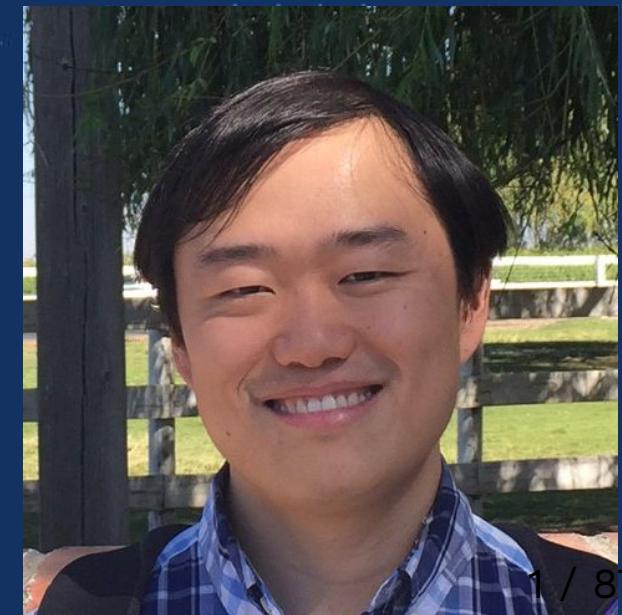


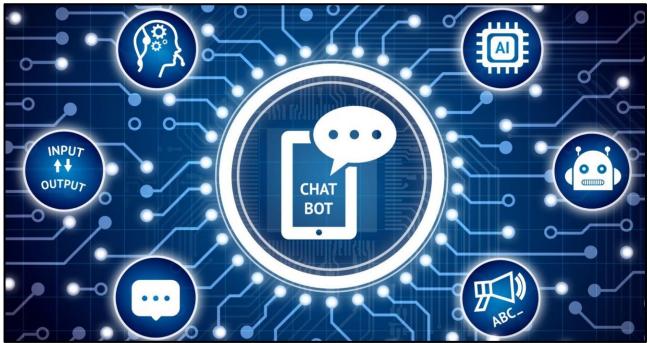
# Towards a unified framework of Neural and Symbolic Decision Making

Yuandong Tian  
Research Scientist Director

Meta AI (FAIR)



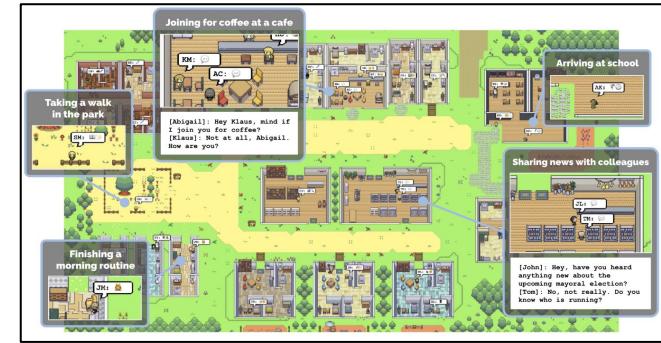
# Large Language Models (LLMs)



Conversational AI



Content Generation



AI Agents

**Standard Prompting**

**Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

A: The answer is 27. ❌

**Chain of Thought Prompting**

**Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

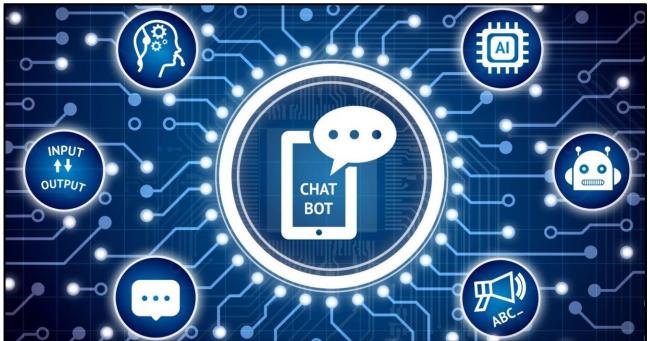
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✓

Reasoning



Planning

# Large Language Models (LLMs)



Conversational AI



Content Generation



AI Agents

**Standard Prompting**

**Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

A: The answer is 27. ❌

**Chain of Thought Prompting**

**Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✓

**Reasoning**

**Planning**

# What LLMs cannot do well yet?



User

I'm going from **Seattle** to **California** from **November 6 to 10, 2023**. I have a **budget of \$6,000**. For lodging, I prefer an **entire room** and the accommodations must be **pet-friendly**.

Travel planning

# What LLMs cannot do well yet



User

I'm going from **Seattle to California** from **November 6 to 10, 2023**. I have a **budget of \$6,000**. For lodging, I prefer an **entire room** and the accommodations must be **pet-friendly**.



