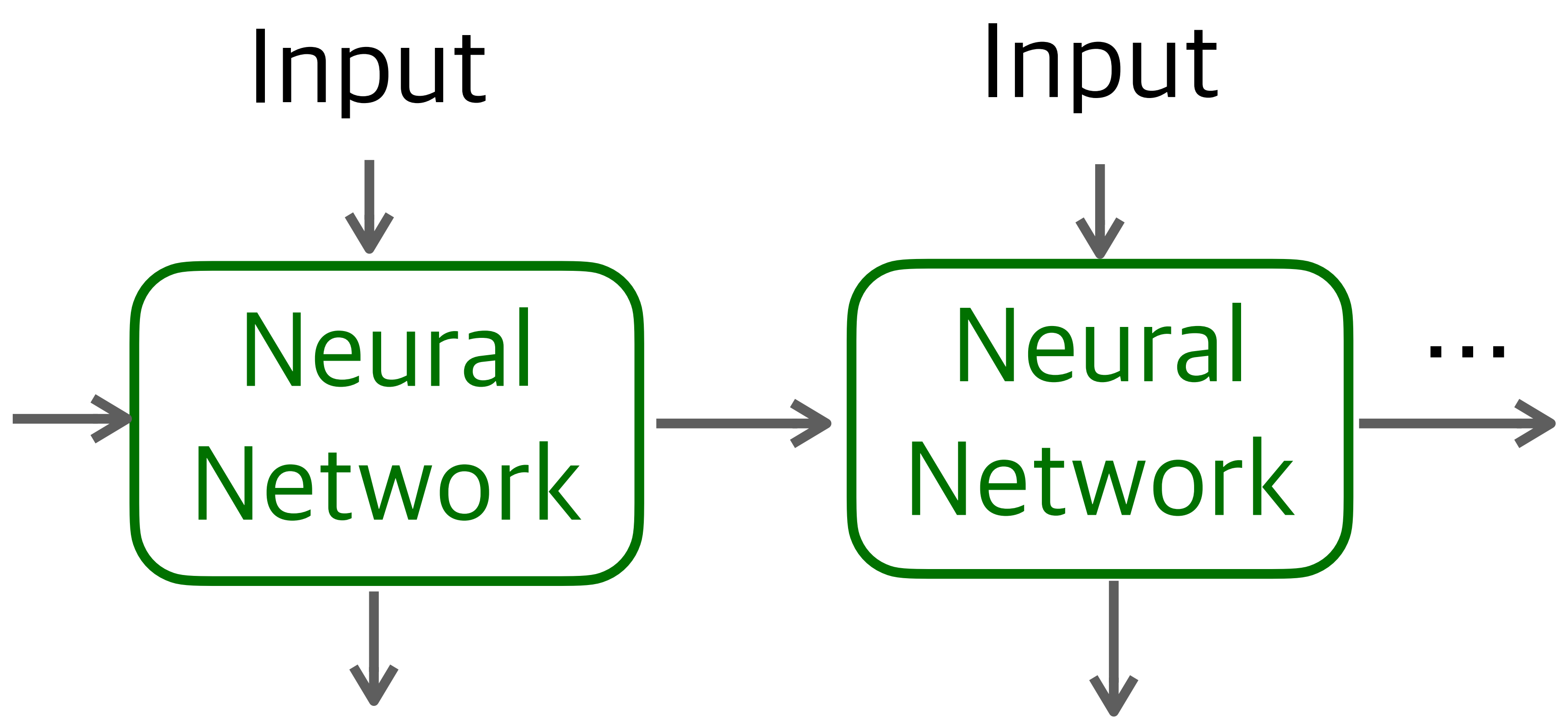


Design principle of the integrated system

- Learn from data
- Satisfy the constraints
- Differentiable

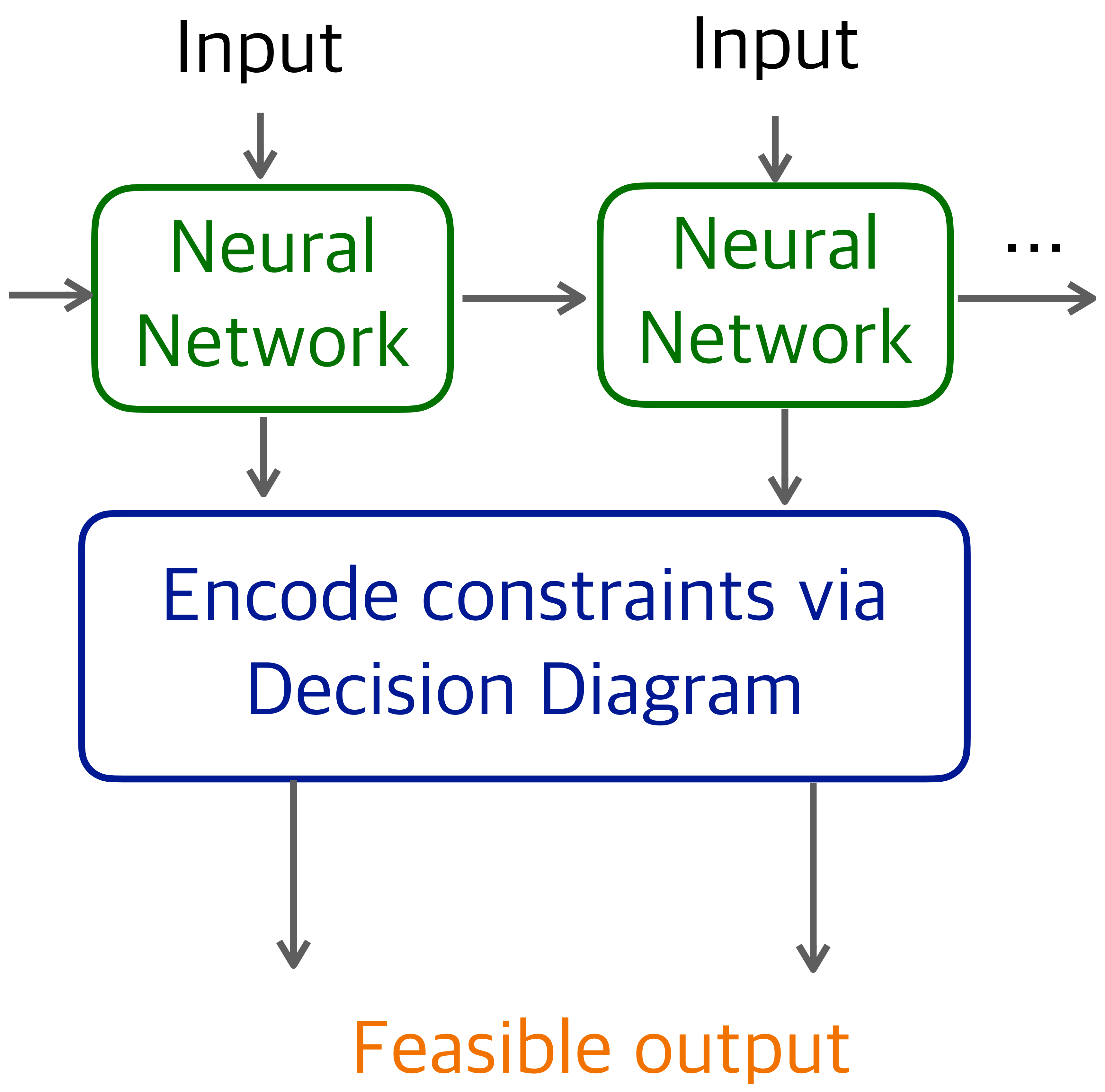


Integrate reasoning with learning to ensure <u>constraint satisfaction</u> for structured prediction



