Large Neighbourhood Search meets Iterative Neural Constraint Heuristics

Yudong Will Xu, Wenhao Li, Scott Sanner, Elias B. Khalil

UNIVERSITY OF TORONTO

CPAIOR 2026 Rabat

Introduction

Reasoning under constraints is a crucial challenge in many domains.

Machine learning techniques have been explored as fast heuristic solvers.

Figure from Khalil (2024) [1]

Learning to Solve Constraint Optimization

2		1	9			7		4
	5	3		1	4	6	9	
			5				8	3
1	3	7	2	4		5		9
			3			4	7	8
	8			7		3	2	
9	4	6		5				
		8	4	6			1	
5		2		3	7	9		6

2	6	1	9	8	3	7	5	4
8	5	3	7	1	4	6	9	2
7	9	4	5	2	6	1	8	3
1	3	7	2	4	8	5	6	9
6	2	5	3	9	1	4	7	8
4	8	9	6	7	5	3	2	1
9	4	6	1	5	2	8	3	7
3	7	8	4	6	9	2	1	5
5	1	2	8	3	7	9	4	6

Learning to Solve Constraint Optimization

2		1	9			7		4
	5	3		1	4	6	9	
			5				8	3
1	3	7	2	4		5		9
			3			4	7	8
	8			7		3	2	
9	4	6		5				
		8	4	6			1	
5		2		3	7	9		6

2	6	1	9	8	3	7	5	4
8	5	3	7	1	4	6	9	2
7	9	4	5	2	6	1	8	3
1	3	7	2	4	8	5	6	9
6	2	5	3	9	1	4	7	8
4	8	9	6	7	5	3	2	1
9	4	6	1	5	2	8	3	7
3	7	8	4	6	9	2	1	5
5	1	2	8	3	7	9	4	6

One-pass Heuristic: The model predicts the full solution in a single pass.

Learning to Solve Constraint Optimization

2		1	9			7		4
	5	3		1	4	6	9	
			5				8	3
1	3	7	2	4		5		9
			3			4	7	8
	8			7		3	2	
9	4	6		5				
		8	4	6			1	
5		2		3	7	9		6

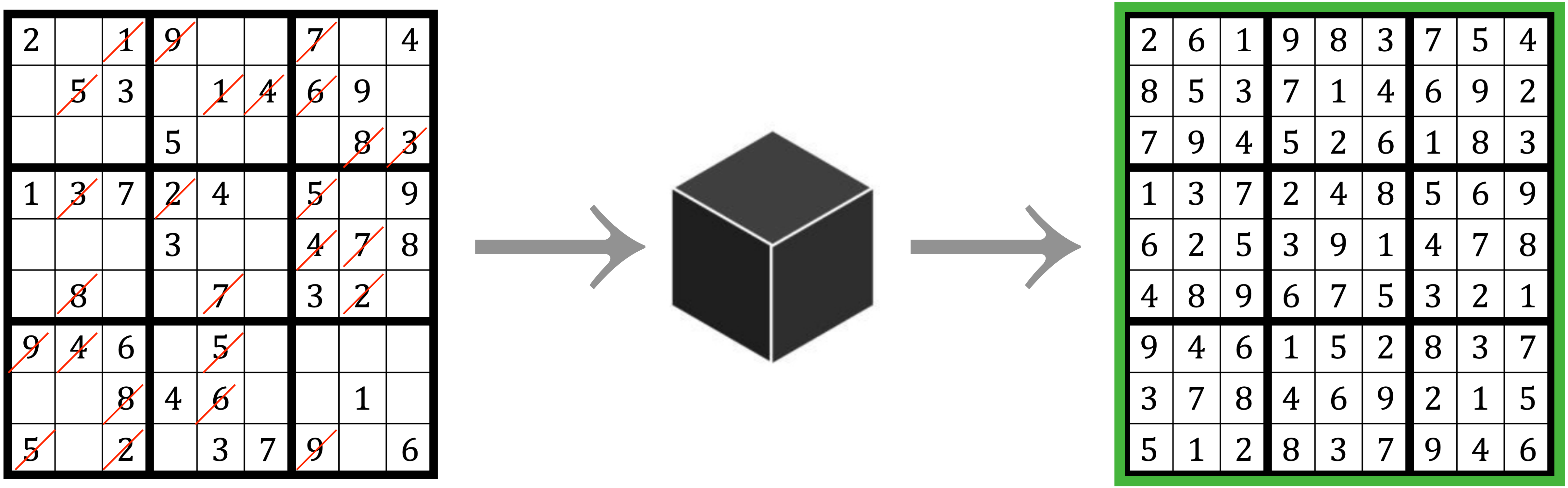
2	6	1	9	8	3	7	5	4
8	5	3	7	1	4	6	9	2
7	9	4	5	2	6	1	8	3
1	3	7	2	4	8	5	6	9
6	2	5	3	9	1	4	7	8
4	8	9	6	7	5	3	2	1
9	4	6	1	5	2	8	3	7
3	7	8	4	6	9	2	1	5
5	1	2	8	3	7	9	4	6

One-pass Heuristic: The model predicts the full solution in a single pass.

UNIVERSITY OF TORONTO

Does this make sense?

Learning to Solve Constraint Optimization



One-pass Heuristic: The model predicts the full solution in a single pass.

Learning to Solve Constraint Optimization

2								4
		3					9	
			5					
1		7		4				9
			3					8
						3		
		6						
			4				1	
				3	7			6

2	6	1	9	8	3	7	5	4
8	5	3	7	1	4	6	9	2
7	9	4	5	2	6	1	8	3
1	3	7	2	4	8	5	6	9
6	2	5	3	9	1	4	7	8
4	8	9	6	7	5	3	2	1
9	4	6	1	5	2	8	3	7
3	7	8	4	6	9	2	1	5
5	1	2	8	3	7	9	4	6

One-pass Heuristic: The model predicts the full solution in a single pass.

Learning to Solve Constraint Optimization

2								4
		3					9	
			5					
1		7		4				9
			3					8
						3		
		6						
			4				1	
				3	7			6

Constant computation!

2	6	1	9	8	3	7	5	4
8	5	3	7	1	4	6	9	2
7	9	4	5	2	6	1	8	3
1	3	7	2	4	8	5	6	9
6	2	5	3	9	1	4	7	8
4	8	9	6	7	5	3	2	1
9	4	6	1	5	2	8	3	7
3	7	8	4	6	9	2	1	5
5	1	2	8	3	7	9	4	6

One-pass Heuristic: The model predicts the full solution in a single pass.

Learning to Solve Constraint Optimization

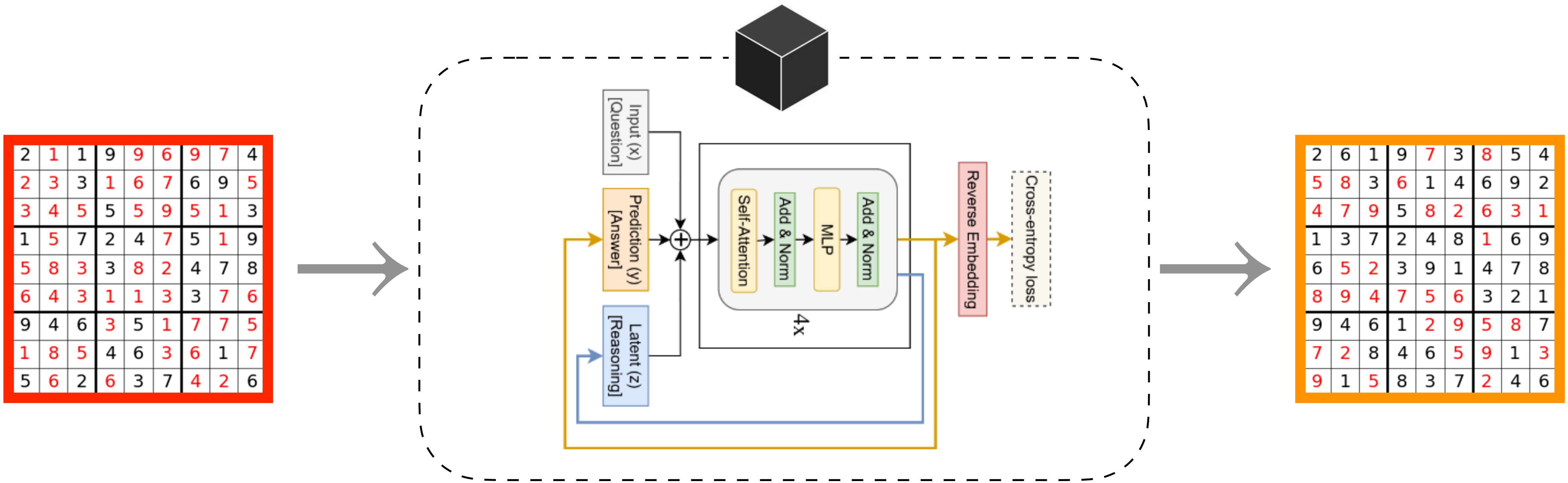
2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	6	9	5
3	4	5	5	5	9	5	1	3
1	5	7	2	4	7	5	1	9
5	8	3	3	8	2	4	7	8
6	4	3	1	1	3	3	7	6
9	4	6	3	5	1	7	7	5
1	8	5	4	6	3	6	1	7
5	6	2	6	3	7	4	2	6

2	6	1	9	7	3	8	5	4
5	8	3	6	1	4	6	9	2
4	7	9	5	8	2	6	3	1
1	3	7	2	4	8	1	6	9
6	5	2	3	9	1	4	7	8
8	9	4	7	5	6	3	2	1
9	4	6	1	2	9	5	8	7
7	2	8	4	6	5	9	1	3
9	1	5	8	3	7	2	4	6

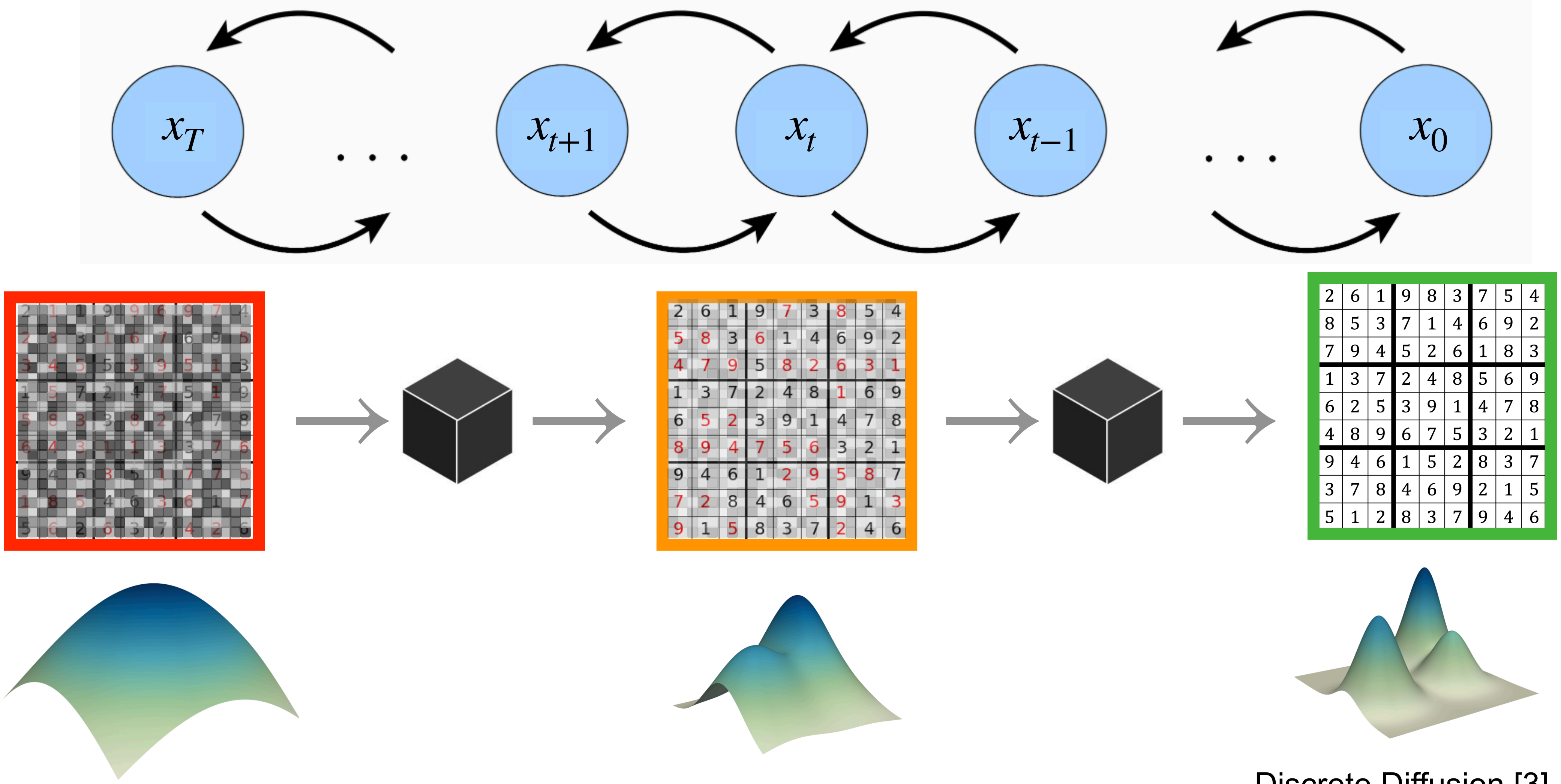
~~One-pass Heuristic: The model predicts the full solution in a single pass.~~

Iterative Heuristic: The model predicts the solution in multiple passes.

Iterative Neural Heuristics

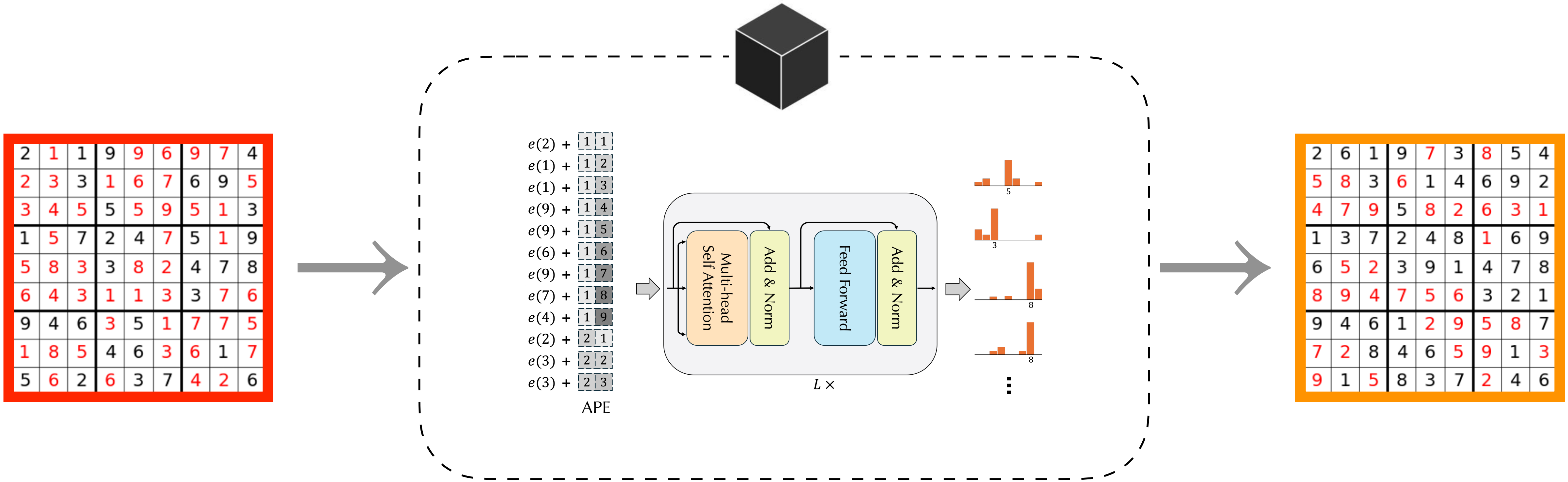


Iterative Neural Heuristics



Discrete Diffusion [3]

Iterative Neural Heuristics



UNIVERSITY OF TORONTO

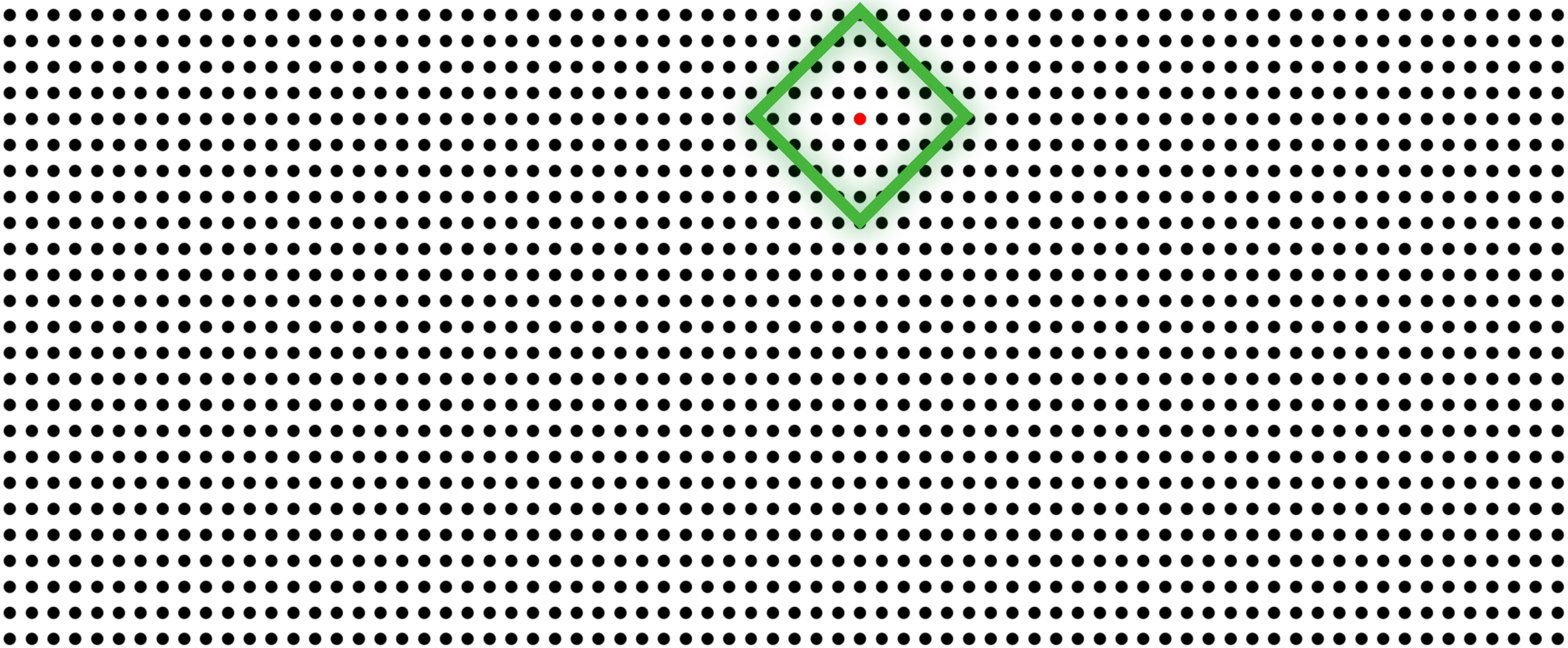
ConsFormer [4]

Why does iterative heuristics work?