## Information Collection

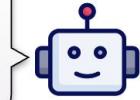
**[Tool] CitySearch[California]**  
**[Result]** San Francisco, Los Angeles, ..., San Diego 

**[Tool] FlightSearch[Seattle, San Francisco, 2023-11-06]**  
**[Result]** No Flights.

**[Tool] FlightSearch[Seattle, Los Angeles, 2023-11-06]**  
**[Result]** Flight Number: F123, 13:40-16:12, Cost: \$120

**[Tool] DistanceMatrix[Los Angeles, San Diego, taxi]**  
**[Result]** Duration: 1 hour 57 mins, Distance: 193 km, Cost: \$200

## Planning

The plan must adhere to certain **constraints**, e.g., **user needs** and **commonsense**. It's also vital to ... 

### User Needs (Hard Constraints)

1. Budget: \$6000
2. Room Type: Entire Room
3. Room Rule: Pet-friendly

### Commonsense Constraints

1. Reasonable City Route
2. Diverse Restaurants
3. Diverse Attractions
4. Non-conflicting Transportation
5. Accommodation meets Minimum Night

## Delivery Plan

2023-11-06	2023-11-07	2023-11-08	2023-11-09	2023-11-10
Seattle > Los Angeles Flight: F123 (13:40-16:12), Cost: \$120	Los Angeles • Breakfast: Chick-fil-A Minar • Accommodation: Luxury building studio • Lunch: Redhanni Restaurant • Dinner: Domino's Pizza • Attractions: Santa Monica Pier; Griffith Park • Accommodation: Luxury building studio	Los Angeles -> San Diego • Take taxi to San Diego • Breakfast: Open Yard • Lunch: The Lost Mughal • Dinner: Burger King • Attractions: Cabrillo Monument • Accommodation: East Side Apartment	San Diego • Breakfast: Baskin Robbins • Lunch: Harry's Bar • Dinner: Dragon Way • Attractions: La Jolla Shores Park; California Tower • Accommodation: East Side Apartment	San Diego -> Seattle Flight: F189 (7:59-10:56), Cost: \$300

**2023-11-06**

**Seattle -> Los Angeles**

- Flight: F123, (13:40-16:12), Cost: \$120
- Accommodation: Luxury building studio
- Dinner: The Attraction