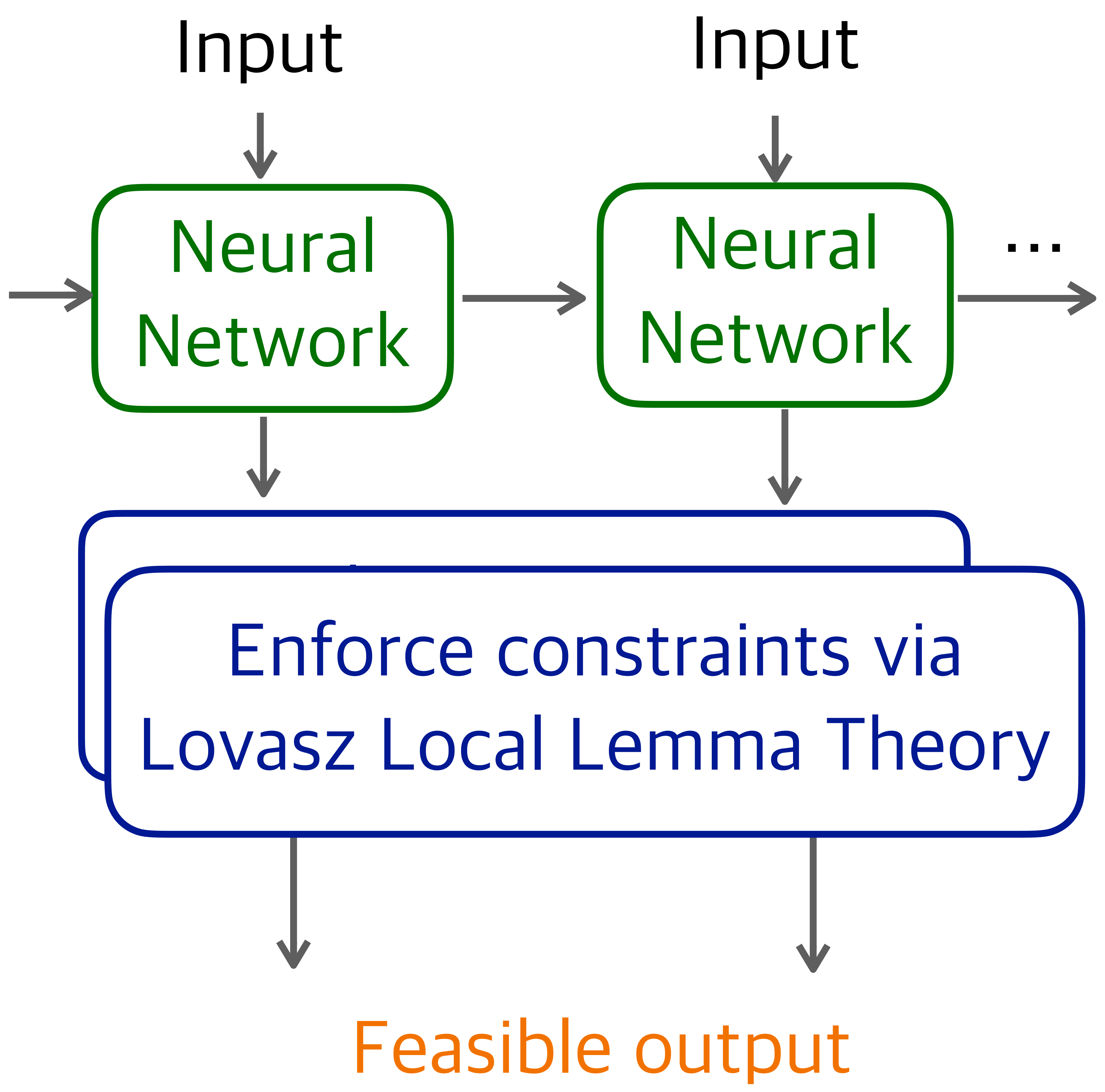
- A series of conference publications: EMNLP2020, UAI2021, JMLR2022, AAAI2023, IJCAI2024.

Integrate reasoning with learning to ensure constraint satisfaction for structured prediction



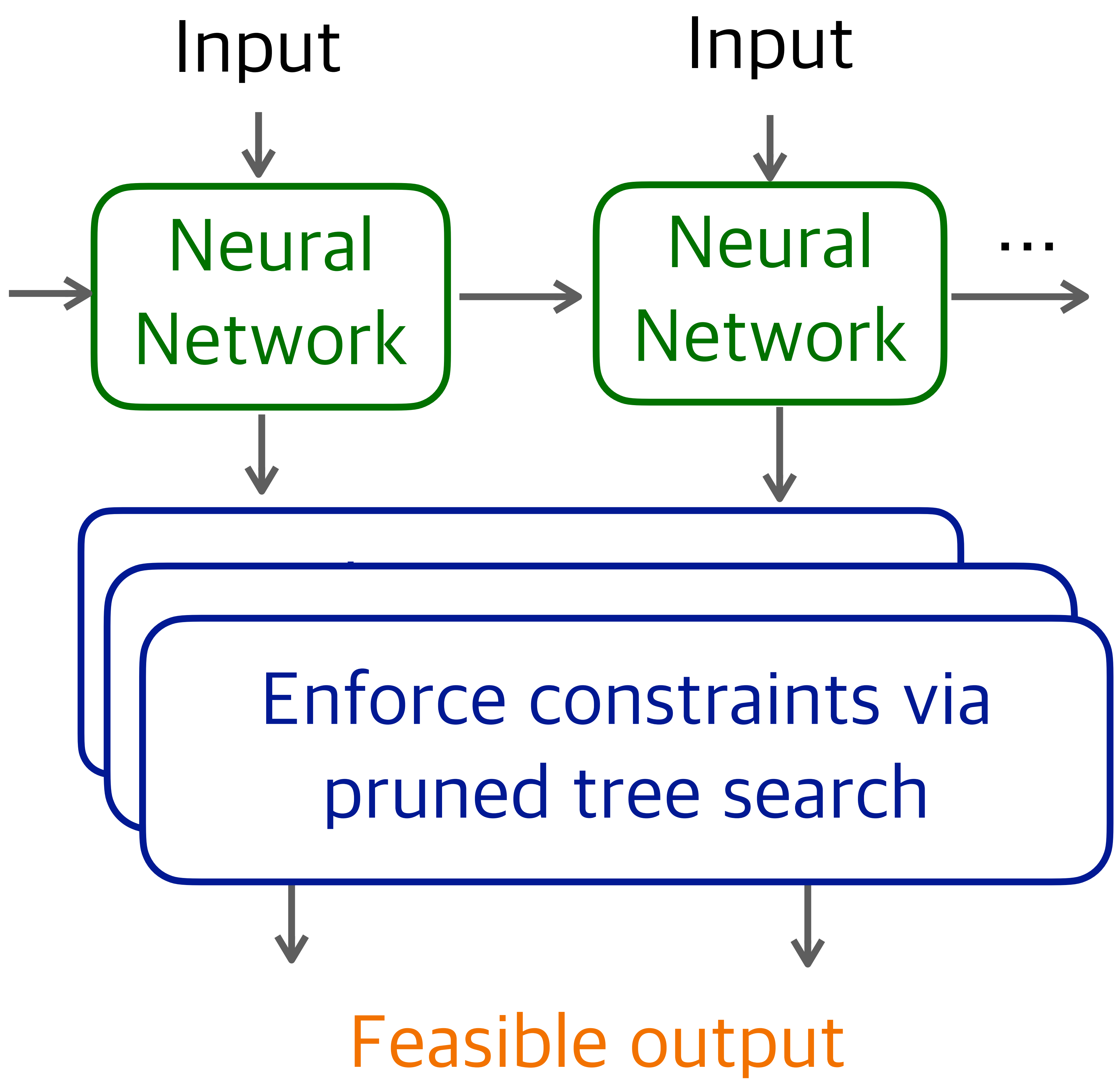
- A series of conference publications: EMNLP2020, UAI2021, JMLR2022, AAAI2023, IJCAI2024.