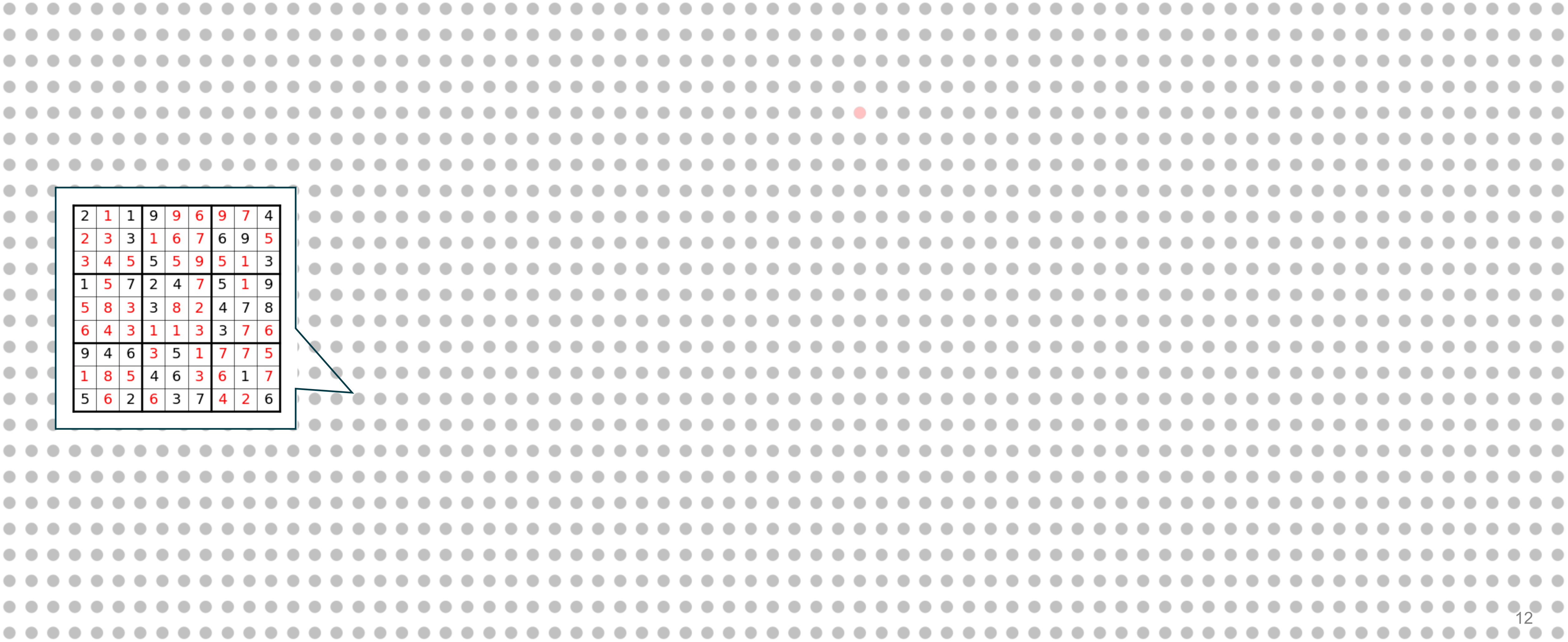
1	7	1	5	8	3	3	2	4
6	5	9	2	1	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

1	7	1	5	8	3	3	2	4
6	5	9	2	1	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	2	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

Why does iterative heuristics work?

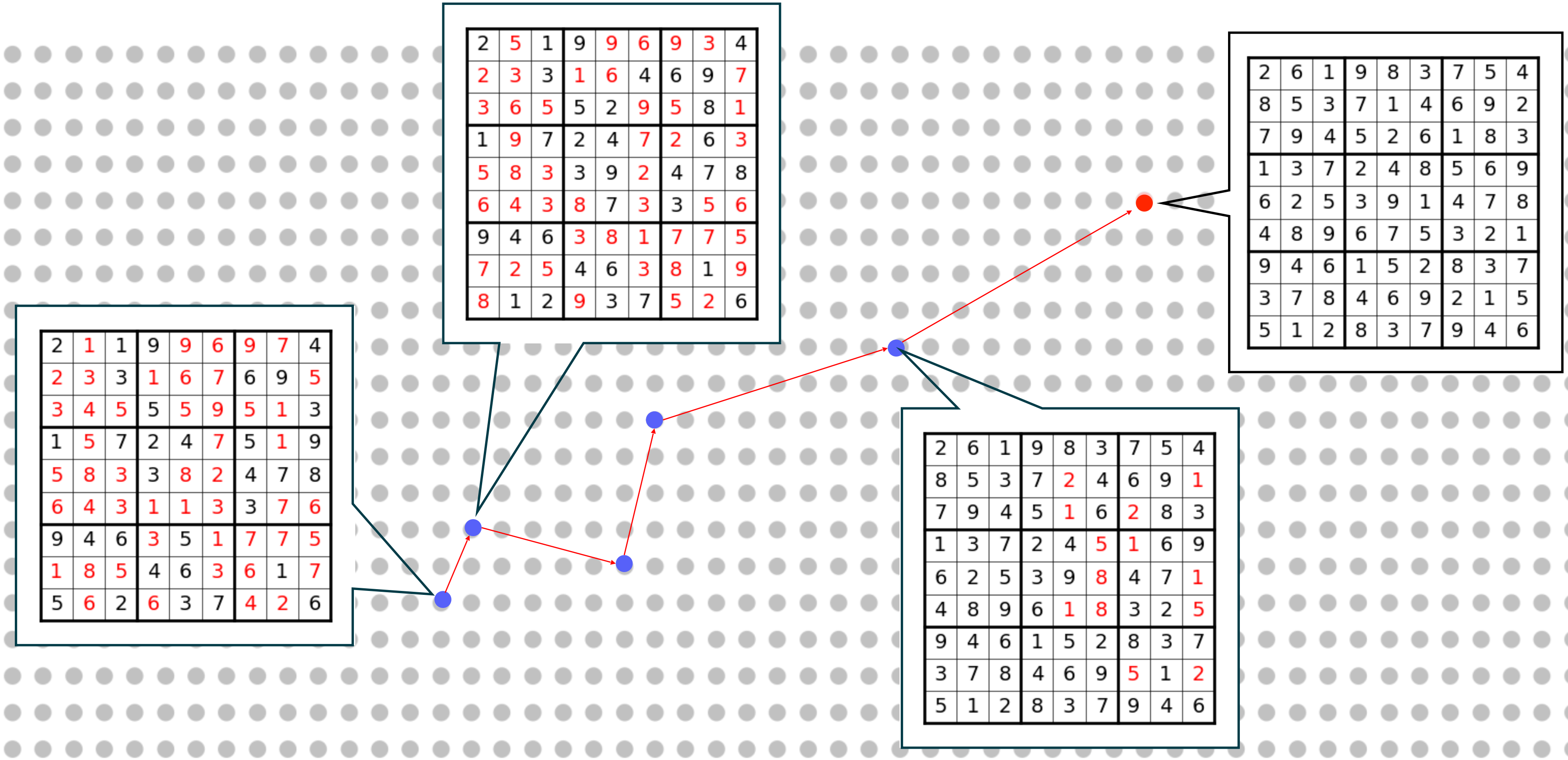


Why does iterative heuristics work?

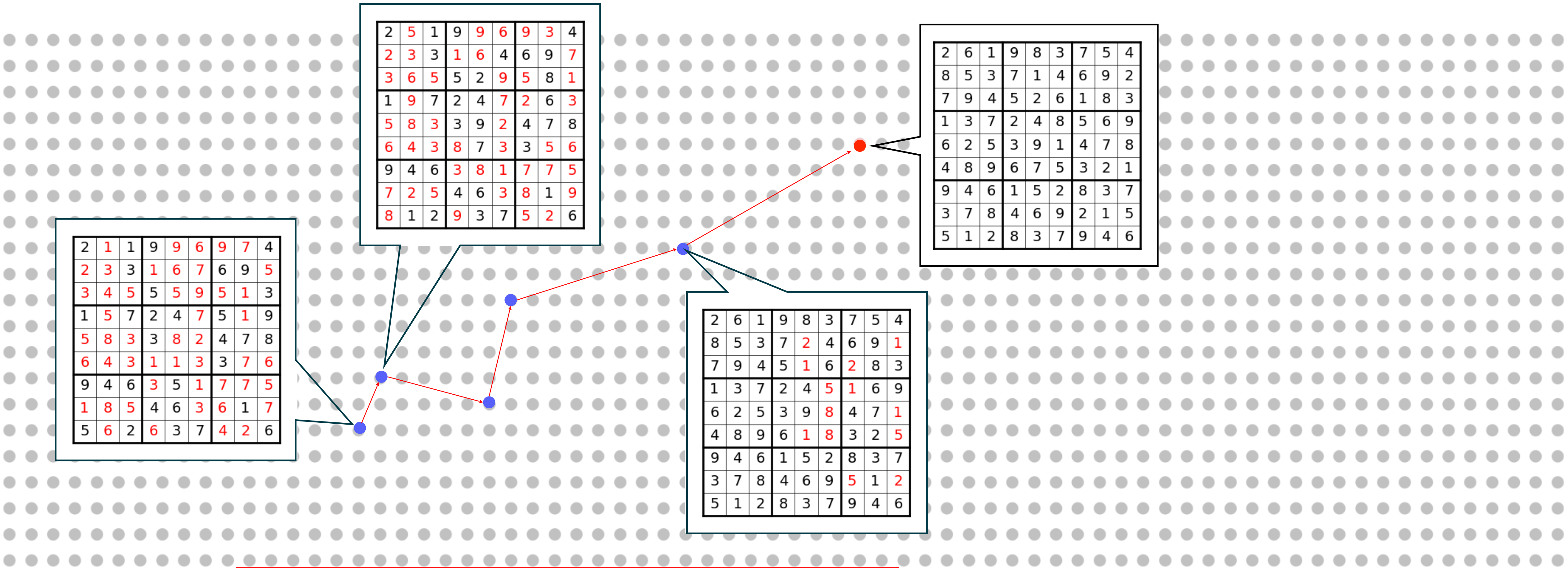


2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	6	9	5
3	4	5	5	5	9	5	1	3
1	5	7	2	4	7	5	1	9
5	8	3	3	8	2	4	7	8
6	4	3	1	1	3	3	7	6
9	4	6	3	5	1	7	7	5
1	8	5	4	6	3	6	1	7
5	6	2	6	3	7	4	2	6

Why does iterative heuristics work?



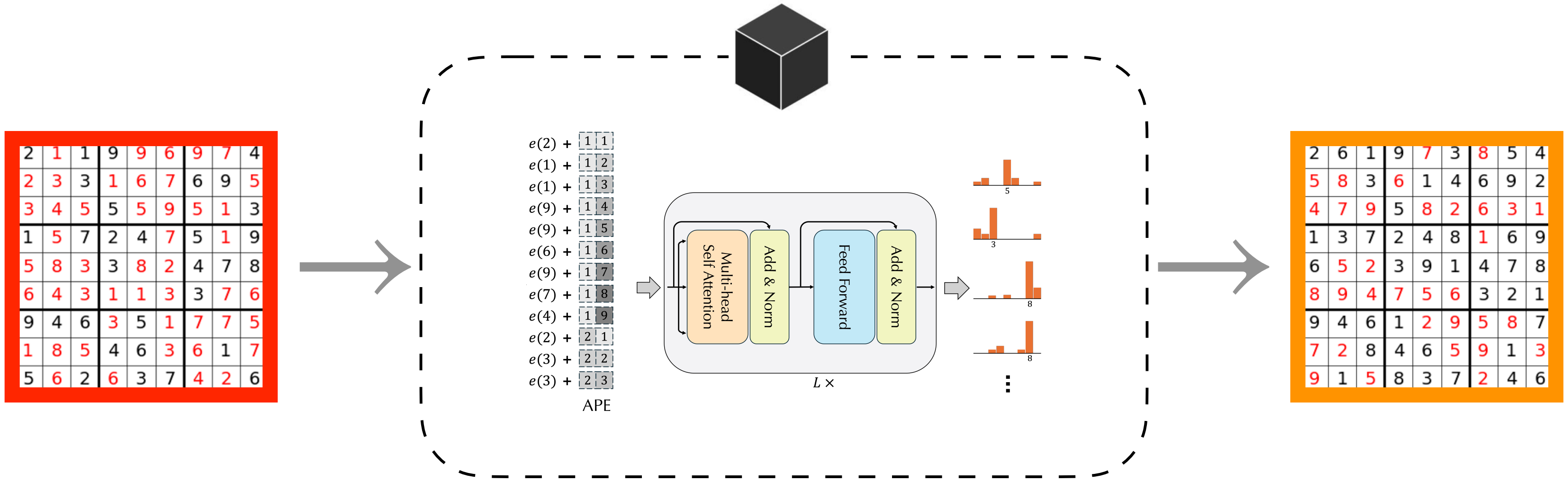
Why does iterative heuristics work?



Intuition: learning a single step of heuristic search.

Can we formalize this intuition?

Method



Iterative Neural Heuristic
(ConsFormer)

→

Neural
Large Neighbourhood Search

ConsFormer

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

ConsFormer

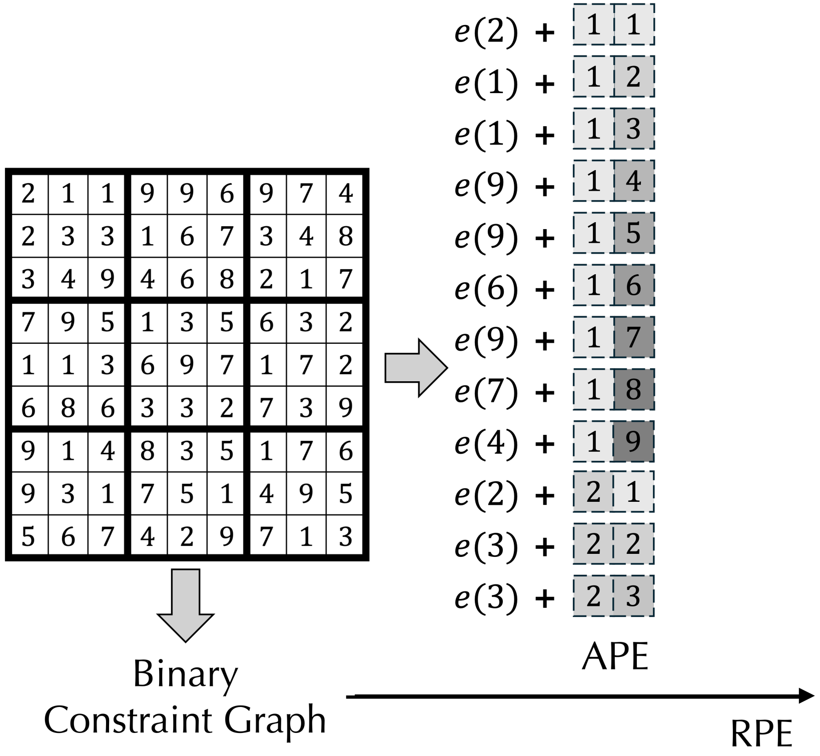
2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

$e(2)$
$e(1)$
$e(1)$
$e(9)$
$e(9)$
$e(6)$
$e(9)$
$e(7)$
$e(4)$
$e(2)$
$e(3)$
$e(3)$