# Using SoTA LLMs for Travel Planning (not great)

Even SoTA LLMs struggle for such hard planning problems

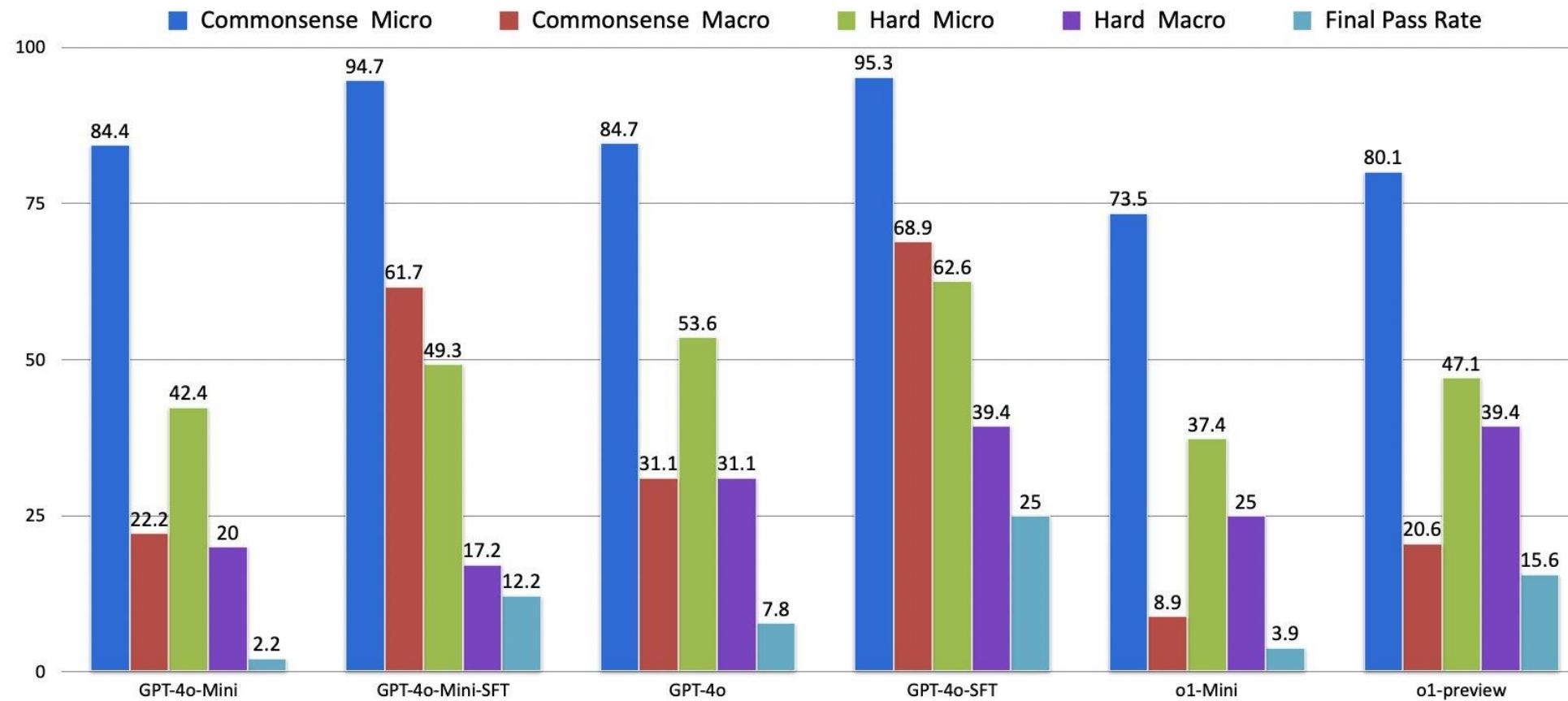
		Validation (#180)						Test (#1,000)					
		Delivery Rate	Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate	Delivery Rate	Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate
			Micro	Macro	Micro	Macro			Micro	Macro	Micro	Macro	
First tool use, Then plan the travel	Greedy Search	100	74.4	0	60.8	37.8	0	100	72.0	0	52.4	31.8	0
	<b><i>Two-stage</i></b>												
	Mistral-7B-32K (Jiang et al., 2023)	8.9	5.9	0	0	0	0	7.0	4.8	0	0	0	0
	Mixtral-8x7B-MoE (Jiang et al., 2024)	49.4	30.0	0	1.2	0.6	0	51.2	32.2	0.2	0.7	0.4	0
	Gemini Pro (G Team et al., 2023)	28.9	18.9	0	0.5	0.6	0	39.1	24.9	0	0.6	0.1	0
	GPT-3.5-Turbo (OpenAI, 2022)	86.7	54.0	0	0	0	0	91.8	57.9	0	0.5	0.6	0
Ground-truth tool use, Then plan the travel	GPT-4-Turbo (OpenAI, 2023)	<b>89.4</b>	<b>61.1</b>	<b>2.8</b>	<b>15.2</b>	<b>10.6</b>	<b>0.6</b>	<b>93.1</b>	<b>63.3</b>	<b>2.0</b>	<b>10.5</b>	<b>5.5</b>	<b>0.6</b>
	<b><i>Sole-planning</i></b>												
	Direct <sub>GPT-3.5-Turbo</sub>	100	60.2	4.4	11.0	2.8	0	100	59.5	2.7	9.5	4.4	0.6
	CoT <sub>GPT-3.5-Turbo</sub>	100	66.3	3.3	11.9	5.0	0	100	64.4	2.3	9.8	3.8	0.4
	ReAct <sub>GPT-3.5-Turbo</sub>	82.2	47.6	3.9	11.4	6.7	0.6	81.6	45.9	2.5	10.7	3.1	0.7
	Reflexion <sub>GPT-3.5-Turbo</sub>	93.9	53.8	2.8	11.0	2.8	0	92.1	52.1	2.2	9.9	3.8	0.6
	Direct <sub>Mixtral-8x7B-MoE</sub>	100	68.1	5.0	3.3	1.1	0	99.3	67.0	3.7	3.9	1.6	0.7
facebook Artificial Intelligence	Direct <sub>Gemini Pro</sub>	93.9	65.0	8.3	9.3	4.4	0.6	93.7	64.7	7.9	10.6	4.7	2.1
	Direct <sub>GPT-4-Turbo</sub>	<b>100</b>	<b>80.4</b>	<b>17.2</b>	<b>47.1</b>	<b>22.2</b>	<b>4.4</b>	<b>100</b>	<b>80.6</b>	<b>15.2</b>	<b>44.3</b>	<b>23.1</b>	<b>4.4</b>