Integrate reasoning with learning to ensure <u>constraint satisfaction</u> for structured prediction



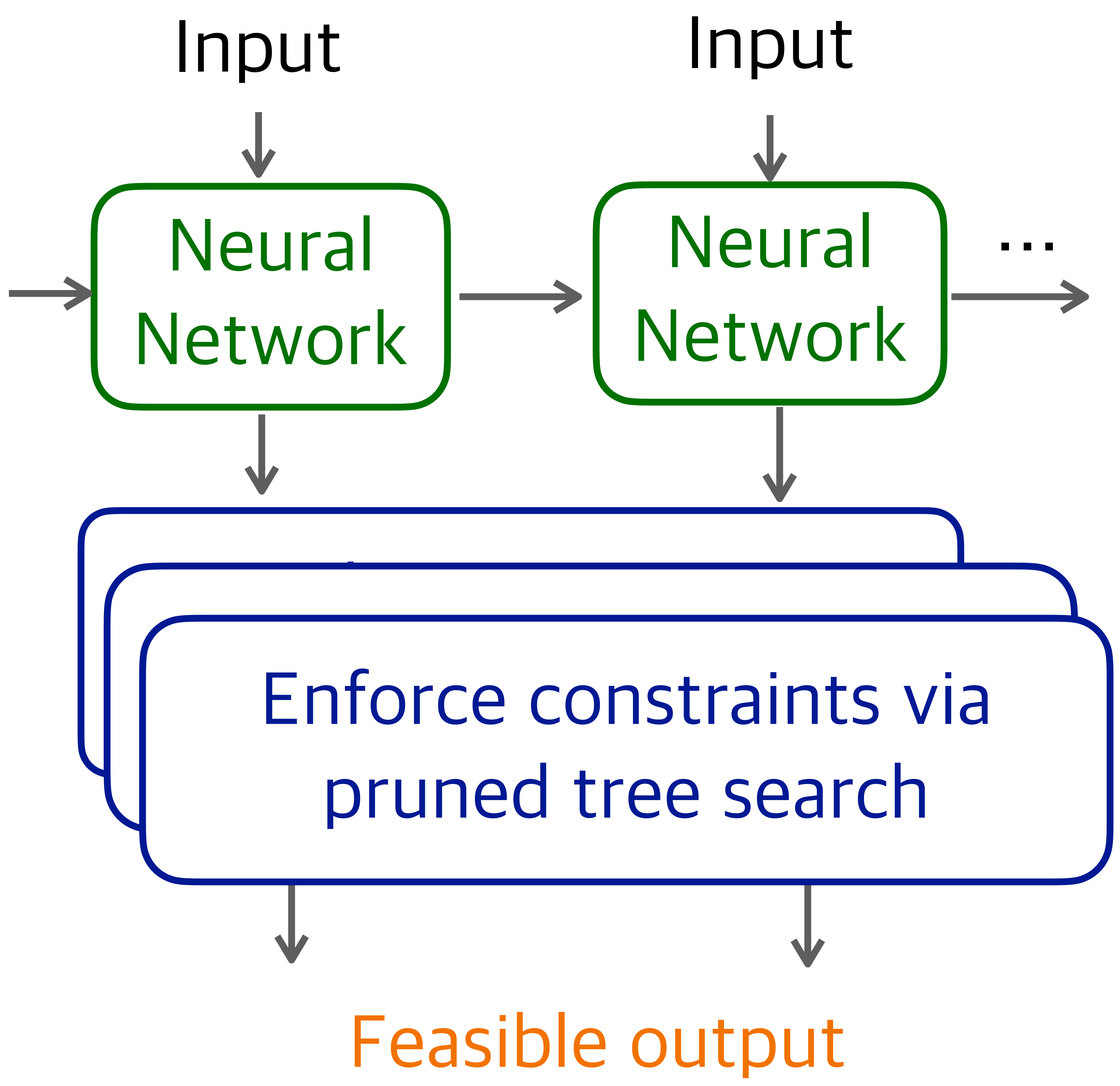
- A series of conference publications: EMNLP2020, UAI2021, JMLR2022, AAAI2023, IJCAI2024.

Integrate reasoning with learning to ensure constraint satisfaction for structured prediction



- A series of conference publications: EMNLP2020, UAI2021, JMLR2022, AAAI2023, IJCAI2024.

Integrate reasoning with learning to ensure constraint satisfaction for structured prediction



- For route planning,
  - Our method generates **100% valid routes**,
  - Pure ML baselines (i.e., Transformer) produce **<1% valid routes**.

- A series of conference publications: EMNLP2020, UAI2021, JMLR2022, AAAI2023, IJCAI2024.

Integrate reasoning with learning to accelerate scientific discovery

- **Task**: Learning physical knowledge in closed-form from 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

Best expression is $x_1 \times x_2 - x_3/x_4$.

- A series of conference publications: ECML2023, AAAI2024, IJCAI2024, AAAI2025.

Integrate reasoning with learning to accelerate scientific discovery

- **Task**: Learning physical knowledge in closed-form from data.

- **Gap**: Pure MLs struggle with **<4 variables setting**, because the search space grows exponentially.

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

Best expression is $x_1 \times x_2 - x_3/x_4$.

- A series of conference publications: ECML2023, AAAI2024, IJCAI2024, AAAI2025.

Integrate reasoning with learning to accelerate scientific discovery

- **Task**: Learning physical knowledge in closed-form from data.
- **Gap**: Pure MLs struggle with **<4 variables setting**, because the search space grows exponentially.
- **Our Solution**:
  - Combine ML with scientific approach-inspired reasoning.

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

Best expression is $x_1 \times x_2 - x_3/x_4$.

- A series of conference publications: ECML2023, AAAI2024, IJCAI2024, AAAI2025.

Integrate reasoning with learning to accelerate scientific discovery

- **Task**: Learning physical knowledge in closed-form from data.
- **Gap**: Pure MLs struggle with **<4 variables setting**, because the search space grows exponentially.
- **Our Solution**:
  - Combine ML with scientific approach-inspired reasoning.
- Our method discovers scientific equations involving **50 variables.**
- A series of conference publications: ECML2023, AAAI2024, IJCAI2024, AAAI2025.