Variable assignments as tokens

→ Token embedding for discrete domains

ConsFormer



Variable assignments as tokens

→ Token embedding for discrete domains

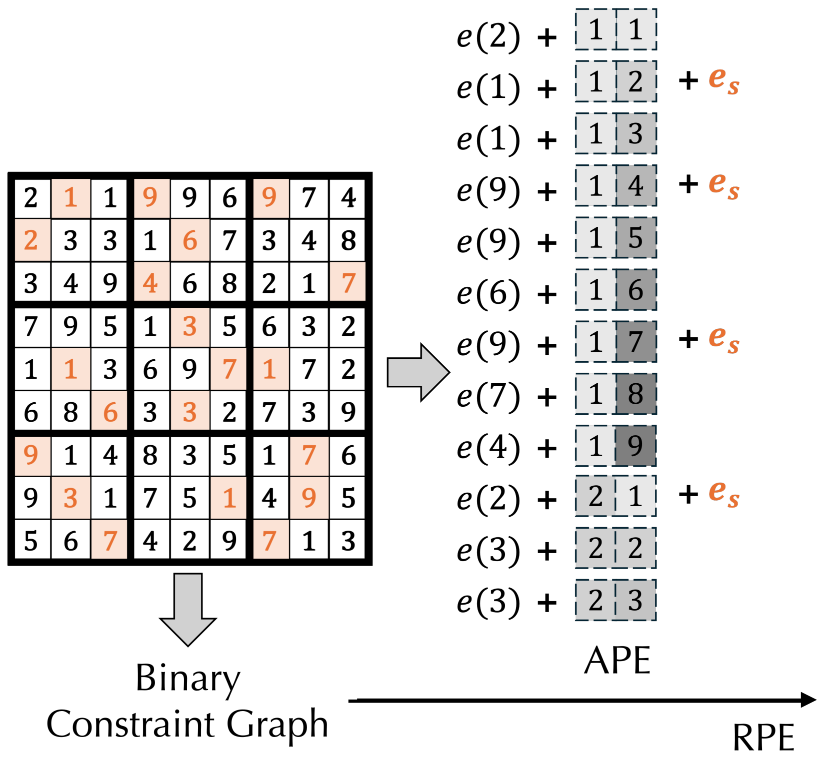
Variables indices

→ Absolute Positional Encodings

Variables under the scope of the same constraints

→ Relative Positional Encodings

ConsFormer



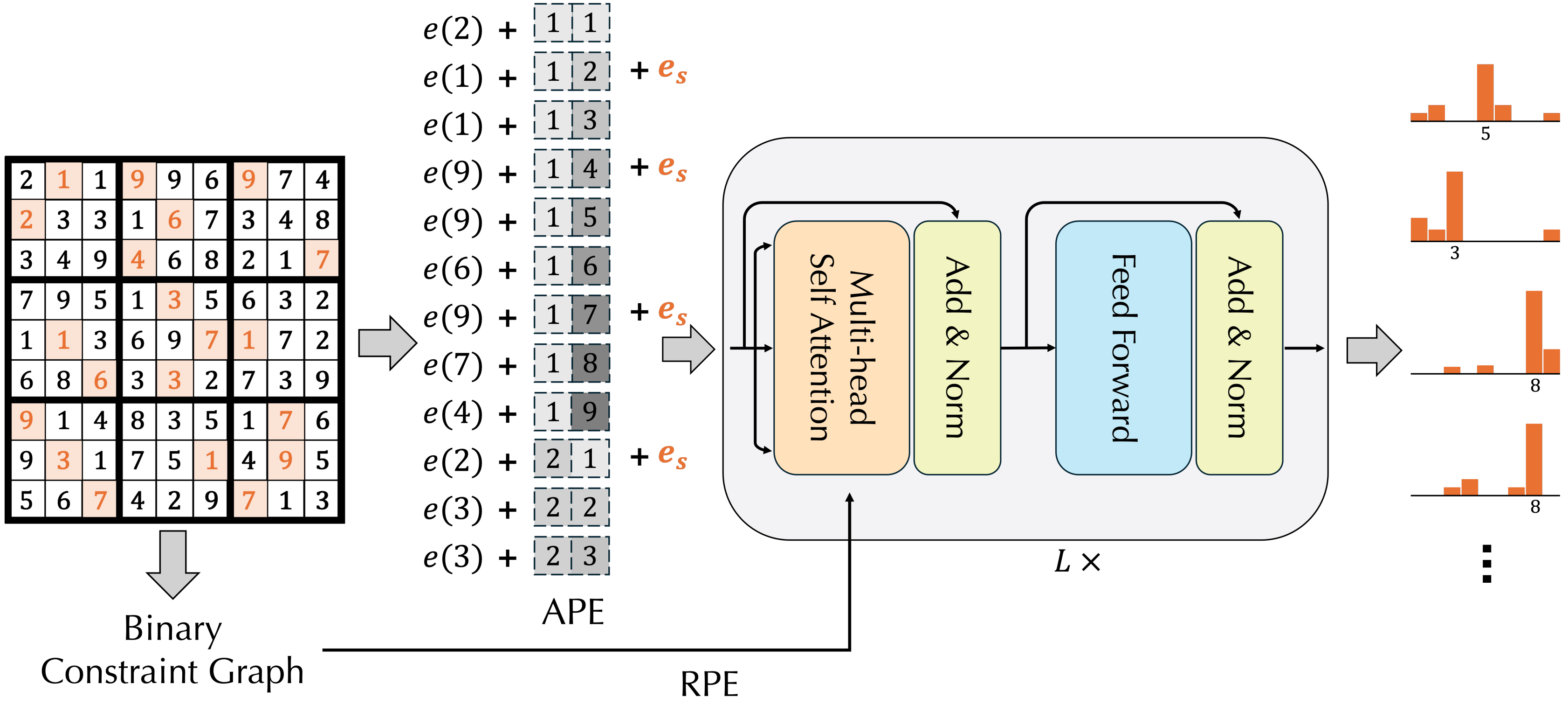
Variable assignments as tokens
→ Token embedding for discrete domains

Variables indices
→ Absolute Positional Encodings

Variables under the scope of the same constraints
→ Relative Positional Encodings

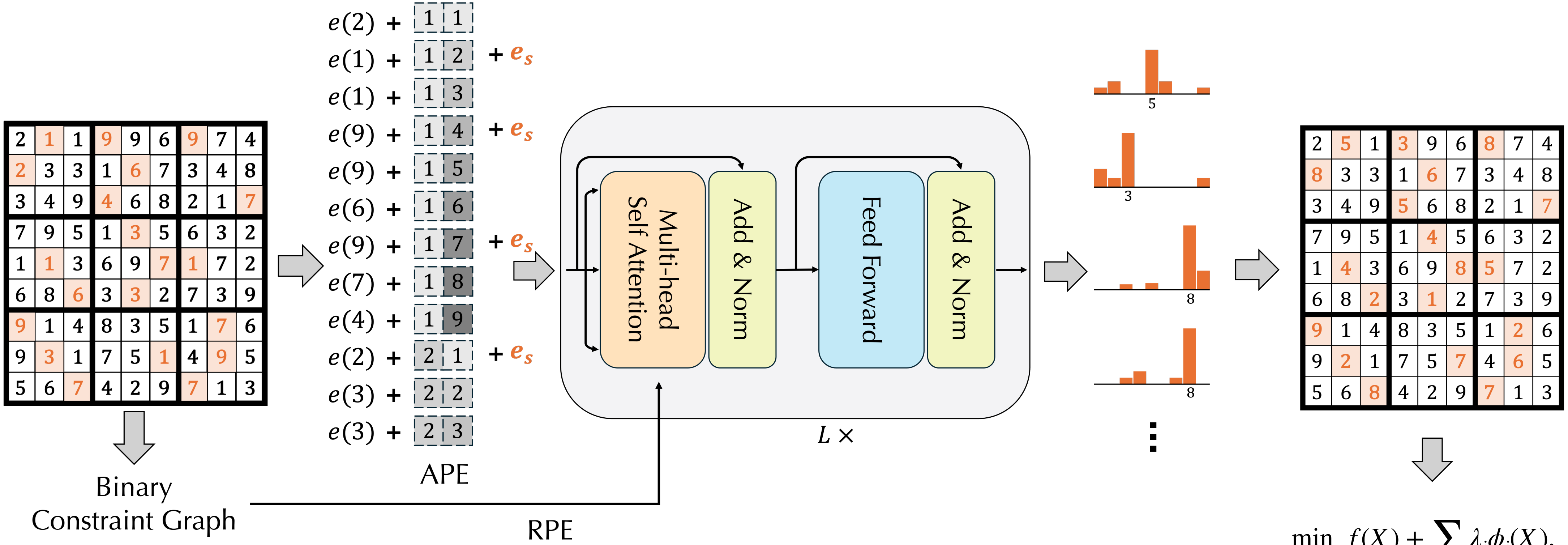
Restrict what the model is able to update.
→ Subset Selection

ConsFormer



Predicts Variable Assignment
→ Transformer

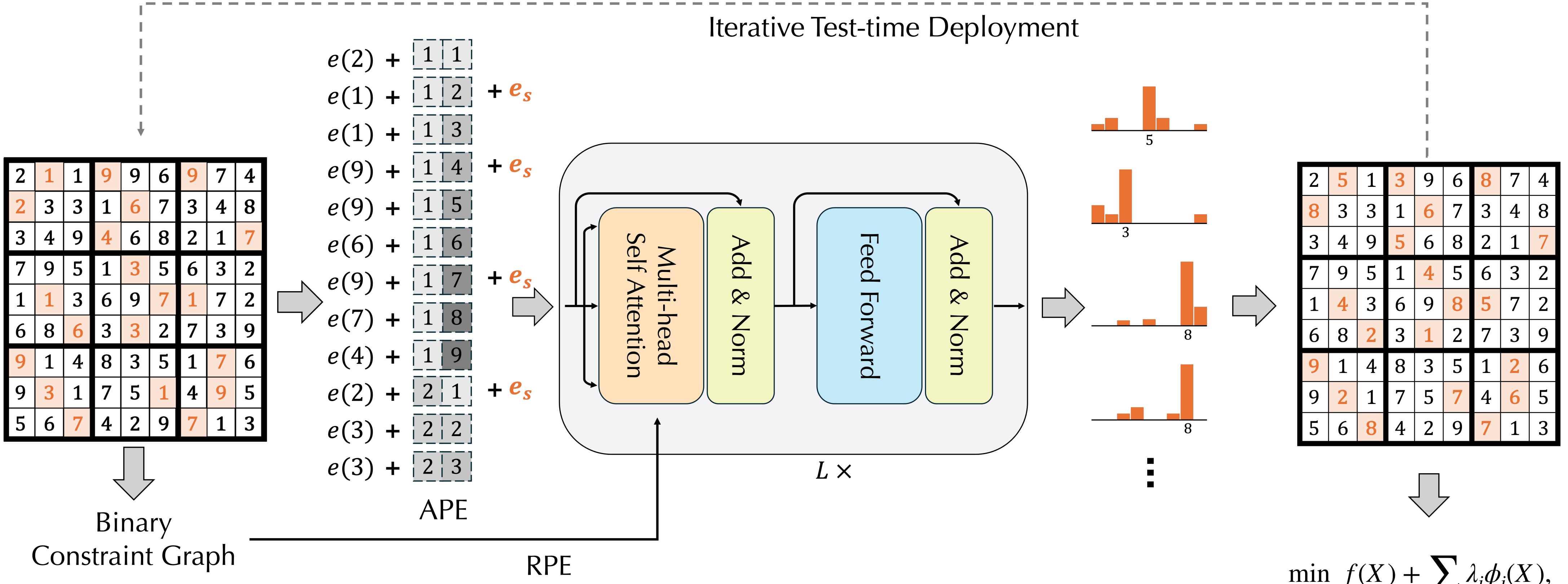
ConsFormer



$$\min_{X \in \mathcal{X}} \; f(X) + \sum_j \lambda_j \phi_j(X),$$

$$\phi_j(X) = 0 \iff c_j(X) = \text{True}$$

ConsFormer



$$\min_{X \in \mathcal{X}} \; f(X) + \sum_j \lambda_j \phi_j(X),$$

$$\phi_j(X) = 0 \iff c_j(X) = \text{True}$$

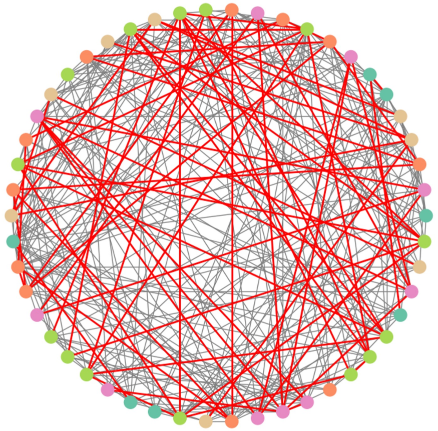
ConsFormer

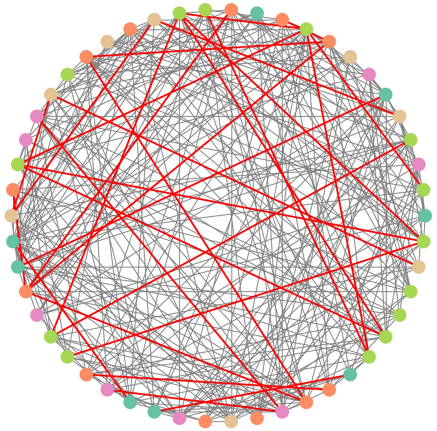
Random Initial Solution

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	6	9	5
3	4	5	5	5	9	5	1	3
1	5	7	2	4	7	5	1	9
5	8	3	3	8	2	4	7	8
6	4	3	1	1	3	3	7	6
9	4	6	3	5	1	7	7	5
1	8	5	4	6	3	6	1	7
5	6	2	6	3	7	4	2	6

Iteration 1

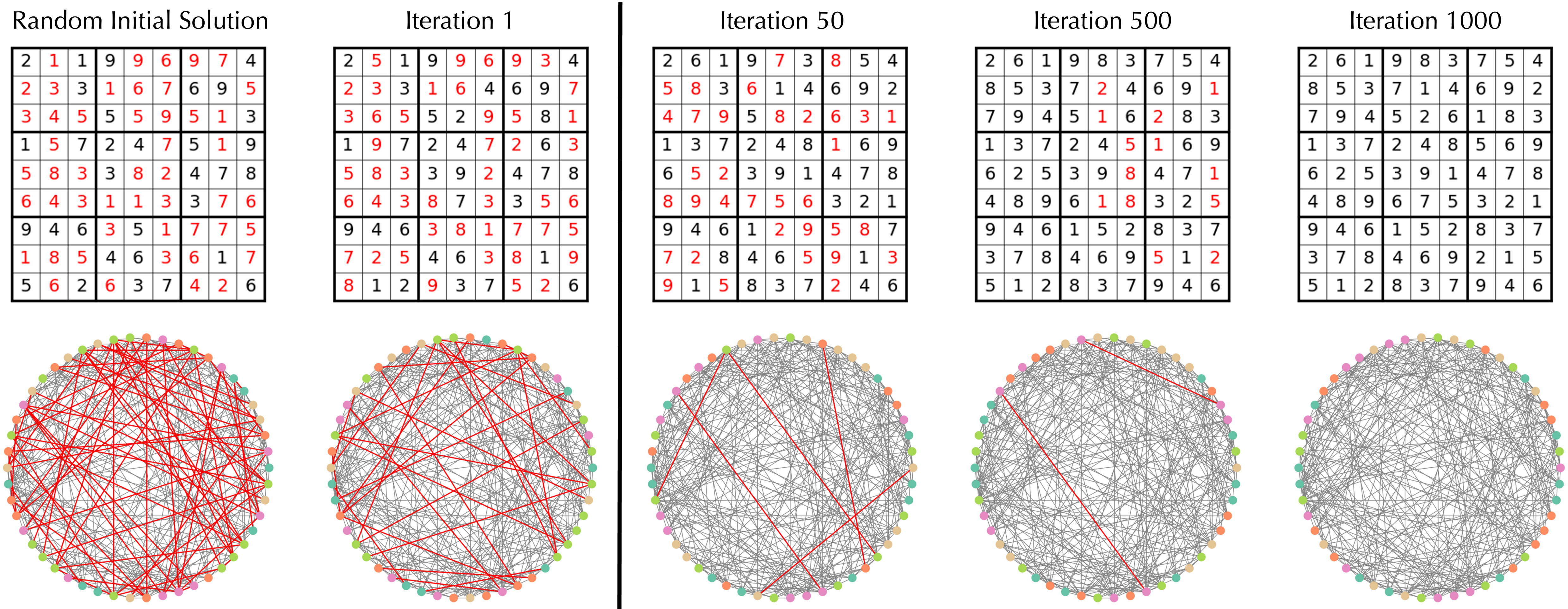
2	5	1	9	9	6	9	3	4
2	3	3	1	6	4	6	9	7
3	6	5	5	2	9	5	8	1
1	9	7	2	4	7	2	6	3
5	8	3	3	9	2	4	7	8
6	4	3	8	7	3	3	5	6
9	4	6	3	8	1	7	7	5
7	2	5	4	6	3	8	1	9
8	1	2	9	3	7	5	2	6





Single-Step Training

ConsFormer



Single-Step Training

Iterative Test-Time Deployment

Large Neighbourhood Search

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	6	9	5
3	4	5	5	5	9	5	1	3
1	5	7	2	4	7	5	1	9
5	8	3	3	8	2	4	7	8
6	4	3	1	1	3	3	7	6
9	4	6	3	5	1	7	7	5
1	8	5	4	6	3	6	1	7
5	6	2	6	3	7	4	2	6