# How about o1?

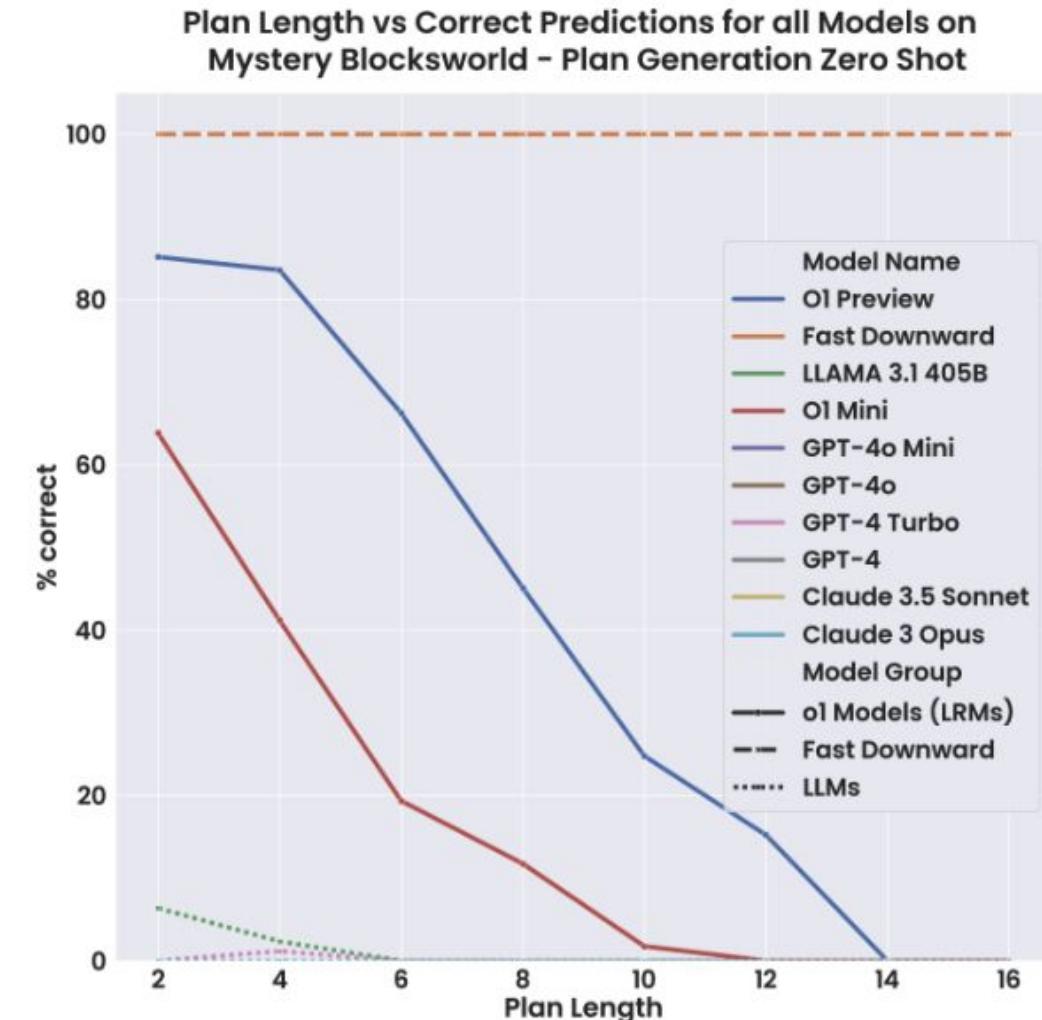
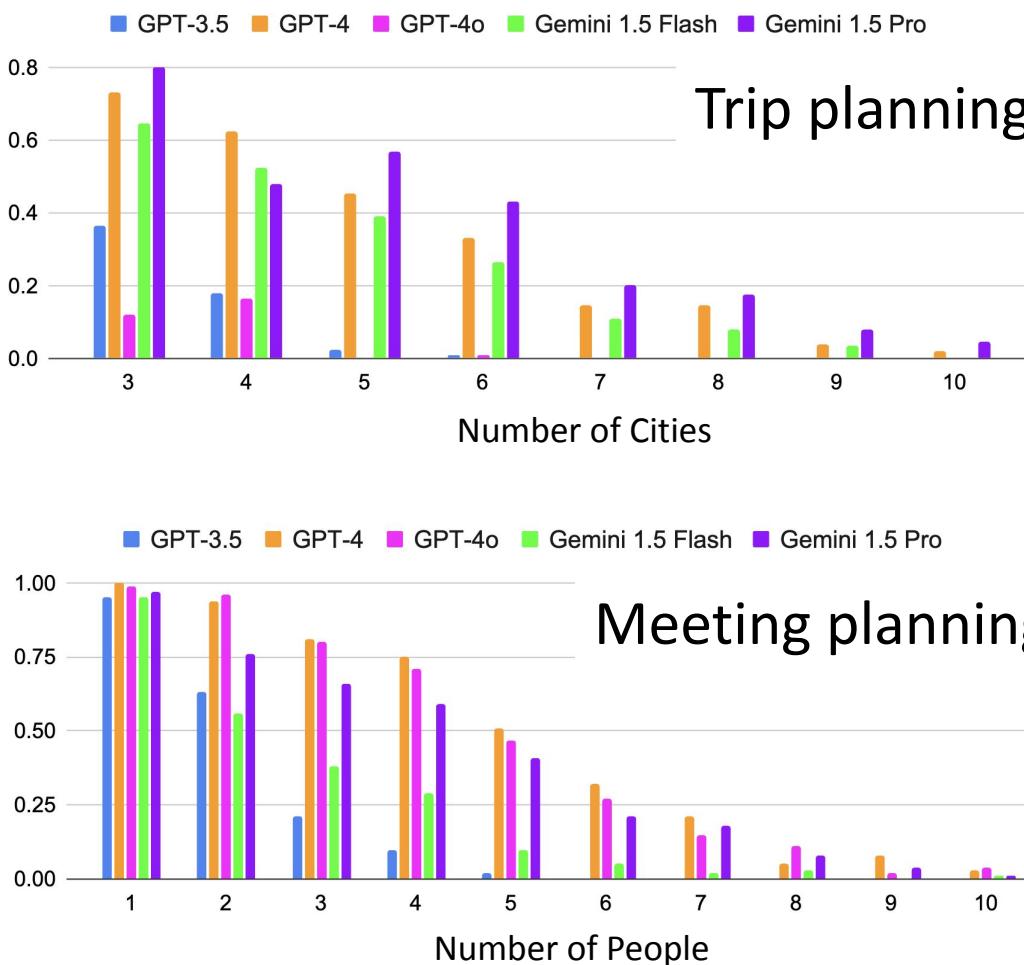
Final Pass Rate