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

Best expression is $x_1 \times x_2 - x_3/x_4$.

Design principle of the integrated system: For CNF-SAT logical constraints satisfying "extreme conditions"

for CNF-SAT Logical constraint, i.e.,

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

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

Design principle of the integrated system:
For CNF-SAT logical constraints satisfying "extreme conditions"

Input $x_1, \ldots, x_n$

- Learn from data

Markov Random Fields

for CNF-SAT Logical constraint, i.e.,

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

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

Design principle of the integrated system:
For CNF-SAT logical constraints satisfying "extreme conditions"

Input $x_1, \dots, x_n$

↓

- Learn from data — **Markov Random Fields**

↓

- Satisfy CNF-SAT logical constraints — **Sampling through Lovasz Local Lemma Theory**

for CNF-SAT Logical constraint, i.e.,

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

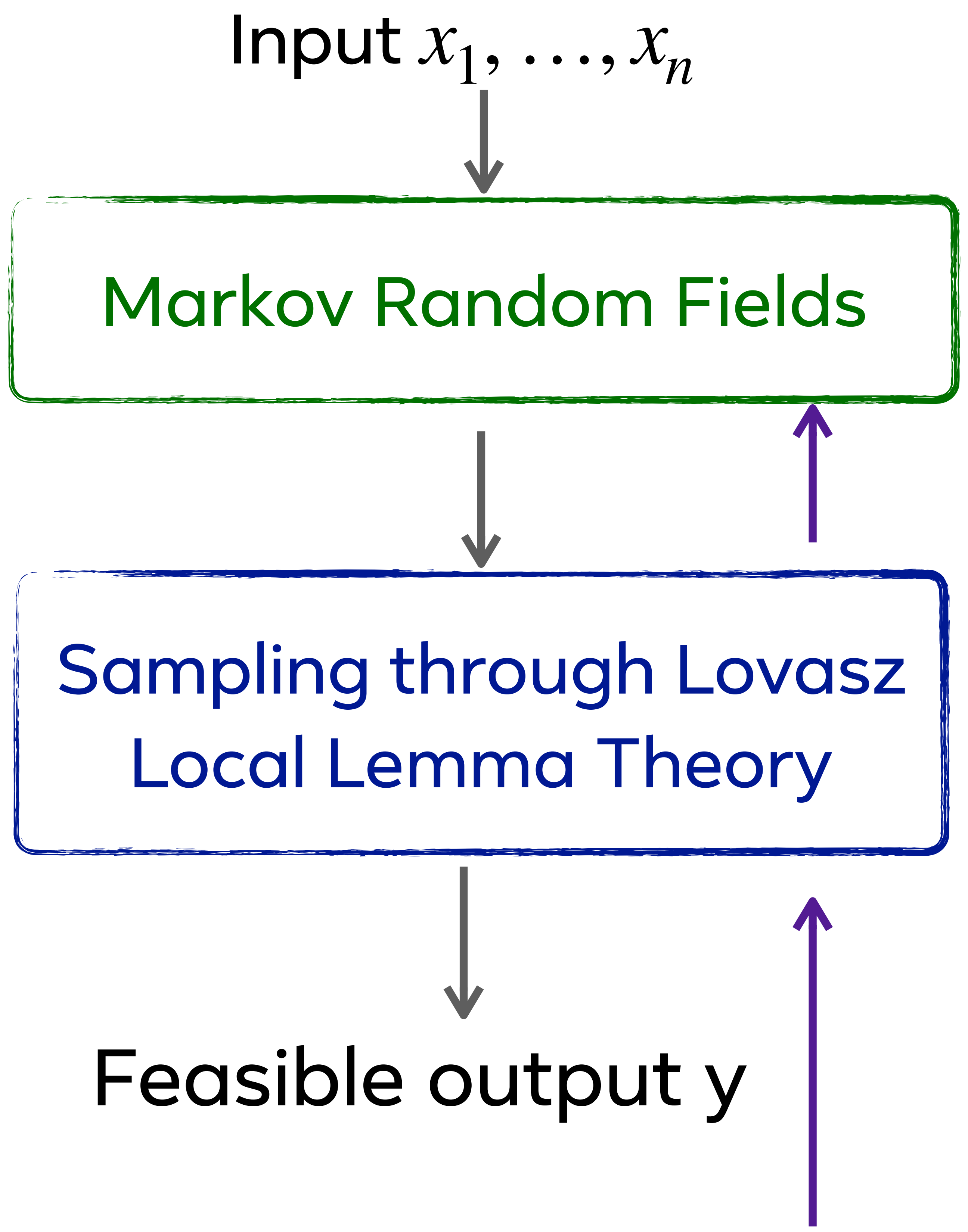
↓

Feasible output y

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

Design principle of the integrated system:
For CNF-SAT logical constraints satisfying “extreme conditions”

- Learn from data
- Satisfy CNF-SAT logical constraints
- Differentiable



Back-propagate the gradient

for CNF-SAT Logical constraint, i.e.,

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

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

Background on Lovasz Local Lemma

Heng Guo and Mark Jerrum and Jingcheng Liu. Uniform Sampling Through the Lovasz Local Lemma. In Journal of ACM, 2019.

Background on Lovasz Local Lemma

- In 1973, Erdos and Lovasz give the existence proof:

there exists a positive probability that none of a series of bad events occur, as long as these events are mostly independent from one another and are not too likely individually.

Heng Guo and Mark Jerrum and Jingcheng Liu. Uniform Sampling Through the Lovasz Local Lemma. In Journal of ACM, 2019.

Background on Lovasz Local Lemma

- In 1973, Erdos and Lovasz give the existence proof:

there exists a positive probability that none of a series of bad events occur, as long as these events are mostly independent from one another and are not too likely individually.

- In 2010, Moser and Tardos proposed Algorithmic-LLL:

which is an randomized algorithm to find solutions without breaking any bad events

The Gödel Prize 2020 - Laudation

The 2020 Gödel Prize is awarded to **Robin A. Moser** and **Gábor Tardos** for their algorithmic version of the Lovász Local Lemma in the paper:
"A constructive proof of the general Lovász Local Lemma," Journal of the ACM 57(2): 11:1-11:15 (2010).

Heng Guo and Mark Jerrum and Jingcheng Liu. Uniform Sampling Through the Lovasz Local Lemma. In Journal of ACM, 2019.

Background on Lovasz Local Lemma

- In 1973, Erdos and Lovasz give the existence proof:

there exists a positive probability that none of a series of bad events occur, as long as these events are mostly independent from one another and are not too likely individually.

- In 2010, Moser and Tardos proposed Algorithmic-LLL:

which is an randomized algorithm to find solutions without breaking any bad events

The Gödel Prize 2020 - Laudation

The 2020 Gödel Prize is awarded to **Robin A. Moser** and **Gábor Tardos** for their algorithmic version of the Lovász Local Lemma in the paper:

"A constructive proof of the general Lovász Local Lemma," Journal of the ACM 57(2): 11:1-11:15 (2010).

- In 2019 (JACM), Heng Guo, Mark Jerrum et al.:

The probability distribution of Algorithmic-LLL and the necessary condition

Heng Guo and Mark Jerrum and Jingcheng Liu. Uniform Sampling Through the Lovasz Local Lemma. In Journal of ACM, 2019.