Large Neighbourhood Search

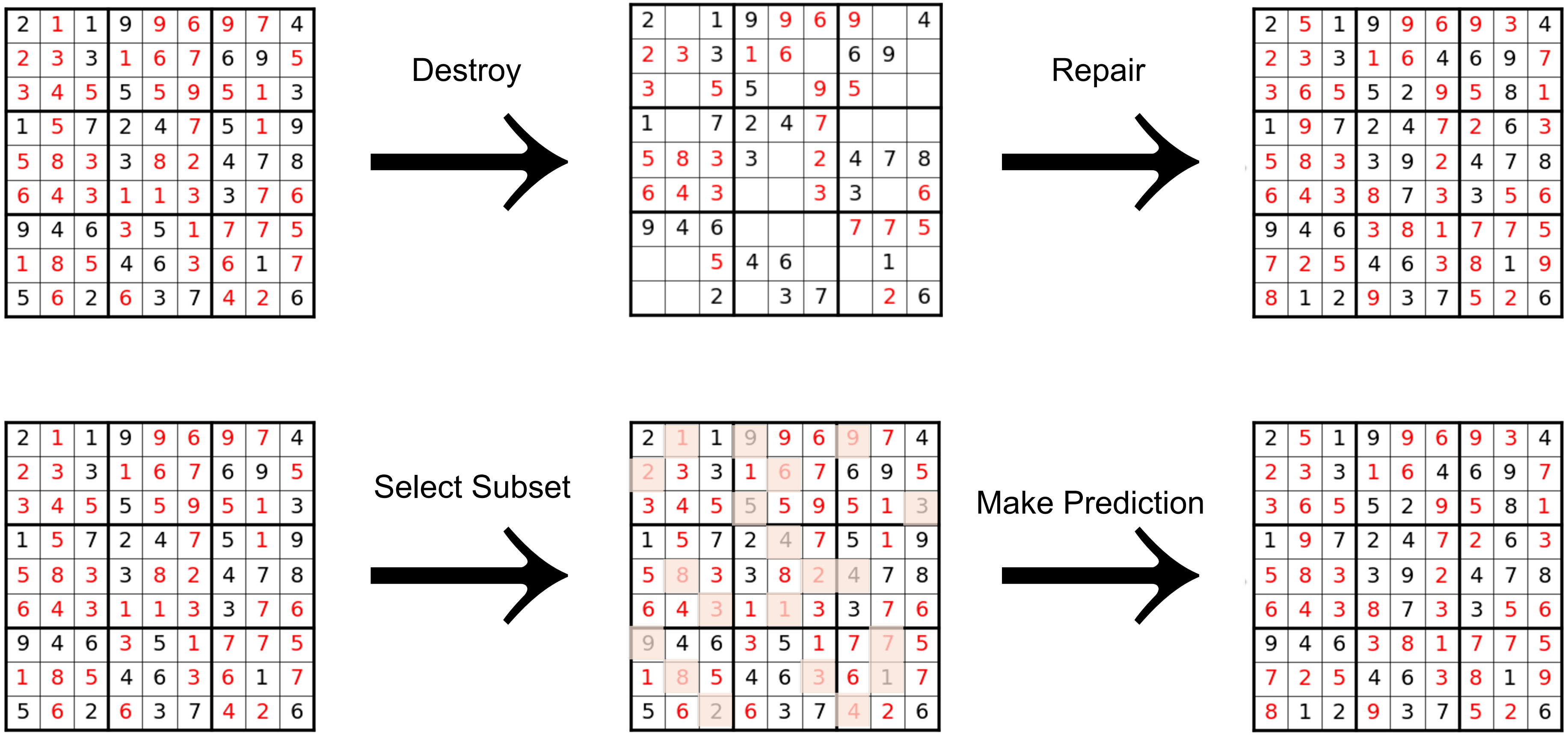
2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	6	9	5
3	4	5	5	5	9	5	1	3
1	5	7	2	4	7	5	1	9
5	8	3	3	8	2	4	7	8
6	4	3	1	1	3	3	7	6
9	4	6	3	5	1	7	7	5
1	8	5	4	6	3	6	1	7
5	6	2	6	3	7	4	2	6

Destroy

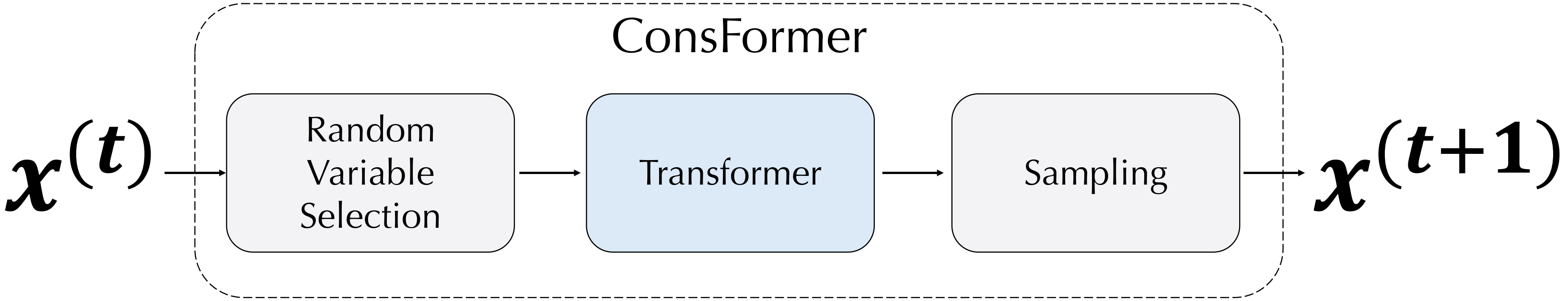
2		1	9	9	6	9		4
2	3	3	1	6		6	9	
3		5	5		9	5		
1		7	2	4	7			
5	8	3	3		2	4	7	8
6	4	3			3	3		6
9	4	6				7	7	5
		5	4	6			1	
		2		3	7		2	6

Large Neighbourhood Search

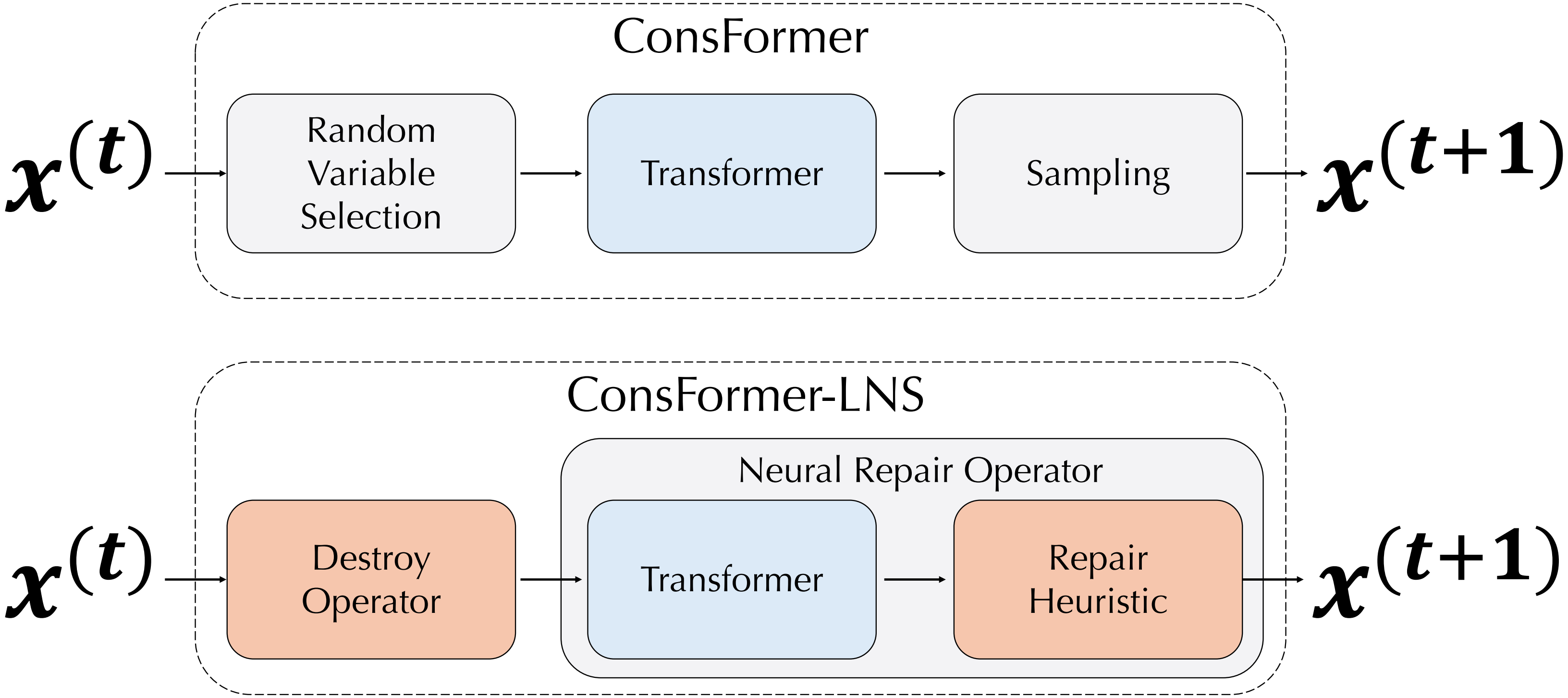
Large Neighbourhood Search



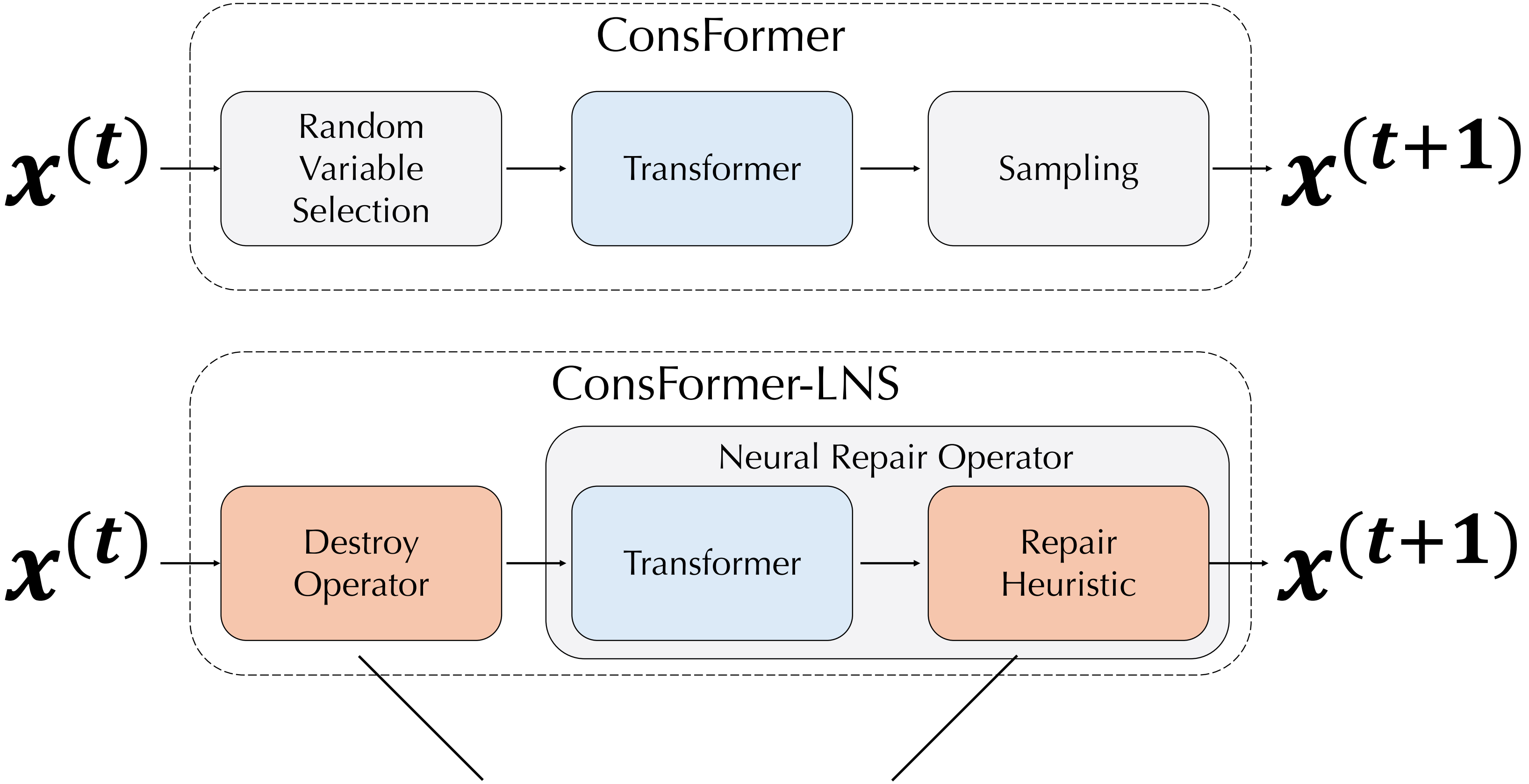
ConsFormer as Large Neighbourhood Search



ConsFormer as Large Neighbourhood Search



ConsFormer as Large Neighbourhood Search



We can now improve on these components through the lens of LNS.

Destroy Operator: which variables to update?

Some ideas from classical LNS [5]:

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

Destroy Operator: which variables to update?

Some ideas from classical LNS [5]:

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

Contribute the most to constraint violations.

Destroy Operator: which variables to update?

Some ideas from classical LNS [5]:

2	**1**	1	9	**9**	6	9	**7**	4
2	**3**	**3**	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	**3**	5	6	3	2
1	**1**	3	6	9	7	1	**7**	2
6	8	6	3	**3**	2	7	3	9
9	**1**	4	8	3	5	**1**	**7**	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	**7**	1	3

Contribute the most to constraint violations.

2	1	1	9	9	6	**9**	7	4
2	3	3	1	6	7	**3**	4	8
3	**4**	**9**	**4**	**6**	**8**	**2**	**1**	**7**
7	9	5	1	3	5	**6**	3	2
1	1	3	6	9	7	**1**	7	2
6	8	6	3	3	2	**7**	3	9
9	1	4	**8**	**3**	**5**	**1**	7	6
9	3	1	**7**	**5**	**1**	**4**	9	5
5	6	7	**4**	**2**	**9**	**7**	1	3

Belongs to the same constraint.

Destroy Operator: which variables to update?

Some ideas from classical LNS [5]:

2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

Contribute the most to constraint violations.

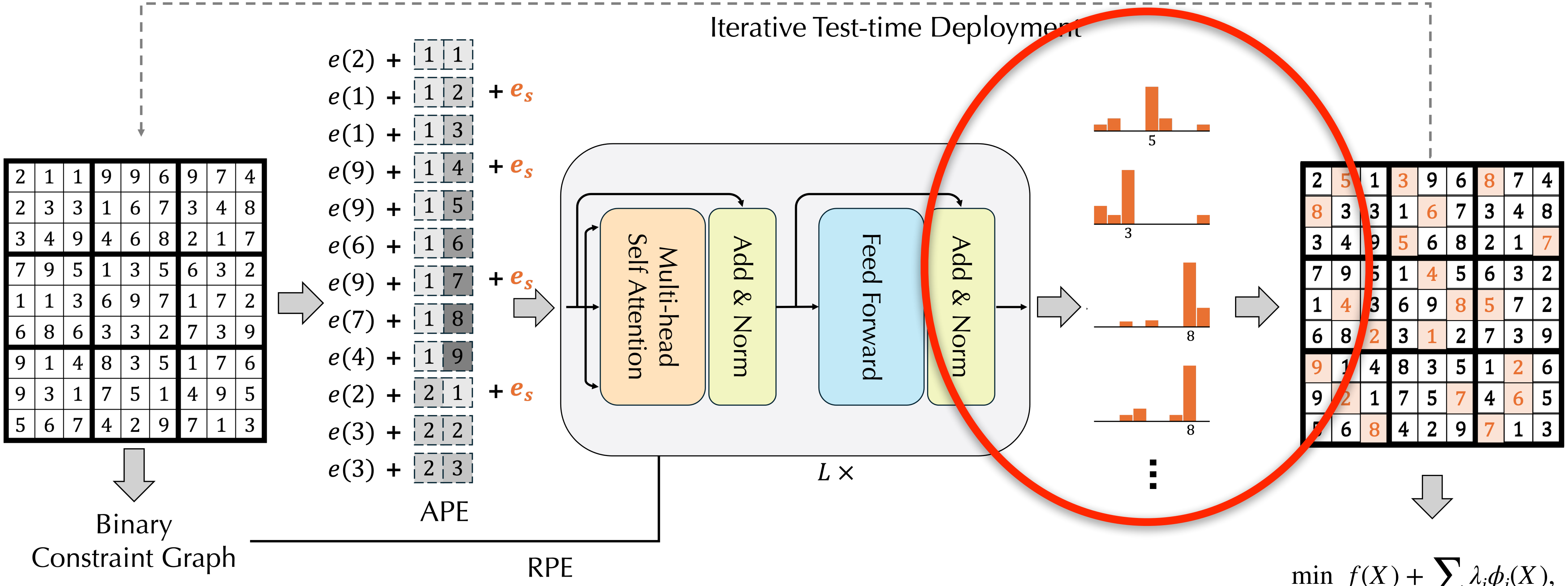
2	1	1	9	9	6	9	7	4
2	3	3	1	6	7	3	4	8
3	4	9	4	6	8	2	1	7
7	9	5	1	3	5	6	3	2
1	1	3	6	9	7	1	7	2
6	8	6	3	3	2	7	3	9
9	1	4	8	3	5	1	7	6
9	3	1	7	5	1	4	9	5
5	6	7	4	2	9	7	1	3

Belongs to the same constraint.

Can we capitalize on our neural approach?