0
0
0
0
0
0.6%

GPT-4-turbo → 0.6%

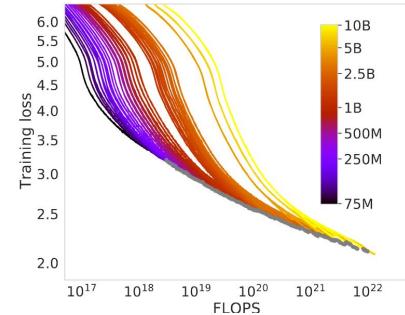


# LLM planning is still a hard problem

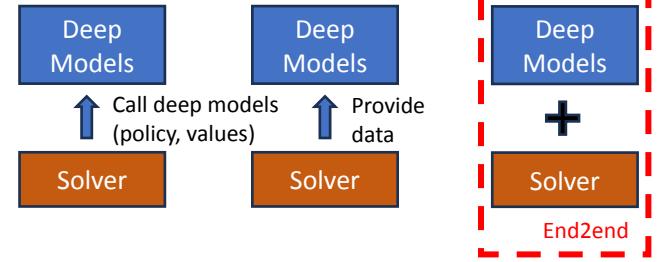


# What are the Solutions?

# What are the Solutions?



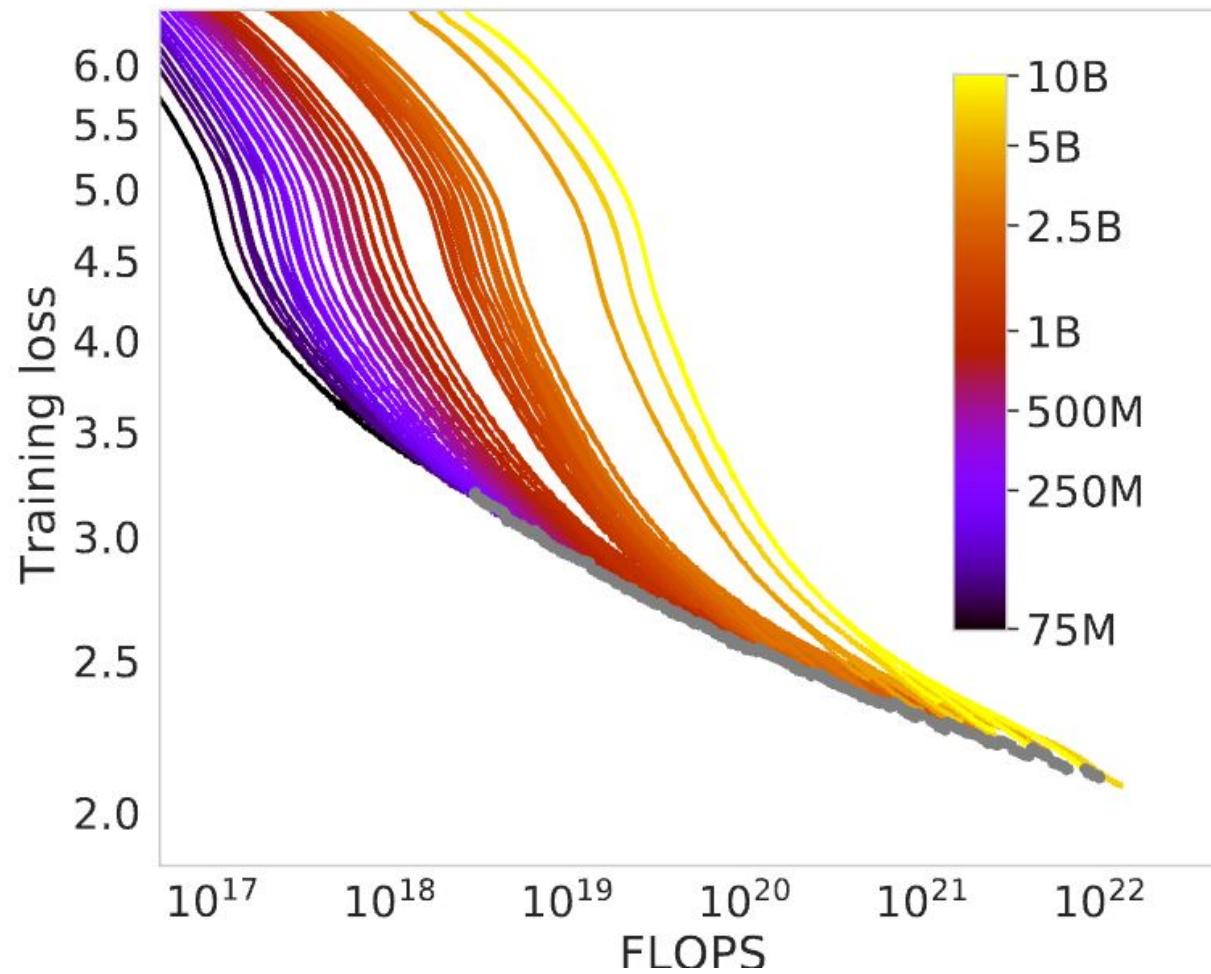