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

Sample X_i from $P(X_i)$ →

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

c_2 is violated →

X_1	X_2	X_3
1	0	0

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

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 $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

X_1	X_2	X_3
1	0	0
0	0	1

c_1 is violated → (row 2)

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

Resample X_1, X_2 from $P(X_1), P(X_2)$ →

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

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

All constraints are satisfied! →

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

Sampling through Lovasz Local Lemma

Inputs: Discrete variables $X = [X_1, X_2, X_3]$, 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 $\prod_{i=1}^{3} P(x_i \mid C)$.

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

All constraints are satisfied!

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

Implementing Sampling through Lovasz Local Lemma as several **Fully Differentiable** Neural Network Layers

Input:
Discrete variables X_1, X_2, X_3,
with $X_1 \in \{0,1\}$.
Marginal distributions
$P(X_1), P(X_2), P(X_3)$.
Constraints

$C = c_1 \wedge c_2,$
$c_1 = X_1 \vee X_2,$
$c_2 = \neg X_1 \vee X_3$

$$\mathrm{W} = \begin{bmatrix} [1 & 0 & 0] & [0 & 1 & 0] \\ [-1 & 0 & 0] & [0 & 0 & 1] \end{bmatrix}$$

$$b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\mathrm{V} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

1. Initialize

$$x_i = 1[u_i > P(X_i)]$$

2. Extract violated constraints

$$Z = W \otimes x + b$$

$$S_j = 1 - \max_{1 \le k \le K} Z_{jk}$$

3. Variables to be resamples

$$A_i = 1\left[\sum_{j=1}^{L} S_j j V_{ji} \ge 1\right]$$

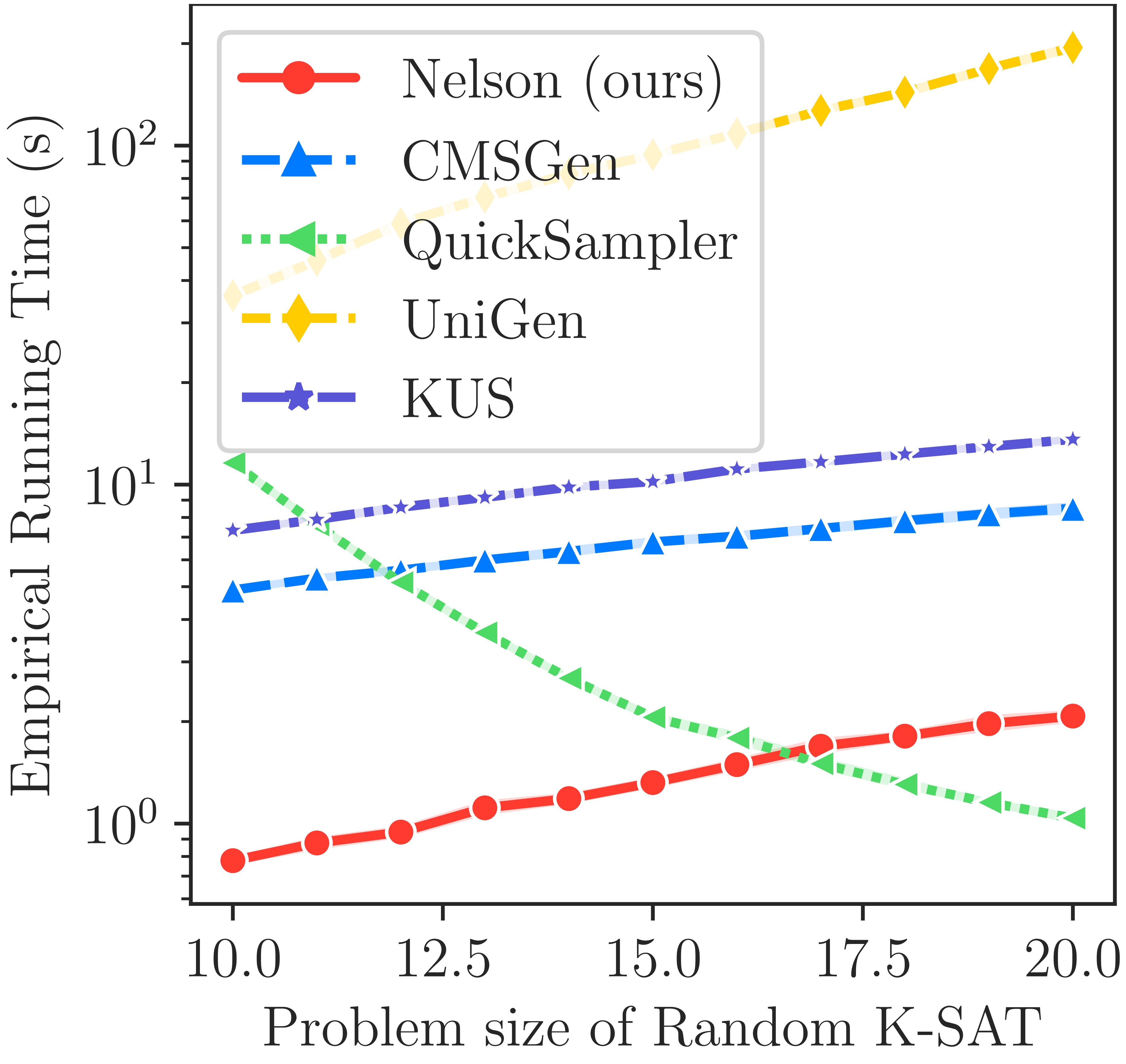
4. Resample variables

$$x = (1 - \mathrm{A}) * \mathrm{x} + \mathrm{A} * 1[u_i > P(X_i)]$$

↑
Fully Differentiable

Implemented as a series of matrix computations

Experiments: Our Nelson draw samples faster than baselines with constraint satisfaction