UNIVERSITY OF TORONTO

ConsFormer

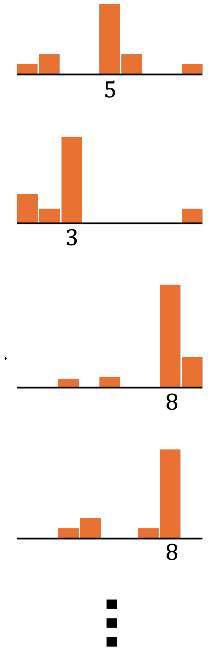


$$\min_{X \in \mathcal{X}} f(X) + \sum_j \lambda_j \phi_j(X),$$

$$\phi_j(X) = 0 \iff c_j(X) = \text{True}$$

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?



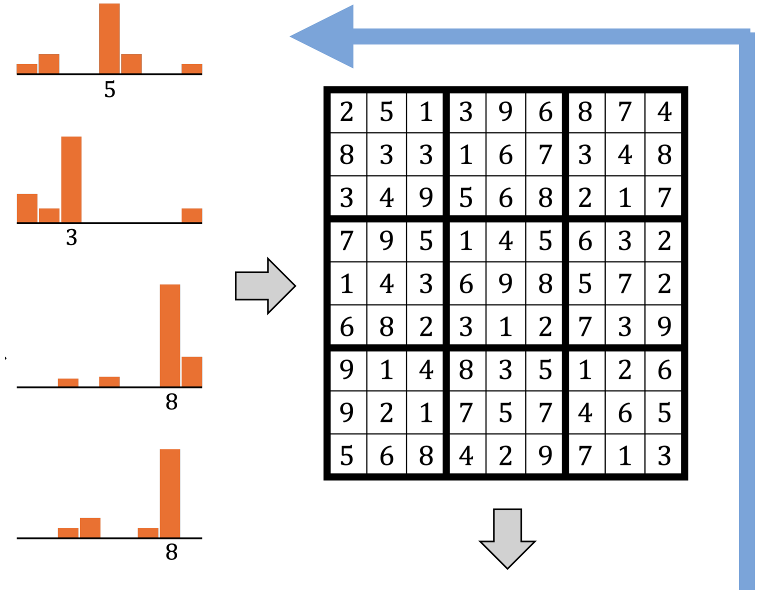
2	5	1	3	9	6	8	7	4
8	3	3	1	6	7	3	4	8
3	4	9	5	6	8	2	1	7
7	9	5	1	4	5	6	3	2
1	4	3	6	9	8	5	7	2
6	8	2	3	1	2	7	3	9
9	1	4	8	3	5	1	2	6
9	2	1	7	5	7	4	6	5
5	6	8	4	2	9	7	1	3

$$p = \sum_j \Big[\mathrm{ReLU}\Big(\sum_i x_i^{(j)} - 1\Big) + \sum_i x_i^{(j)} \Big|1 - \sum_i x_i^{(j)}\Big| \Big].$$

UNIVERSITY OF TORONTO

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?

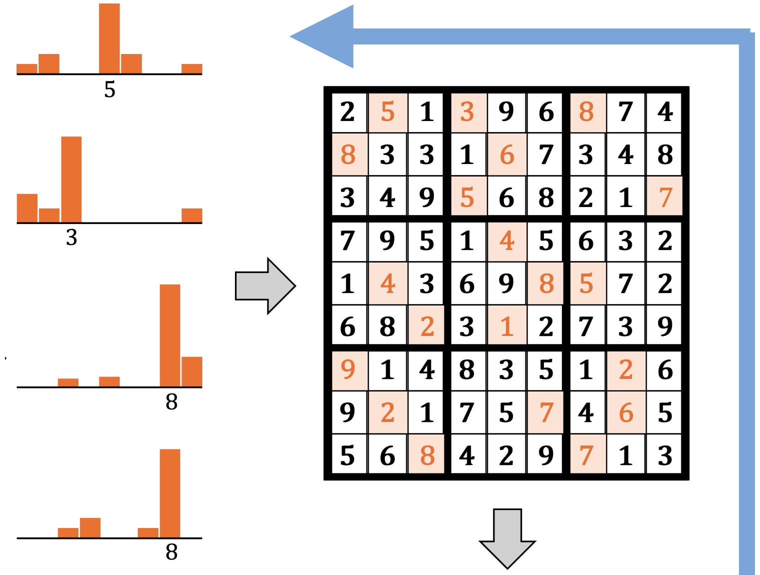


$$p = \sum_j \Big[\mathrm{ReLU}\Big(\sum_i x_i^{(j)} - 1\Big) + \sum_i x_i^{(j)} \Big| 1 - \sum_i x_i^{(j)} \Big| \Big].$$

Has the largest gradient.

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?

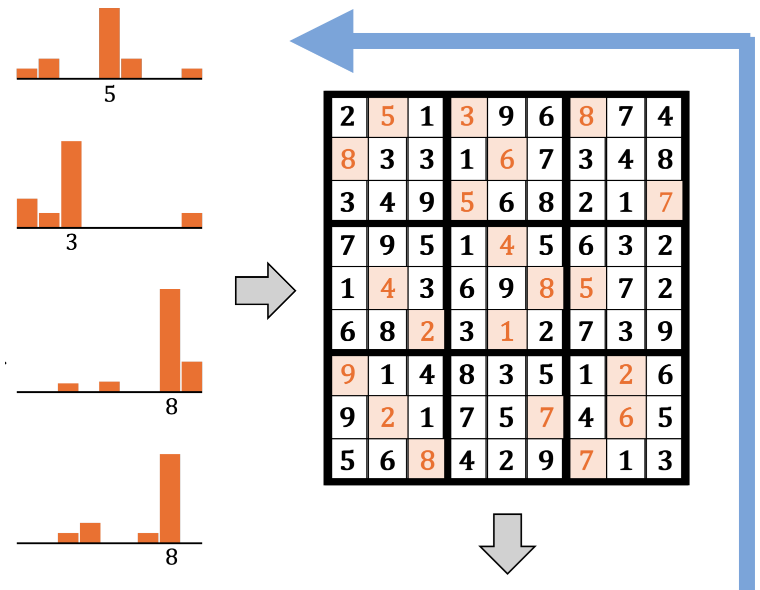


$$p = \sum_j \Big[\mathrm{ReLU}\Big(\sum_i x_i^{(j)} - 1\Big) + \sum_i x_i^{(j)} \Big|1 - \sum_i x_i^{(j)}\Big|\Big].$$

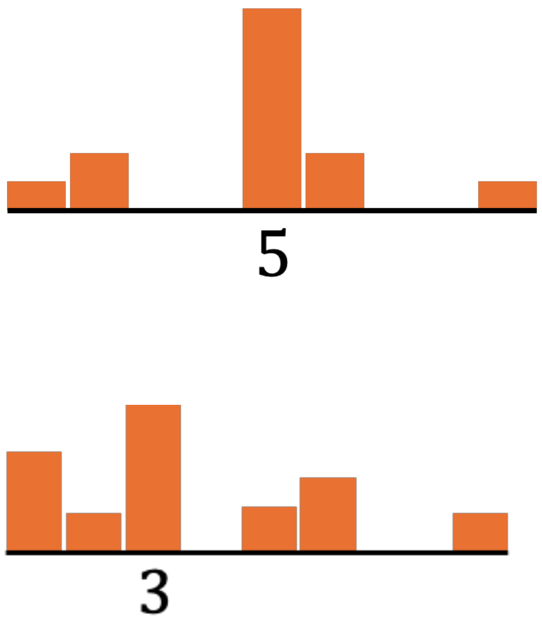
Has the largest gradient.

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?



$$p = \sum_j \Big[\mathrm{ReLU}\Big(\sum_i x_i^{(j)} - 1\Big) + \sum_i x_i^{(j)} \Big|1 - \sum_i x_i^{(j)}\Big| \Big].$$

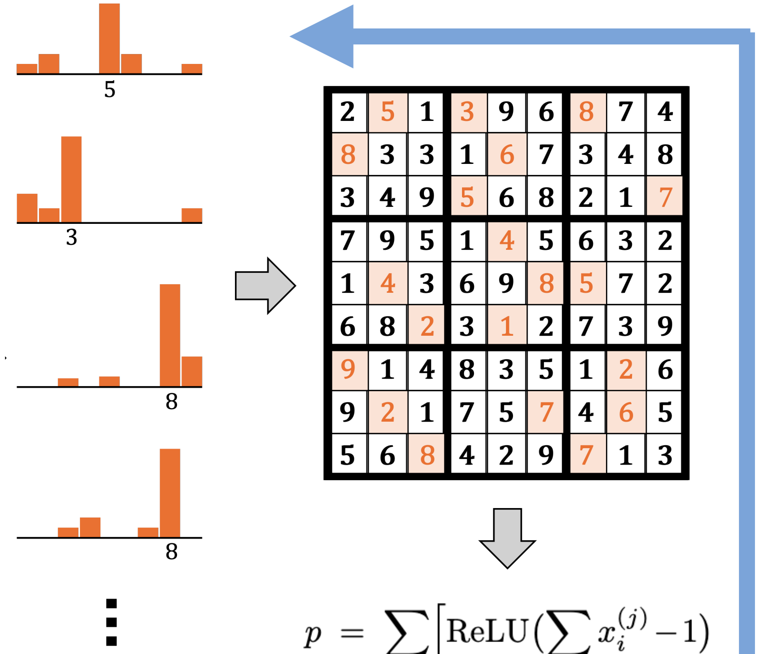


UNIVERSITY OF TORONTO

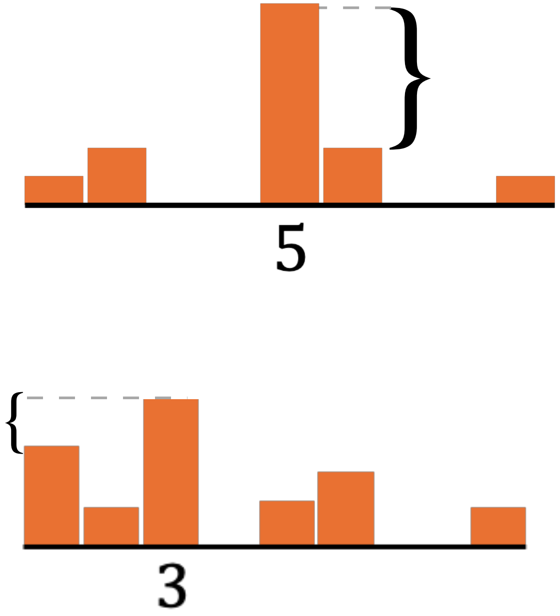
Has the largest gradient.

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?



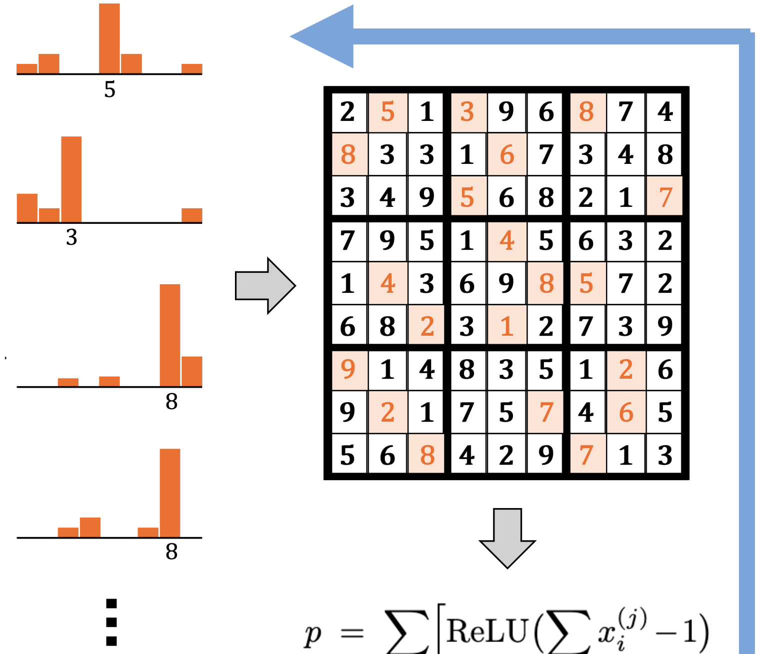
$$p = \sum_j \Big[\mathrm{ReLU}\Big(\sum_i x_i^{(j)} - 1\Big) + \sum_i x_i^{(j)} \Big|1 - \sum_i x_i^{(j)}\Big| \Big].$$



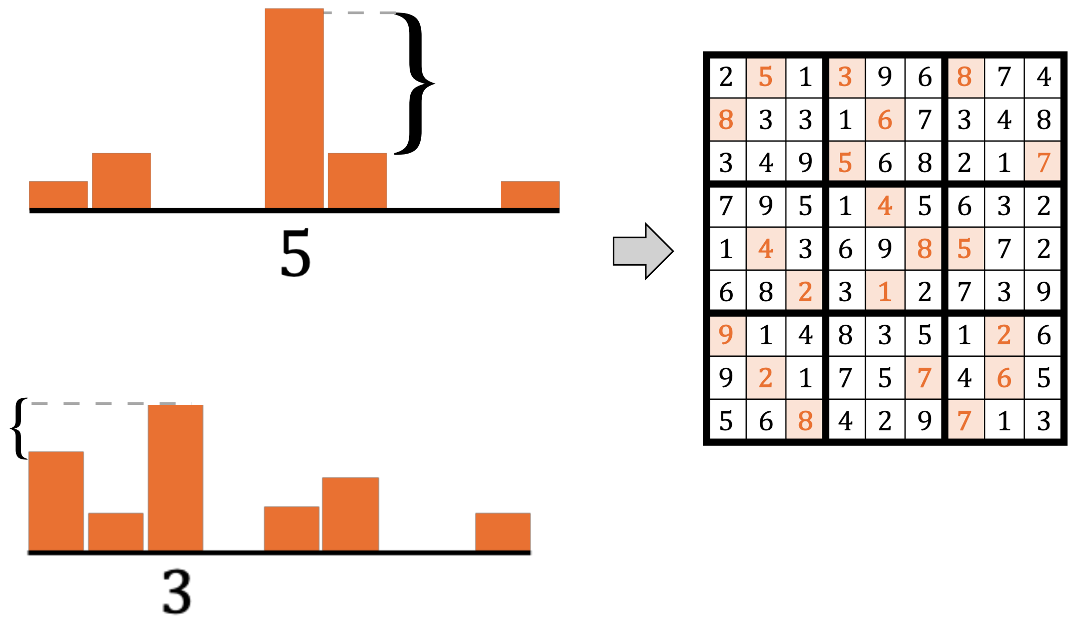
Has the largest gradient.

Destroy Operator: which variables to update?

Can we capitalize on our neural approach?



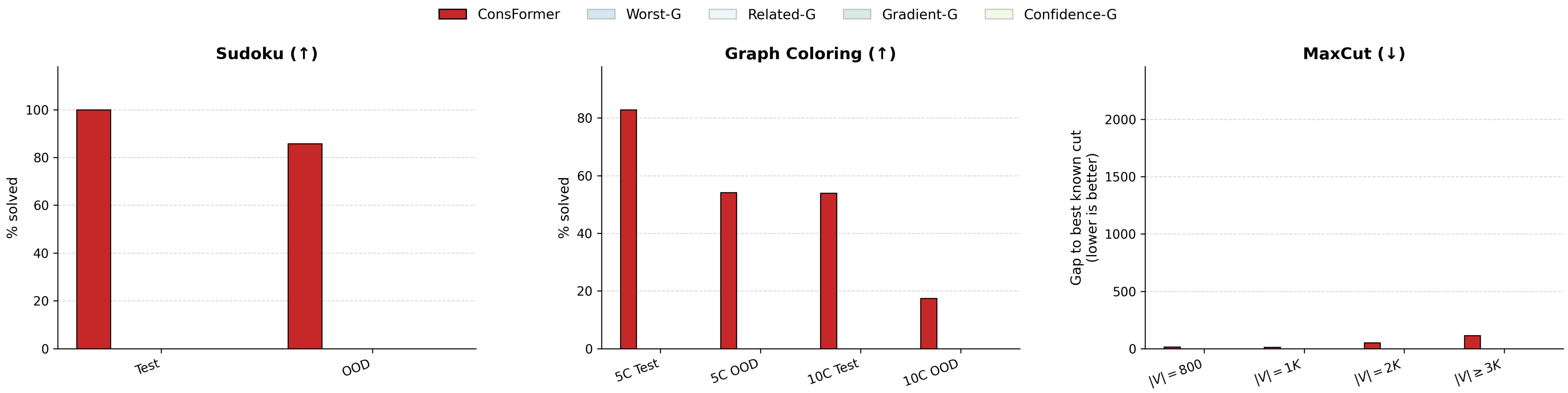
$$p = \sum_j \Big[\mathrm{ReLU}\big(\sum_i x_i^{(j)} - 1\big) + \sum_i x_i^{(j)} \,\big|1 - \sum_i x_i^{(j)}\big| \Big].$$



Has the largest gradient.