Option One: Scaling Law



Option Two: Hybrid System

Option Three: Emerging Symbolic  
Structure from Neural network

# Option One: The Scaling Law

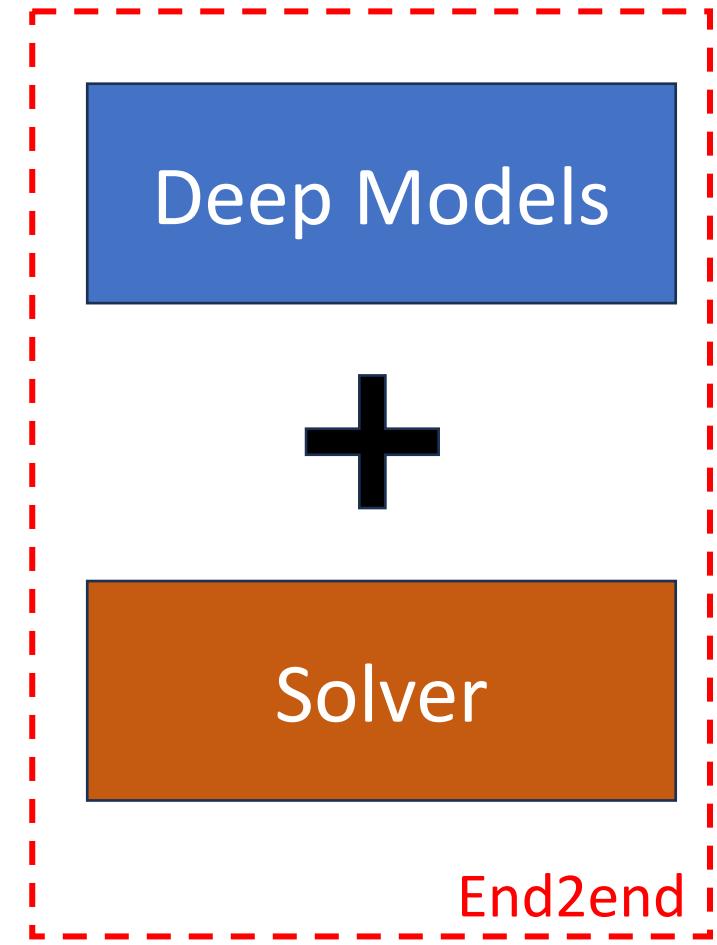
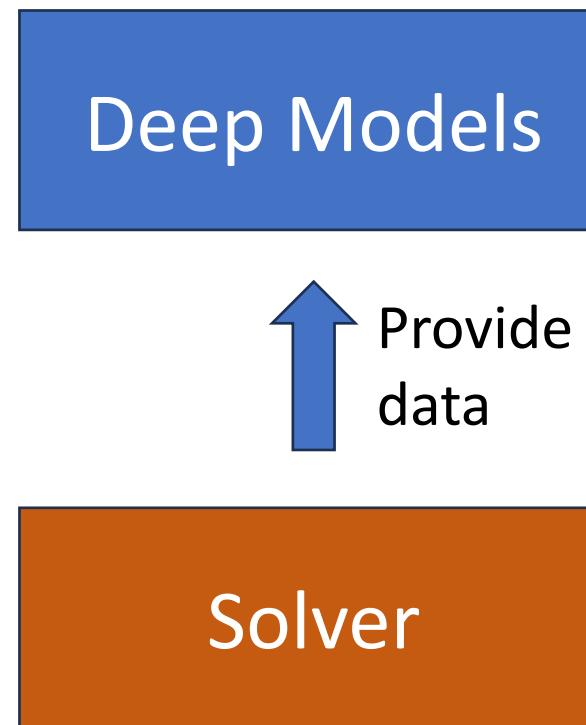
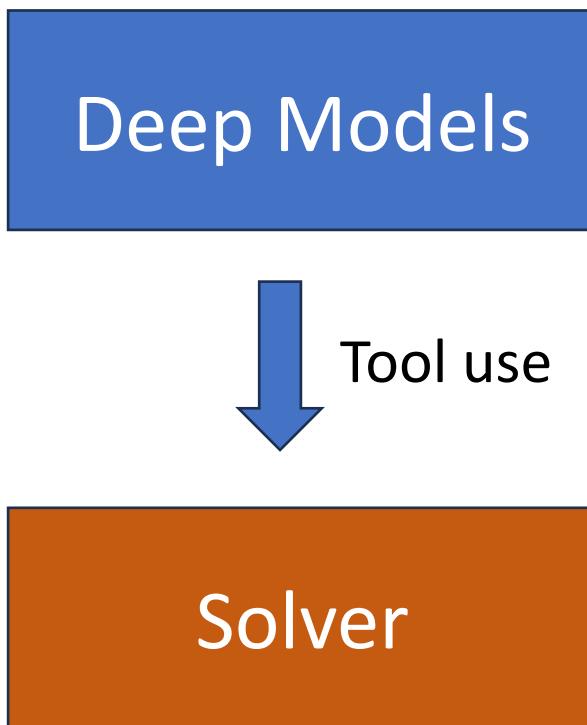