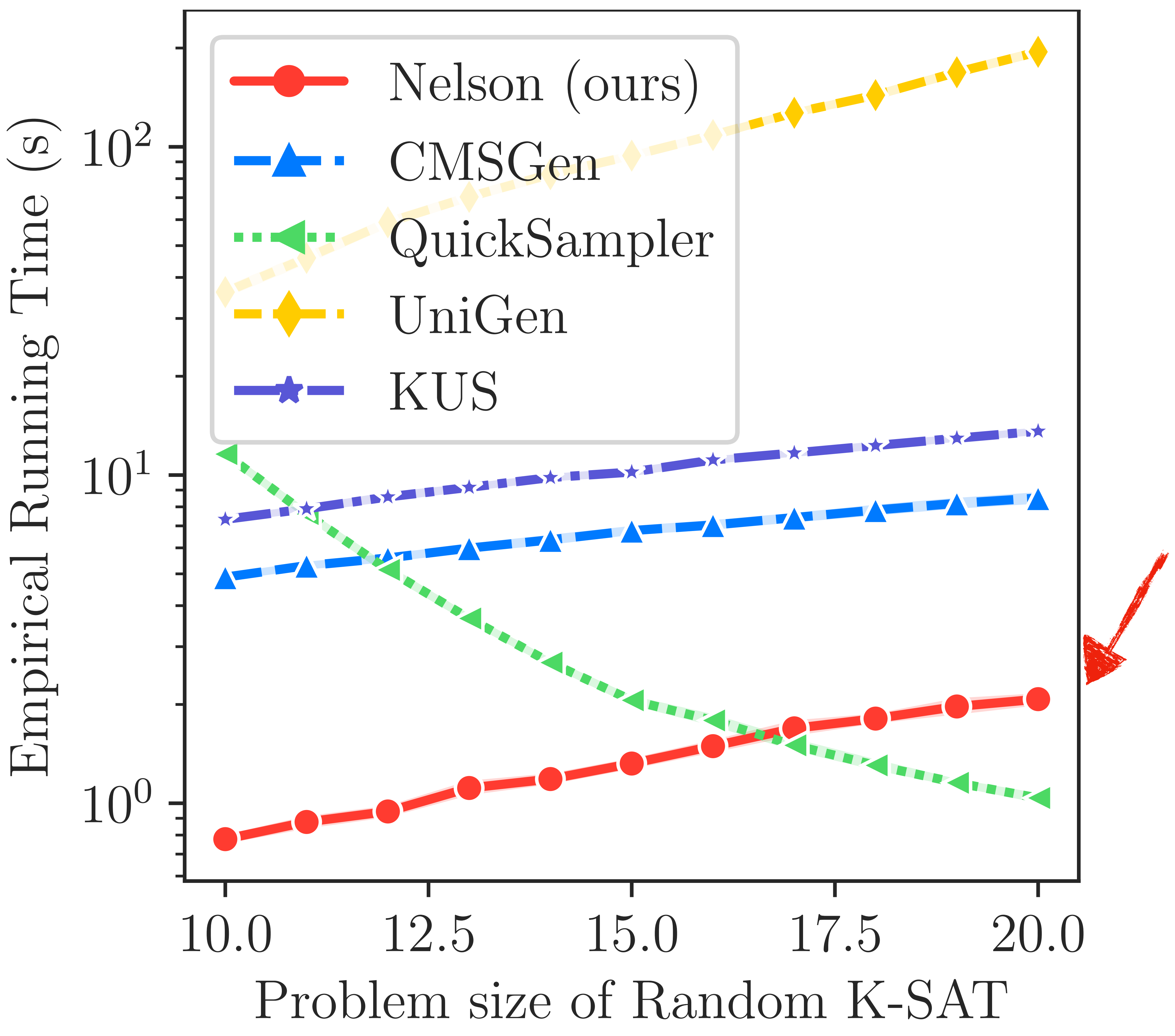


Task: sample feasible output from the model.

UniGen: https://github.com/meelgroup/unigen
CMSGen: https://github.com/meelgroup/cmsgen

KUS sampler: https://github.com/meelgroup/KUS
QuickSampler: https://github.com/RafaelTupynamba/quicksampler/

Experiments: Our Nelson draw samples faster than baselines with constraint satisfaction



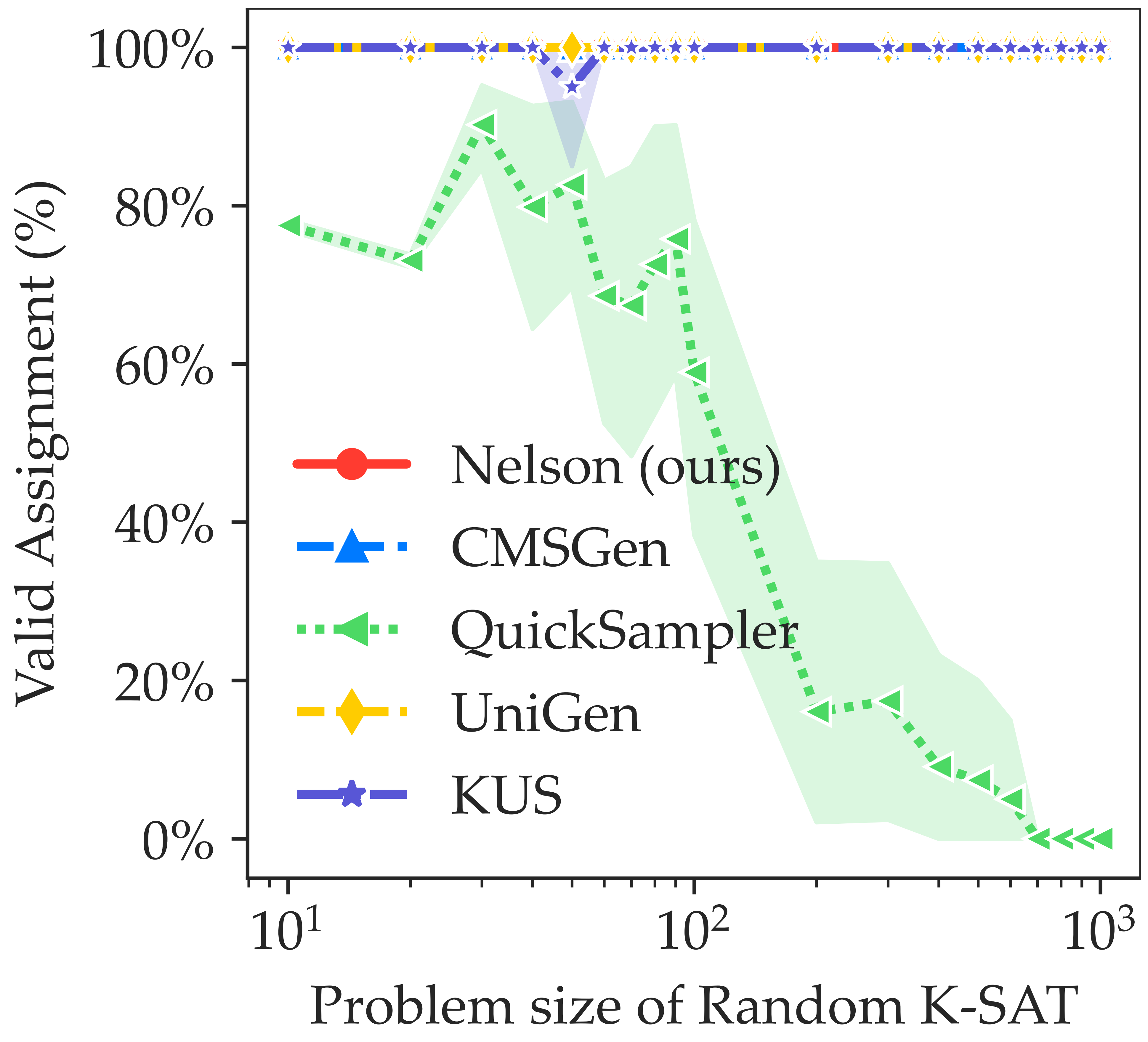
Task: sample feasible output from the model.

Our Nelson draw samples faster than existing methods.