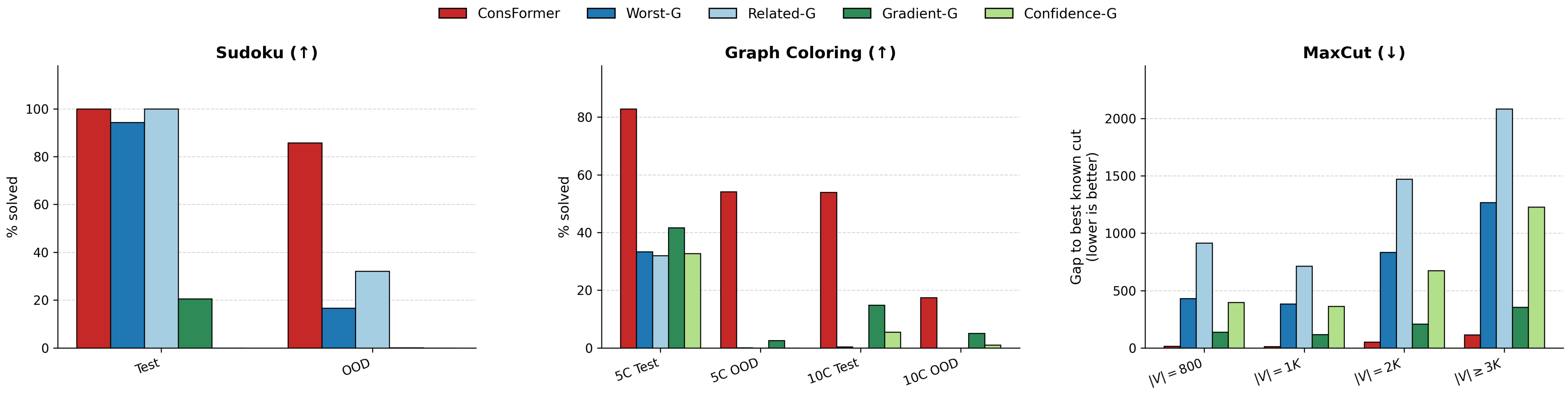
The model is least confident about.

Results



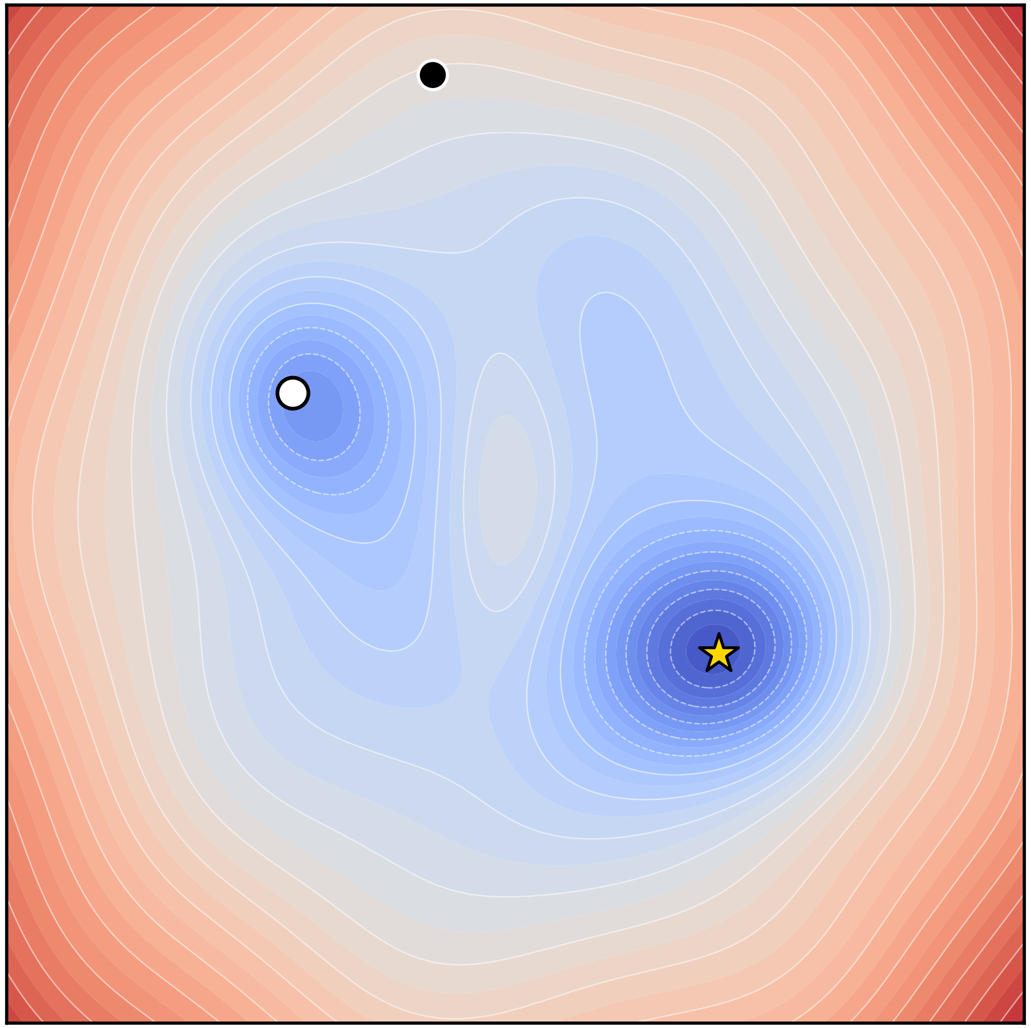
UNIVERSITY OF TORONTO

Results

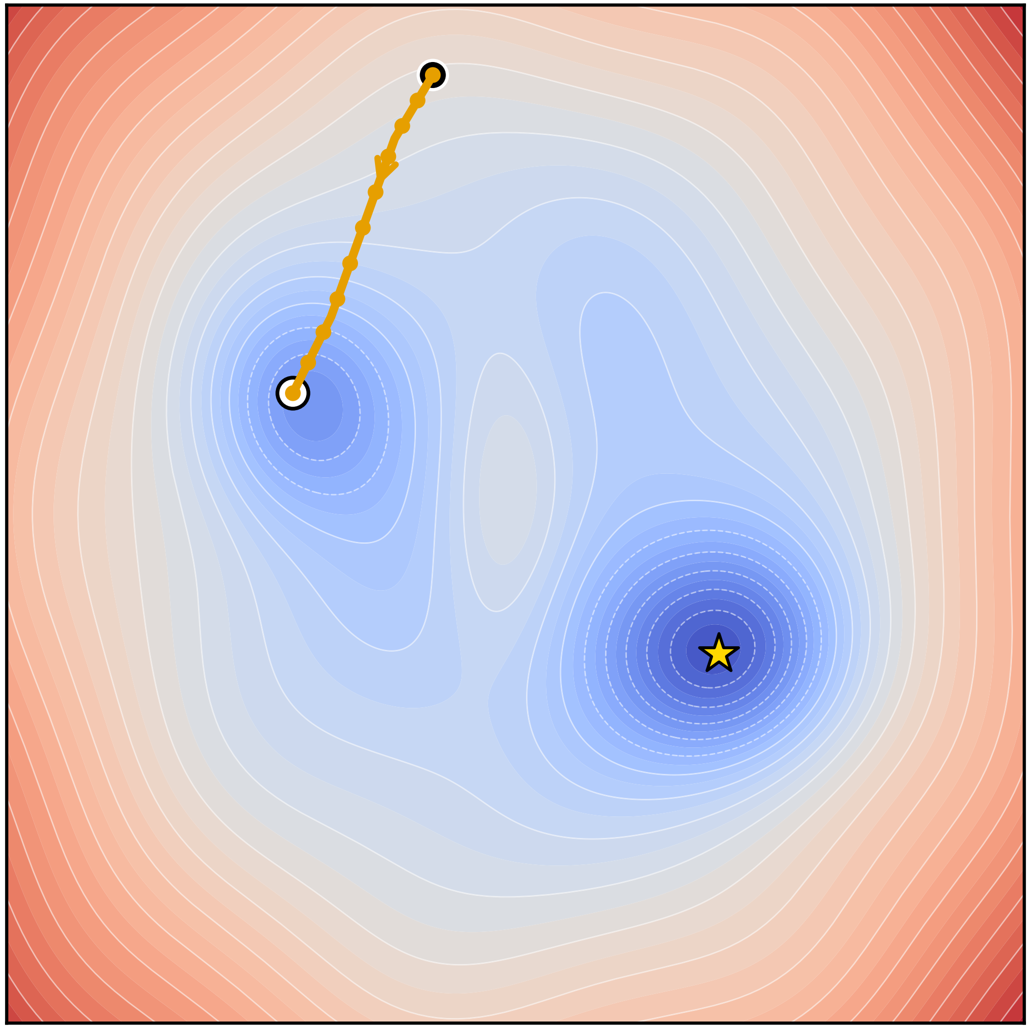


UNIVERSITY OF TORONTO

Stochasticity is important!

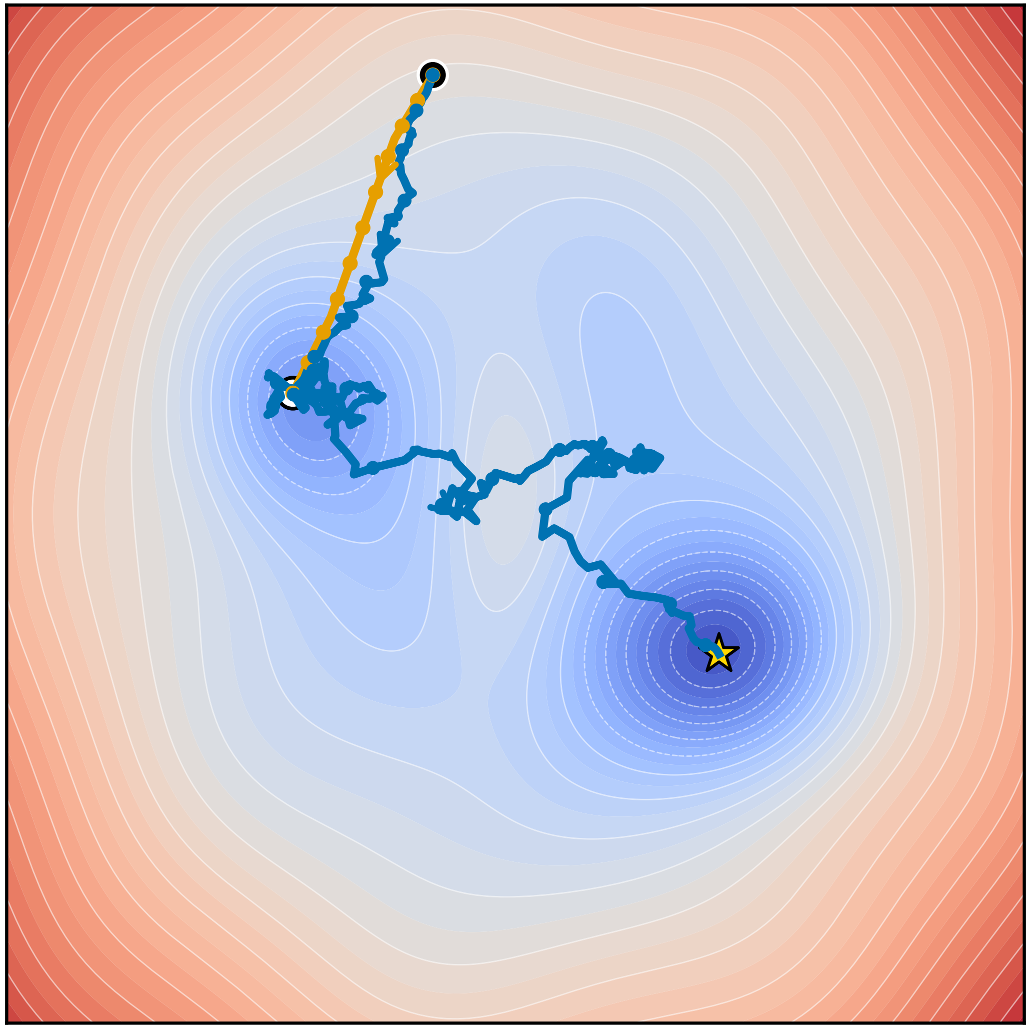


Stochasticity is important!

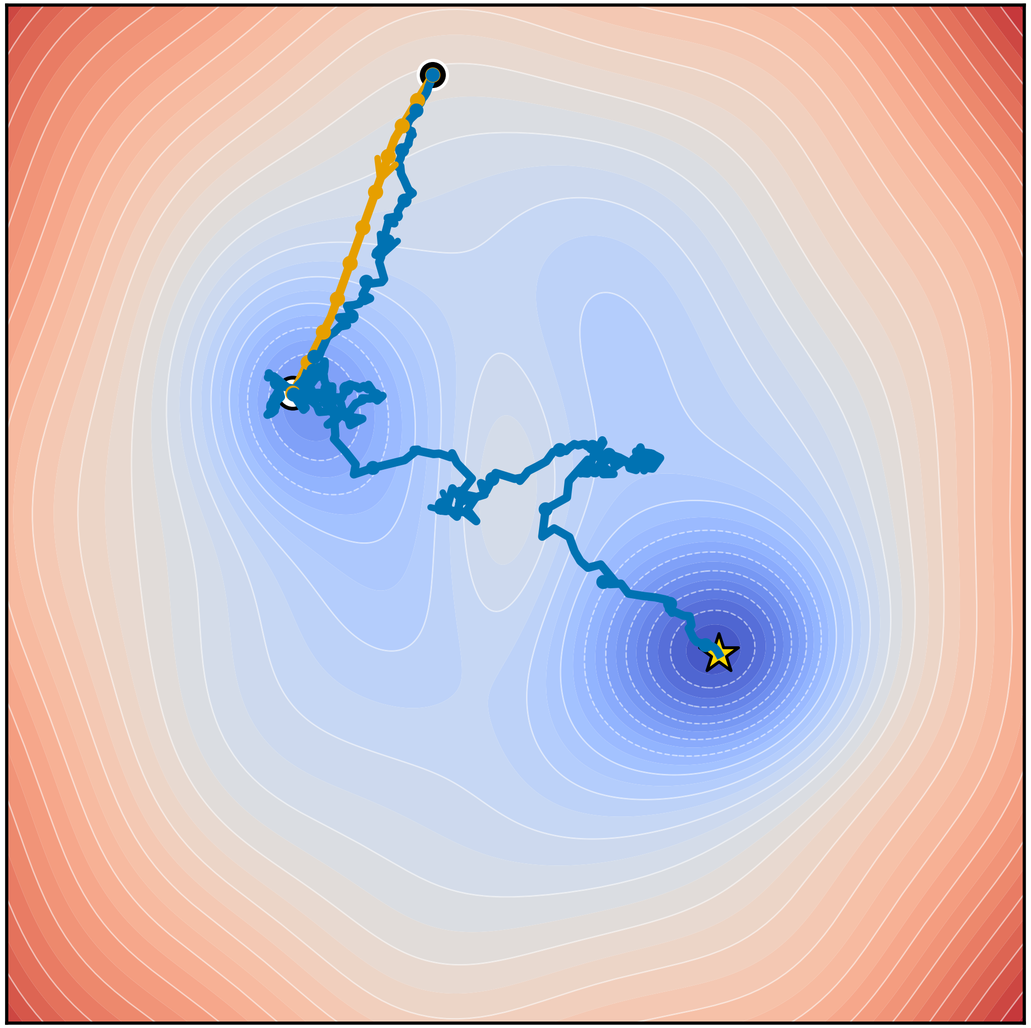


UNIVERSITY OF TORONTO

Stochasticity is important!

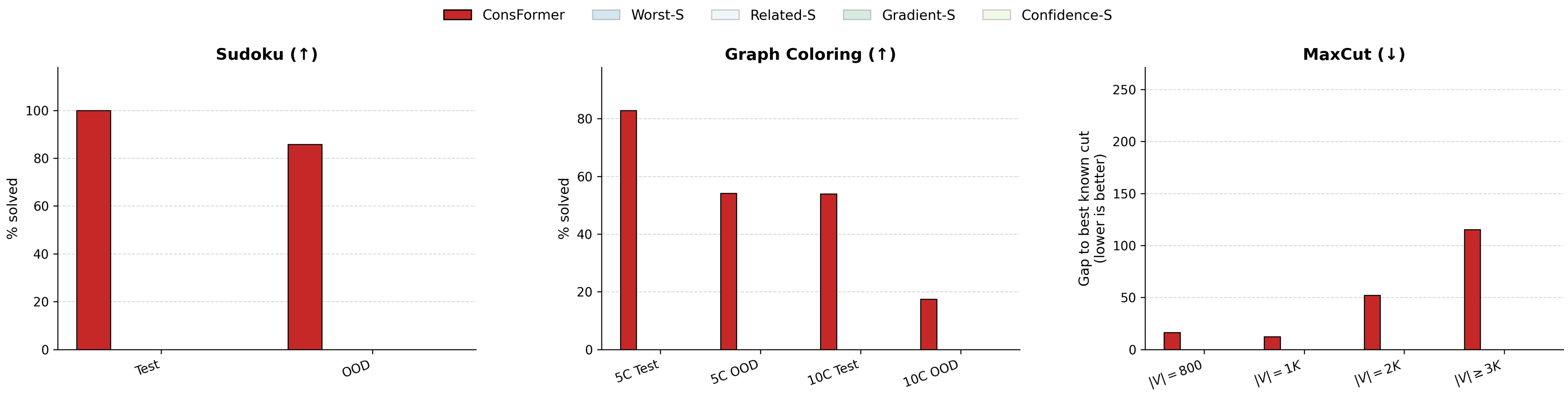


Stochasticity is important!