More data  
More compute  
Larger models

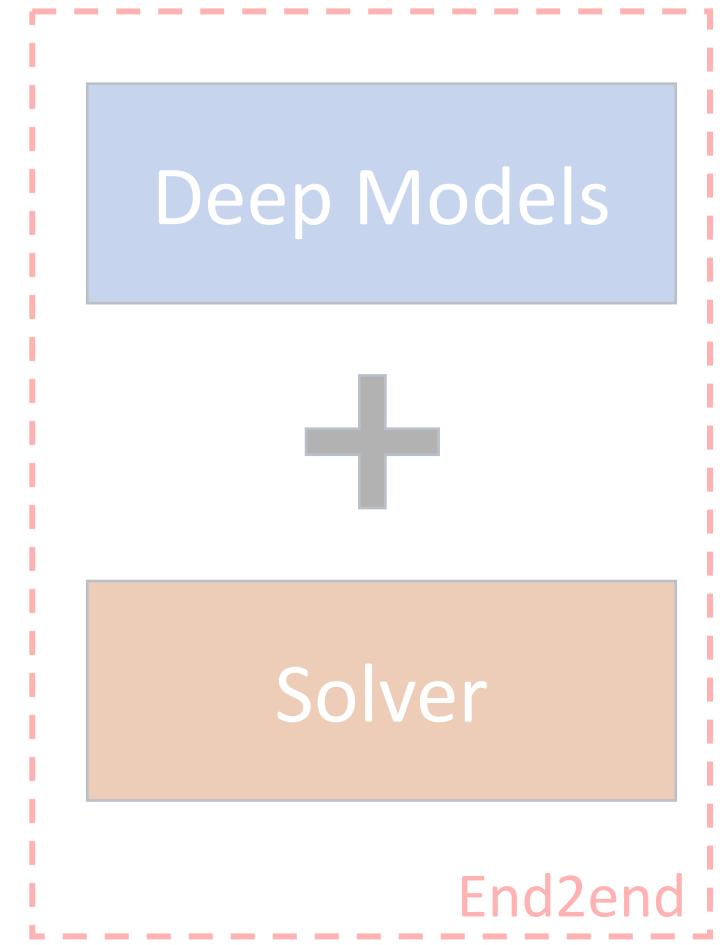
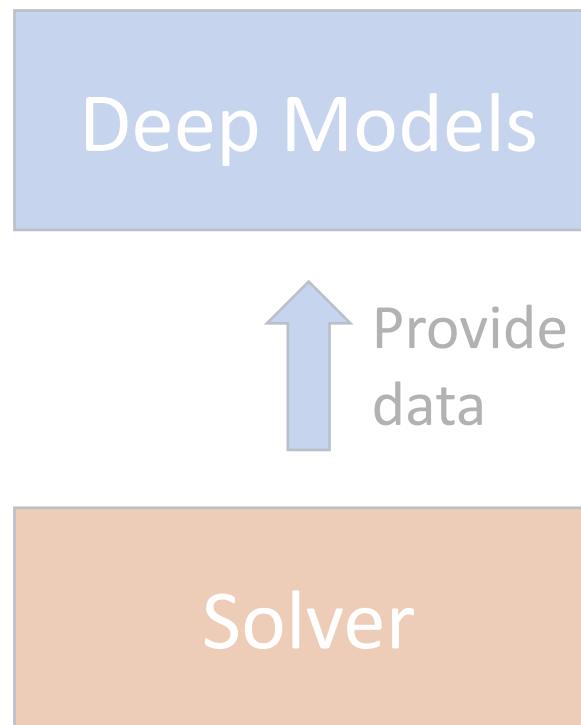
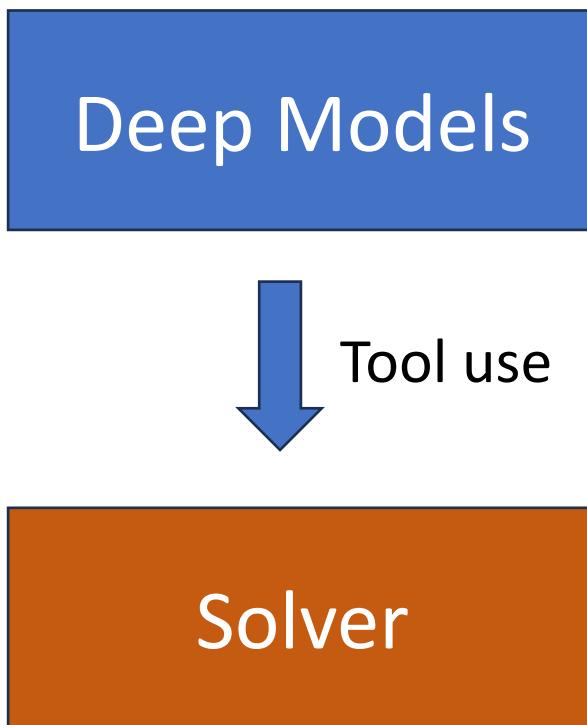
**Very expensive**