UniGen: https://github.com/meelgroup/unigen
CMSGen: https://github.com/meelgroup/cmsgen

KUS sampler: https://github.com/meelgroup/KUS
QuickSampler: https://github.com/RafaelTupynamba/quicksampler/

Experiments: Our Nelson draw samples faster than baselines with constraint satisfaction

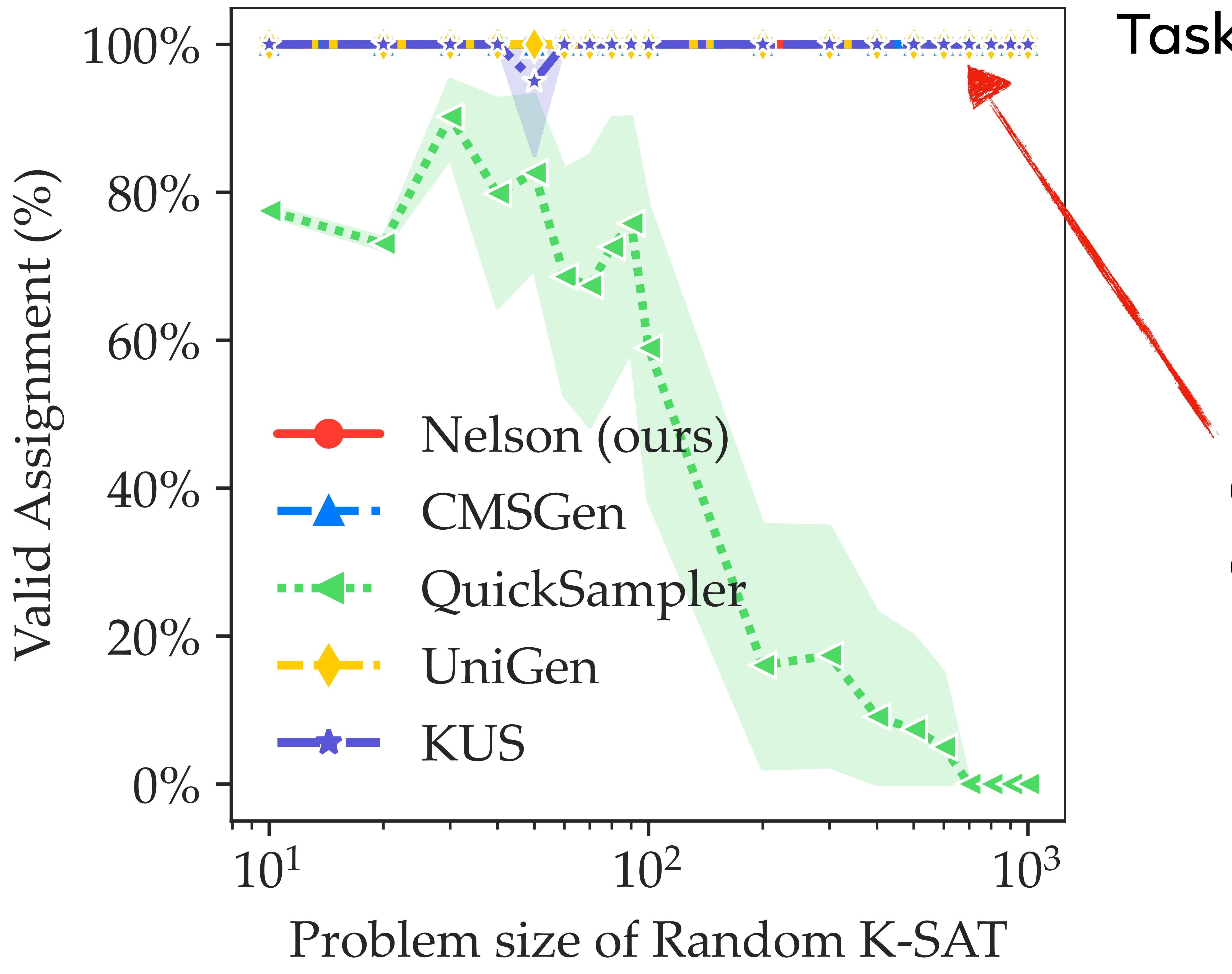


Task: sample feasible output from the model.

UniGen: https://github.com/meelgroup/unigen
CMSGen: https://github.com/meelgroup/cmsgen

KUS sampler: https://github.com/meelgroup/KUS
QuickSampler: https://github.com/RafaelTupynamba/quicksampler/

Experiments: Our Nelson draw samples faster than baselines with constraint satisfaction



Task: sample feasible output from the model.

Our Nelson always sample feasible output from the model.

UniGen: https://github.com/meelgroup/unigen
CMSGen: https://github.com/meelgroup/cmsgen

KUS sampler: https://github.com/meelgroup/KUS
QuickSampler: https://github.com/RafaelTupynamba/quicksampler/

Experiment: Our Nelson estimates the gradient more accurately

Problem size	(a) Training Time Per Epoch (Mins) (↓)								
	NELSON	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
20	**0.15**	134.50	3.04	T.O.	0.26	0.90	0.30	2.12	1.72
30	**0.19**	1102.95	6.62	T.O.	0.28	2.24	0.32	4.72	2.77
40	**0.23**	T.O.	33.70	T.O.	0.31	19.77	0.39	9.38	3.93
50	**0.24**	T.O.	909.18	T.O.	0.33	1532.22	0.37	13.29	5.27
500	**5.99**	T.O.	T.O.	T.O.	34.17	T.O.	T.O.	T.O.	221.83
1000	**34.01**	T.O.	T.O.	T.O.	177.39	T.O.	T.O.	T.O.	854.59
	(b) Validness of CNF Assignments (%) (↑)								
$10-50$	**100**	**100**	**100**	**100**	**100**	**100**	82.65	**100**	90.58
500	**100**	T.O.	T.O.	T.O.	**100**	T.O.	7.42	**100**	54.27
1000	**100**	T.O.	T.O.	T.O.	**100**	T.O.	0.00	**100**	33.91
	(c) Approximation Error of Gradient (↓)								
10	**0.10**	0.21	0.12	3.58	3.96	4.08	3.93	4.16	0.69
12	**0.14**	0.19	0.16	5.58	5.50	5.49	5.55	5.48	0.75
14	**0.15**	0.25	0.19	T.O.	6.55	6.24	7.79	6.34	1.30
16	0.16	0.25	**0.15**	T.O.	9.08	9.05	9.35	9.03	1.67
18	**0.18**	0.30	0.23	T.O.	10.44	10.30	11.73	10.20	1.90