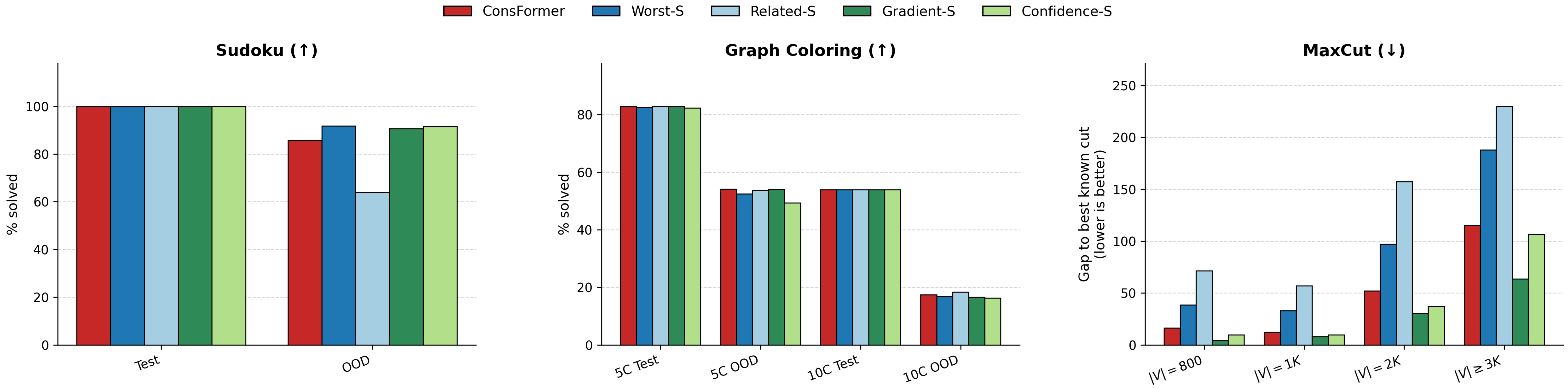
Add randomness!

Results



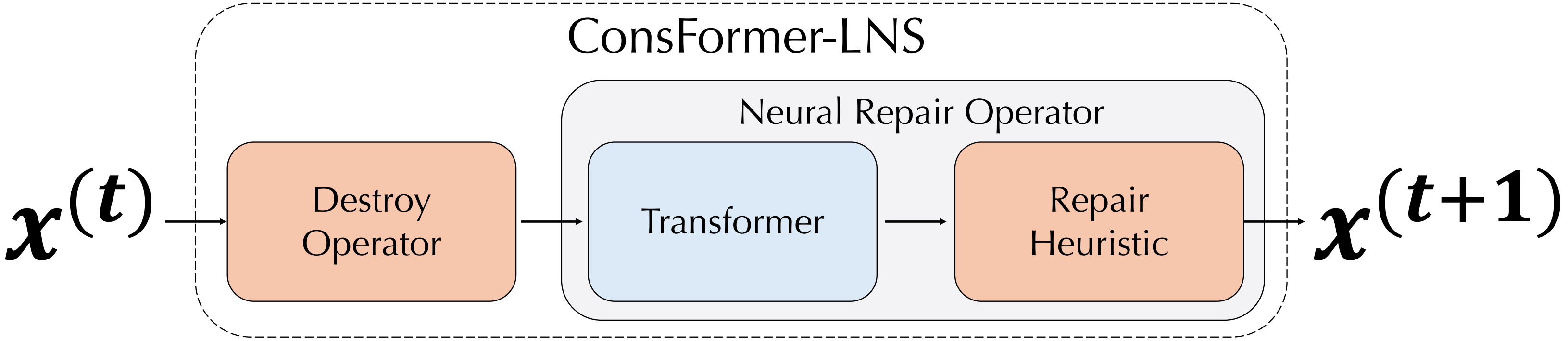
UNIVERSITY OF TORONTO

Results

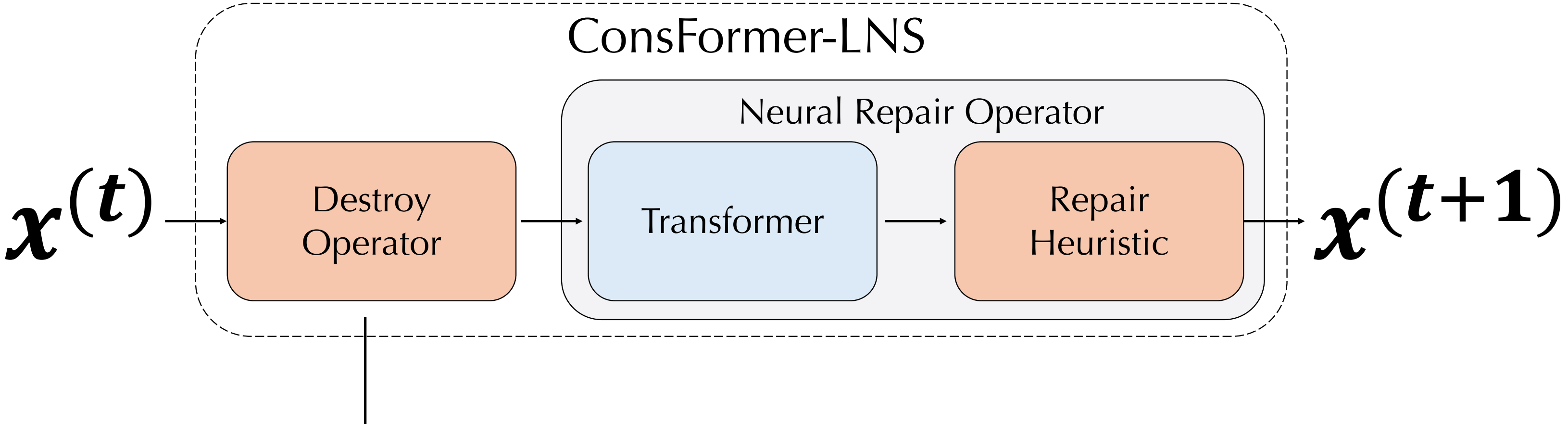


UNIVERSITY OF TORONTO

ConsFormer as Large Neighbourhood Search

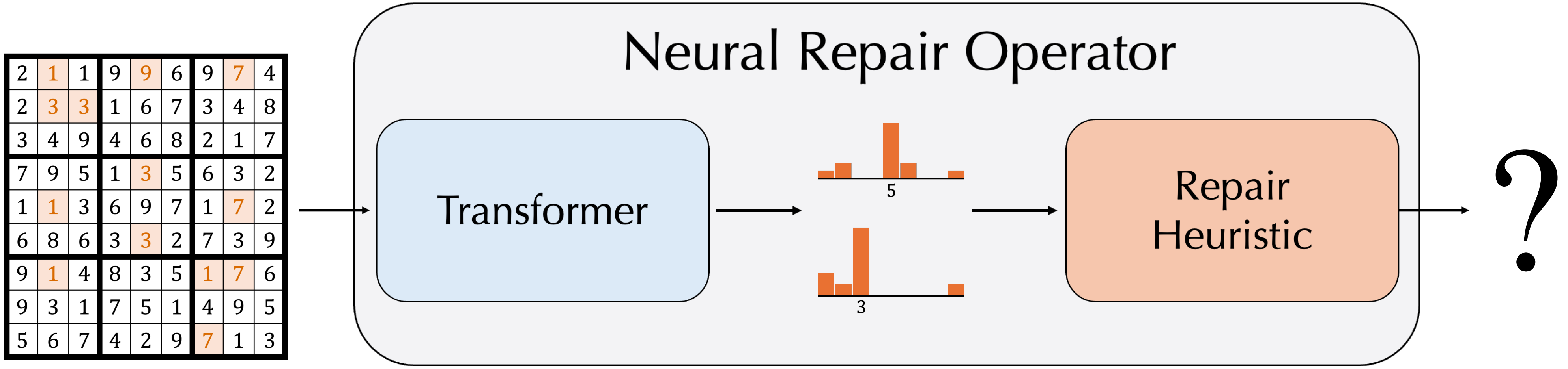


ConsFormer as Large Neighbourhood Search



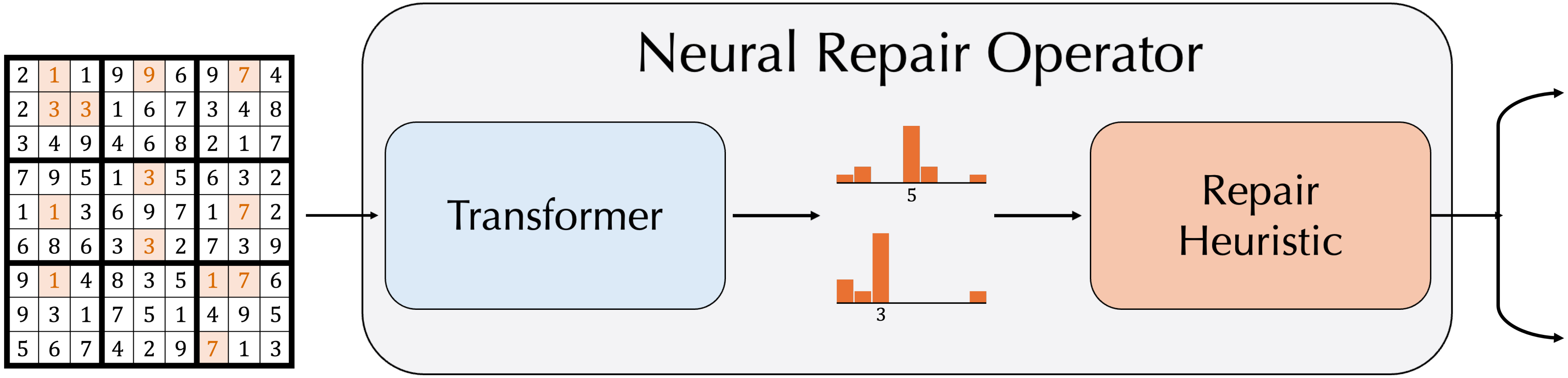
- Related Removal
- Worst Removal
- Gradient Removal
- Confidence Removal

Repair Operator: how to update the variables?

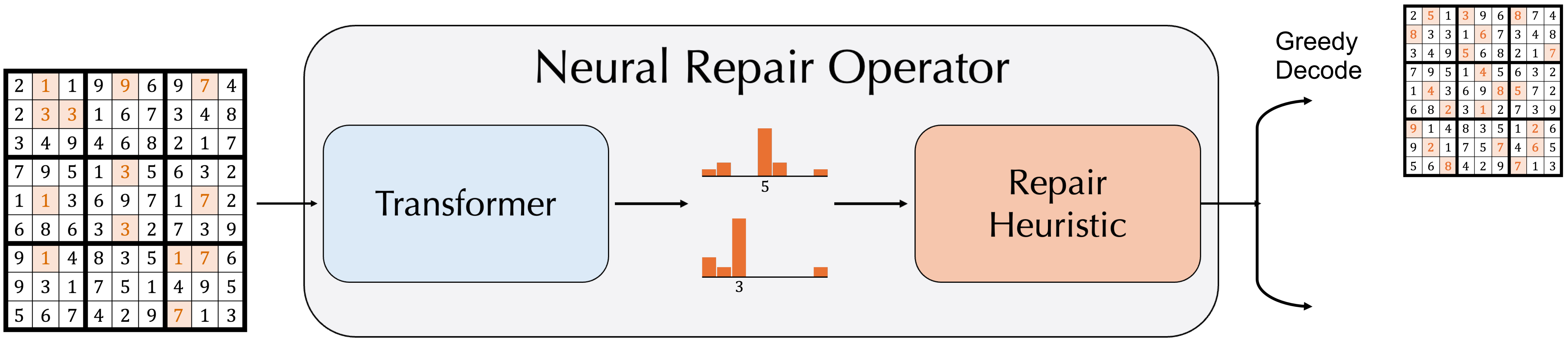


UNIVERSITY OF TORONTO

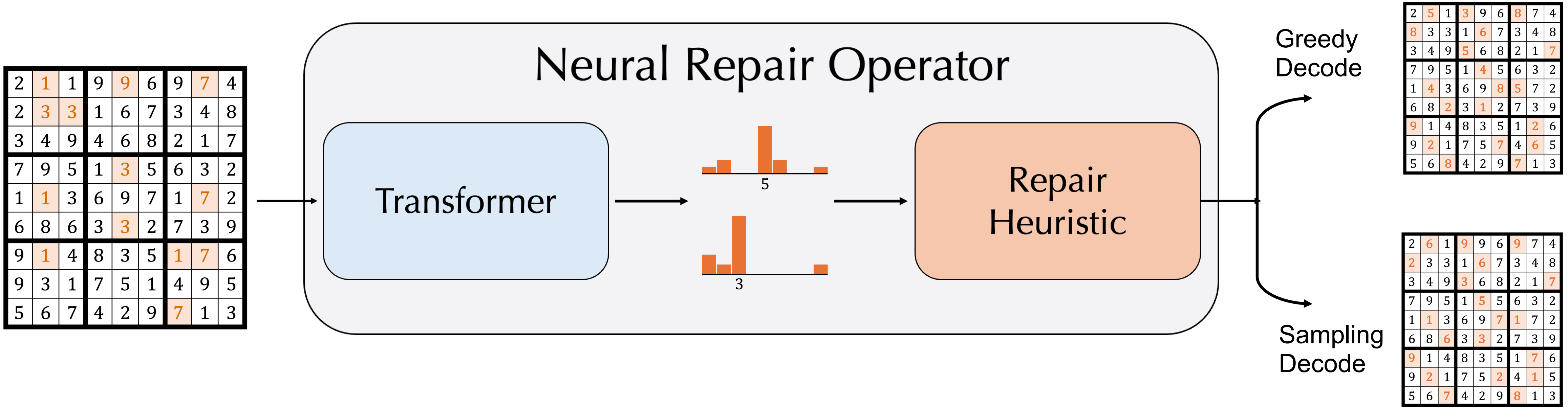
Repair Operator: how to update the variables?



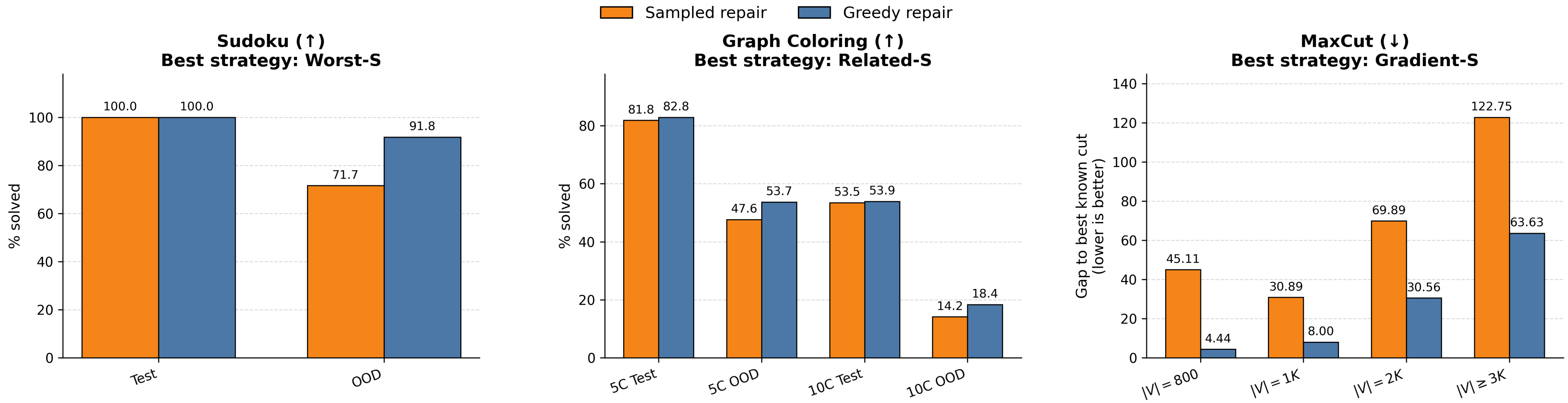
Repair Operator: how to update the variables?



Repair Operator: how to update the variables?