Experiment: Our Nelson estimates the gradient more accurately

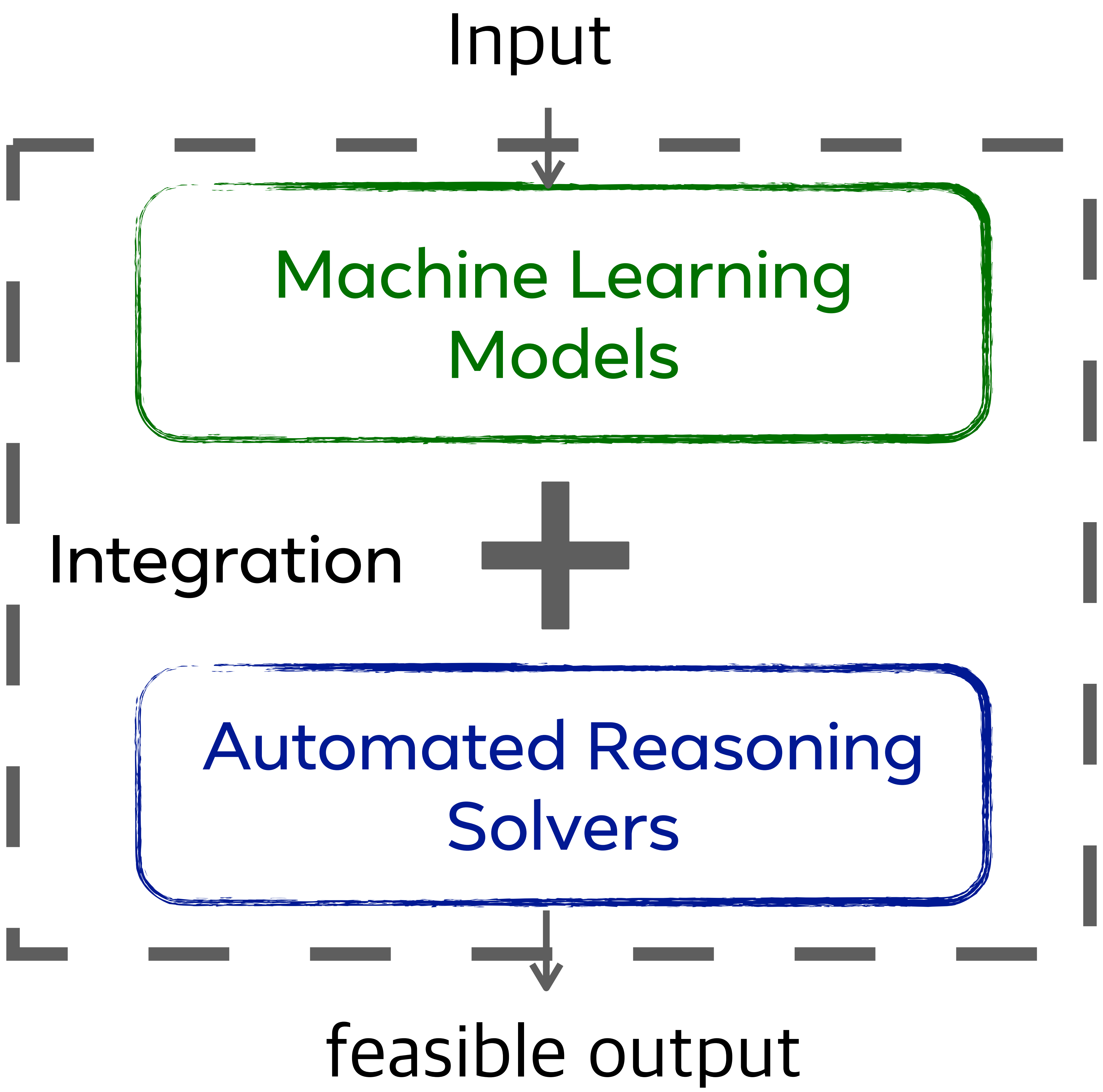
Problem size	(a) Training Time Per Epoch (Mins) (↓)								
	NELSON	XOR	WAPS	WeightGen	CMSGen	KUS	QuickSampler	Unigen	Gibbs
10	**0.13**	26.30	1.75	0.64	0.22	0.72	0.40	0.66	0.86
20	**0.15**	134.50	3.04	T.O.	0.26	0.90	0.30	2.12	1.72
30	**0.19**	1102.95	6.62	T.O.	0.28	2.24	0.32	4.72	2.77
40	**0.23**	T.O.	33.70	T.O.	0.31	19.77	0.39	9.38	3.93
50	**0.24**	T.O.	909.18	T.O.	0.33	1532.22	0.37	13.29	5.27
500	**5.99**	T.O.	T.O.	T.O.	34.17	T.O.	T.O.	T.O.	221.83
1000	**34.01**	T.O.	T.O.	T.O.	177.39	T.O.	T.O.	T.O.	854.59
	(b) Validness of CNF Assignments (%) (↑)								
10 – 50	**100**	**100**	**100**	**100**	**100**	**100**	82.65	**100**	90.58
500	**100**	T.O.	T.O.	T.O.	**100**	T.O.	7.42	**100**	54.27
1000	**100**	T.O.	T.O.	T.O.	**100**	T.O.	0.00	**100**	33.91
	(c) Approximation Error of Gradient (↓)								
10	**0.10**	0.21	0.12	3.58	3.96	4.08	3.93	4.16	0.69
12	**0.14**	0.19	0.16	5.58	5.50	5.49	5.55	5.48	0.75
14	**0.15**	0.25	0.19	T.O.	6.55	6.24	7.79	6.34	1.30
16	0.16	0.25	**0.15**	T.O.	9.08	9.05	9.35	9.03	1.67
18	**0.18**	0.30	0.23	T.O.	10.44	10.30	11.73	10.20	1.90

Our Nelson estimates the gradient more accurately

Takeaway

The benefits are:

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



Reference

1. **Nan Jiang**, Md. Nasim, and Yexiang Xue. Active symbolic discovery of ordinary differential equations via phase portrait sketching. In AAAI, 2025.
2. **Nan Jiang**, Md Nasim, and Yexiang Xue. Vertical symbolic regression via deep policy gradient. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI), 2024.
3. Jinzhao Li, **Nan Jiang**, and Yexiang Xue. Solving satisfiability modulo counting for symbolic and statistical ai integration with provable guarantees. In AAAI, 2024.
4. **Nan Jiang** and Yexiang Xue. Racing control variable genetic programming for symbolic regression. In AAAI, 2024
5. **Nan Jiang**, Jinzhao Li, and Yexiang Xue. A tighter convergence proof of reverse experience replay. Reinforcement Learning Journal, 1:470-480, 2024.
6. Md. Nasim, **Nan Jiang**, Yexiang Xue. Computational Approaches to Scientific Discovery. Lecture Notes in Computer Science. Springer, 2024. Book Chapter.
7. **Nan Jiang** and Yexiang Xue. Symbolic regression via control variable genetic programming. In ECML/PKDD, volume 14172, pages 178-195, 2023.
8. **Nan Jiang**, Yi Gu, and Yexiang Xue. Learning Markov random fields for combinatorial structures via sampling through lovász local lemma. In AAAI, 2023.
9. Maxwell J. Jacobson, Case Q. Wright, **Nan Jiang**, Gustavo Rodriguez-Rivera, and Yexiang Xue. Task detection in continual learning via familiarity autoencoders. In roceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 1-8, 2022.
10. **Nan Jiang**, Chen Luo, Vihan Lakshman, Yesh Dattatreya, and Yexiang Xue. Massive text normalization via an efficient randomized algorithm. In The ACM Web Conference (WWW), pages 2946-2956, 2022.
11. **Nan Jiang**, Maosen Zhang, Willem-Jan van Hoeve, and Yexiang Xue. Constraint reasoning embedded structured prediction. Journal of Machine Learning Research, 23:345:1-345:40, 2022.
12. **Nan Jiang**$_*$, Fan Ding$_*$, Jianzhu Ma, Jian Peng, Jinbo Xu, and Yexiang Xue. PALM: probabilistic area loss minimization for protein sequence alignment. In UAI, volume 161, pages 1100-1109, 2021.
13. Libin Shi, Wenge Rong, Shijie Zhou, **Nan Jiang**, and Zhang Xiong. A dual channel class hierarchy-based recurrent language modeling. Neurocomputing, 2020.
14. Maosen Zhang, **Nan Jiang**, Lei Li, and Yexiang Xue. Constraint satisfaction driven natural language generation: A tree search embedded MCMC approach. In EMNLP Finding, 2020.