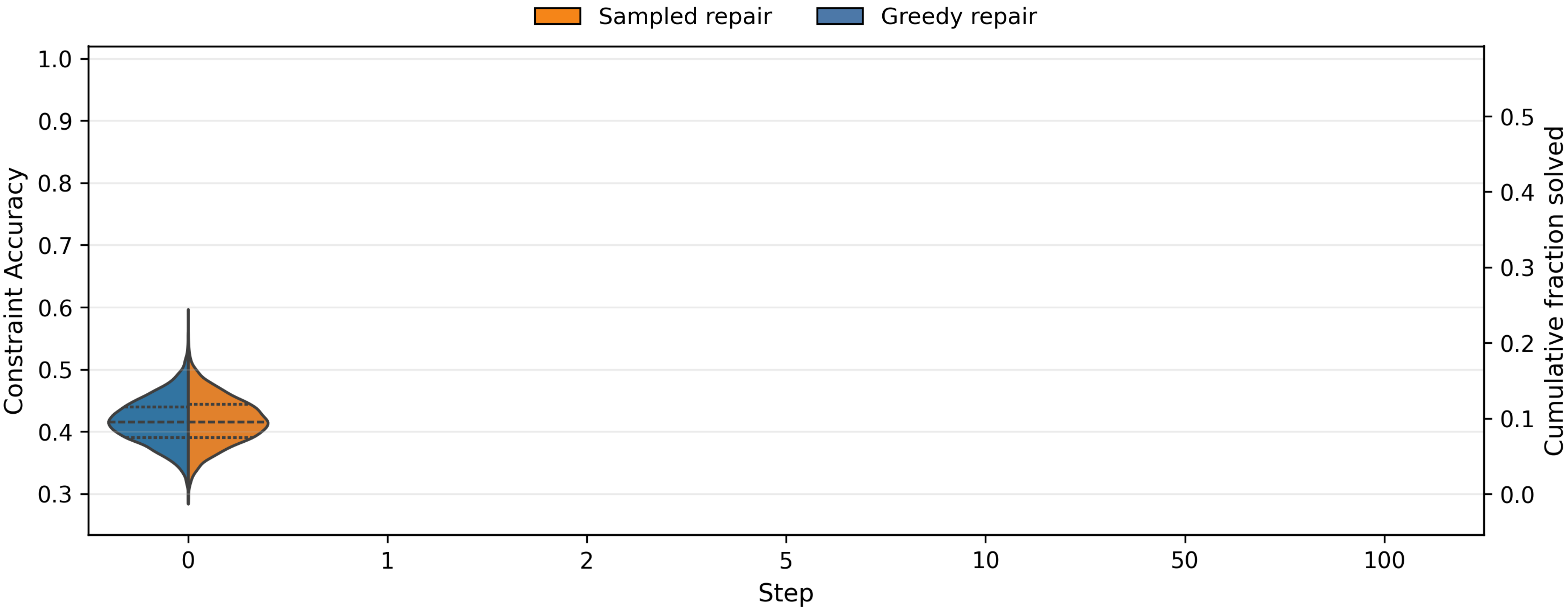


Results



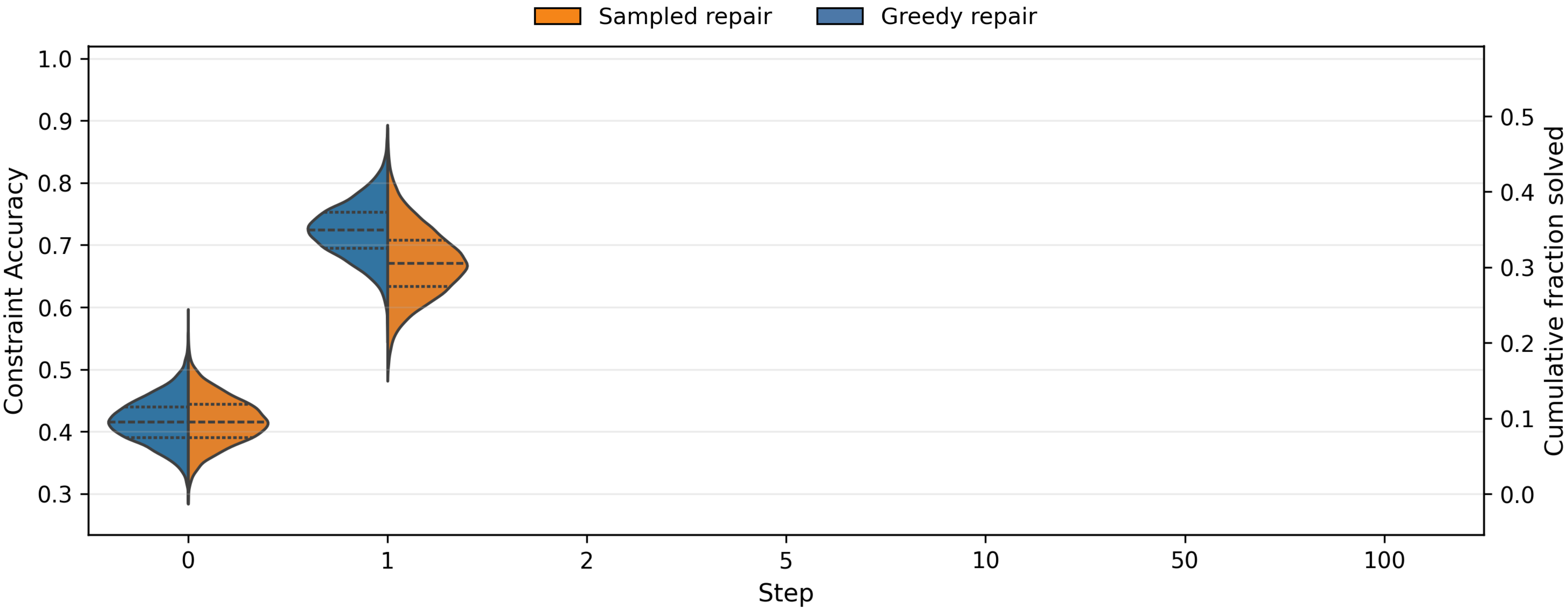
UNIVERSITY OF TORONTO

Results



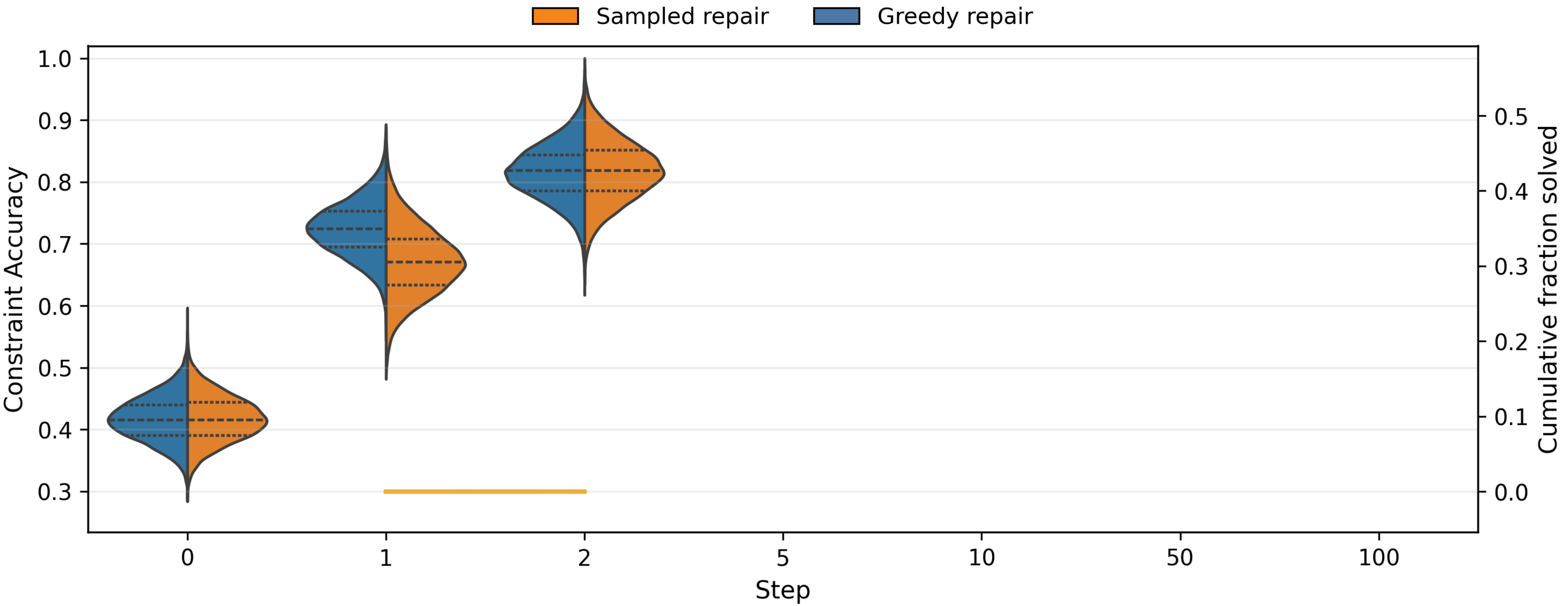
Sampled repair
Greedy repair
Constraint Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Cumulative fraction solved
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
5
10
50
100
Step

Results



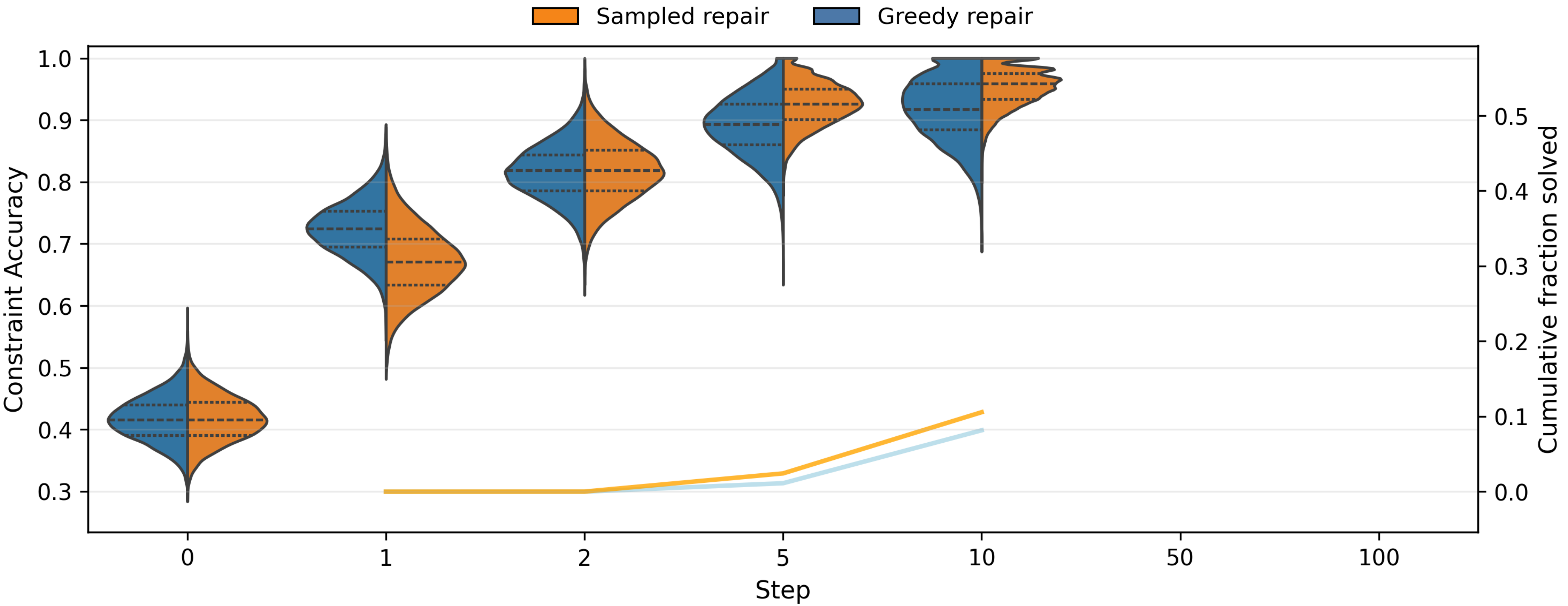
UNIVERSITY OF TORONTO

Results



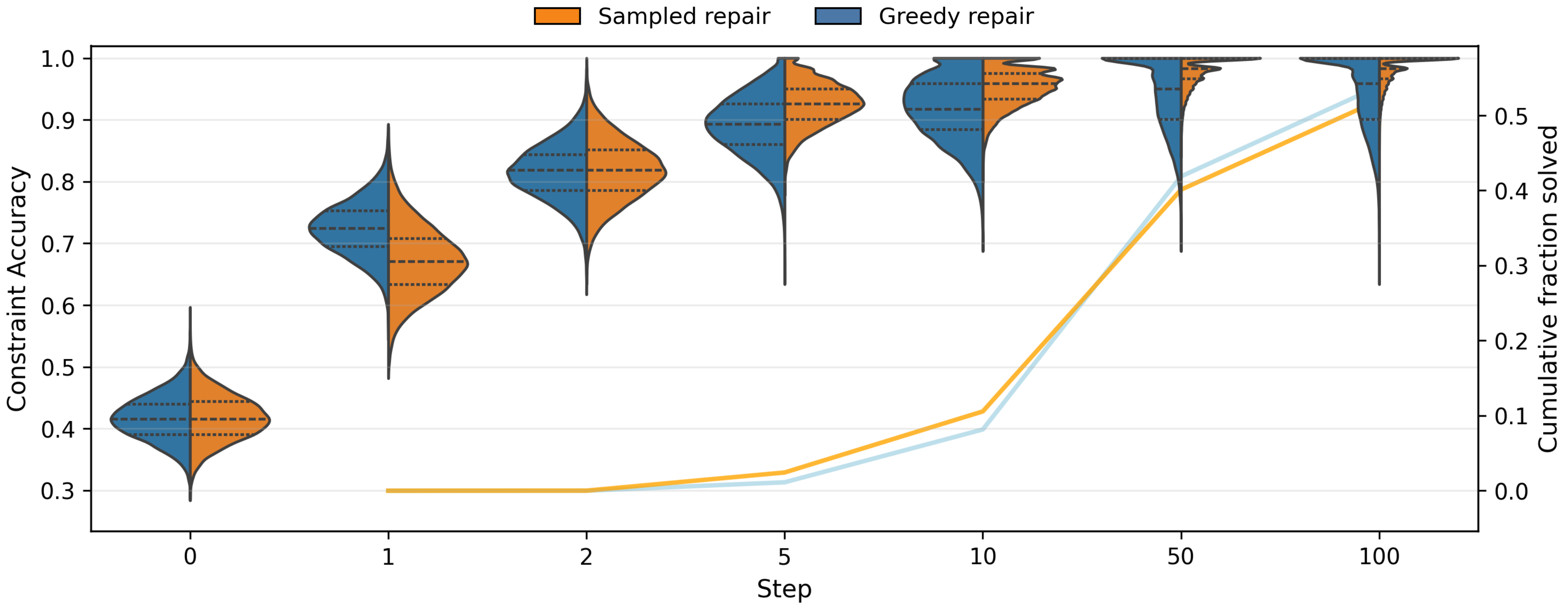
UNIVERSITY OF TORONTO

Results



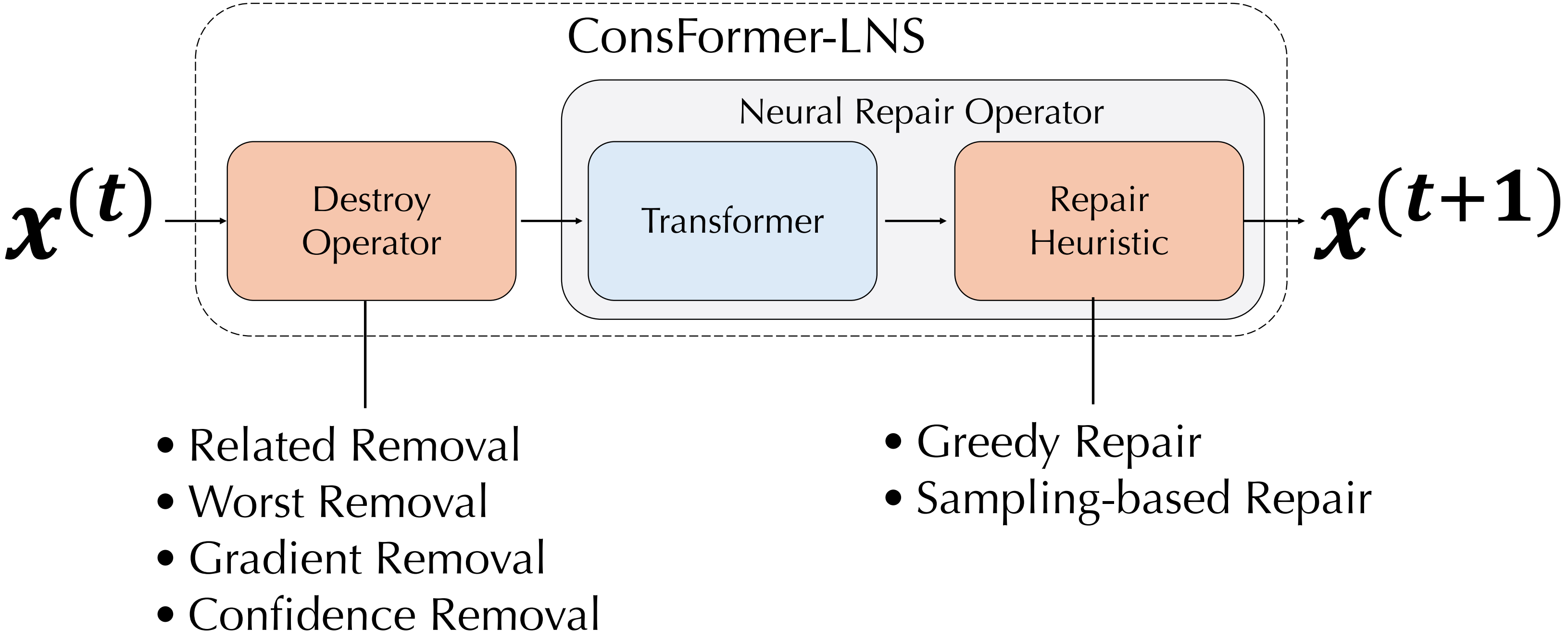
Sampled repair
Greedy repair
Constraint Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Cumulative fraction solved
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
5
10
50
100
Step

Results



UNIVERSITY OF TORONTO

ConsFormer as Large Neighbourhood Search



Conclusion

- We view Iterative Neural Heuristics as Large Neighbourhood Search.
- LNS Destroy and Repair improve performance.
- It is important to combine the power of classical and neural methods!

Contact: wil.xu@mail.utoronto.ca

Thank you!

UNIVERSITY OF TORONTO

References

[1] Khalil, Elias., “Course Overview MIE1666: Machine Learning for Mathematical Optimization” [Lecture Slides], University of Toronto (2024)
[2] Jolicoeur-Martineau, Alexia. "Less is more: Recursive reasoning with tiny networks." *arXiv preprint arXiv:2510.04871* (2025).
[3] Kim, Jaeyeon, et al. "Train for the Worst, Plan for the Best: Understanding Token Ordering in Masked Diffusions." *International Conference on Machine Learning*. PMLR, 2025.
[4] Xu, Yudong, et al. "Self-Supervised Transformers as Iterative Solution Improvers for Constraint Satisfaction." *International Conference on Machine Learning*. PMLR, 2025.
[5] Pisinger, David, and Stefan Ropke. "Large neighborhood search." *Handbook of metaheuristics*. Cham: Springer International Publishing, 2018. 99-127.