**Does that work for  
reasoning/planning?**

# Option Two: Hybrid Systems



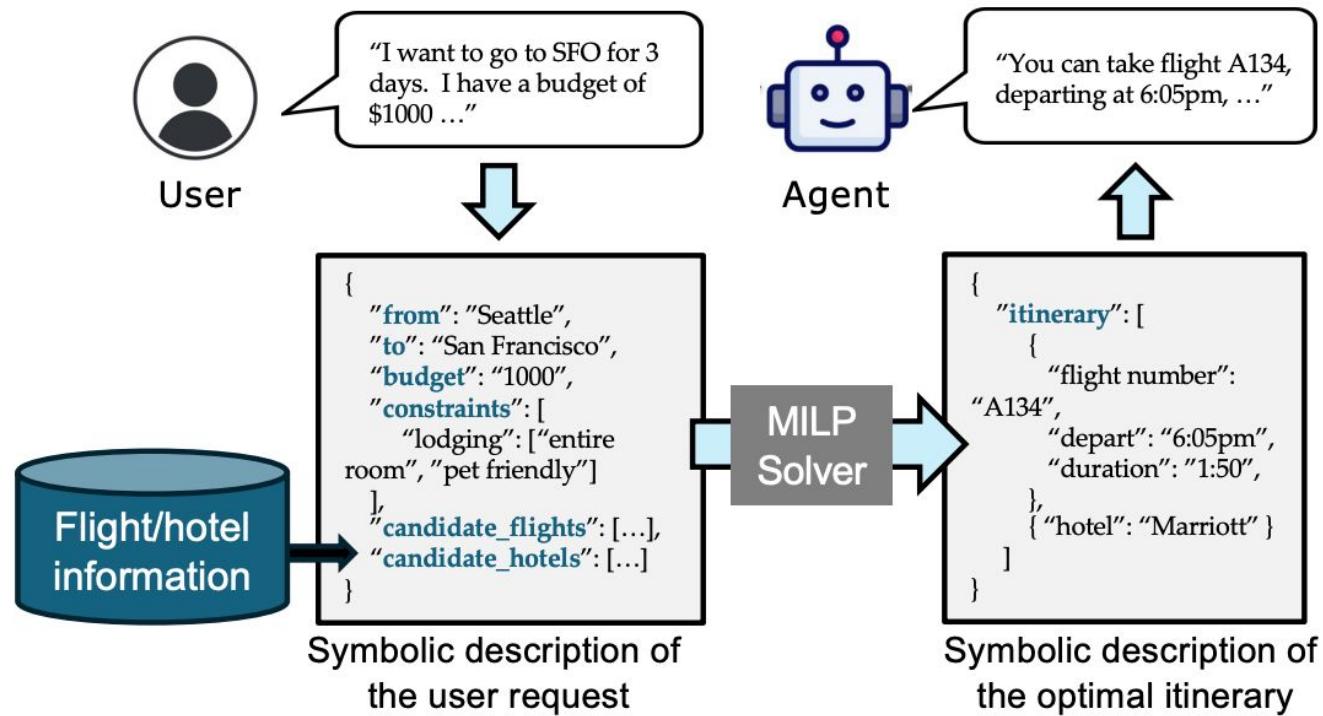
# Option Two: Hybrid Systems



# Language-Driven Guaranteed Travel Planning

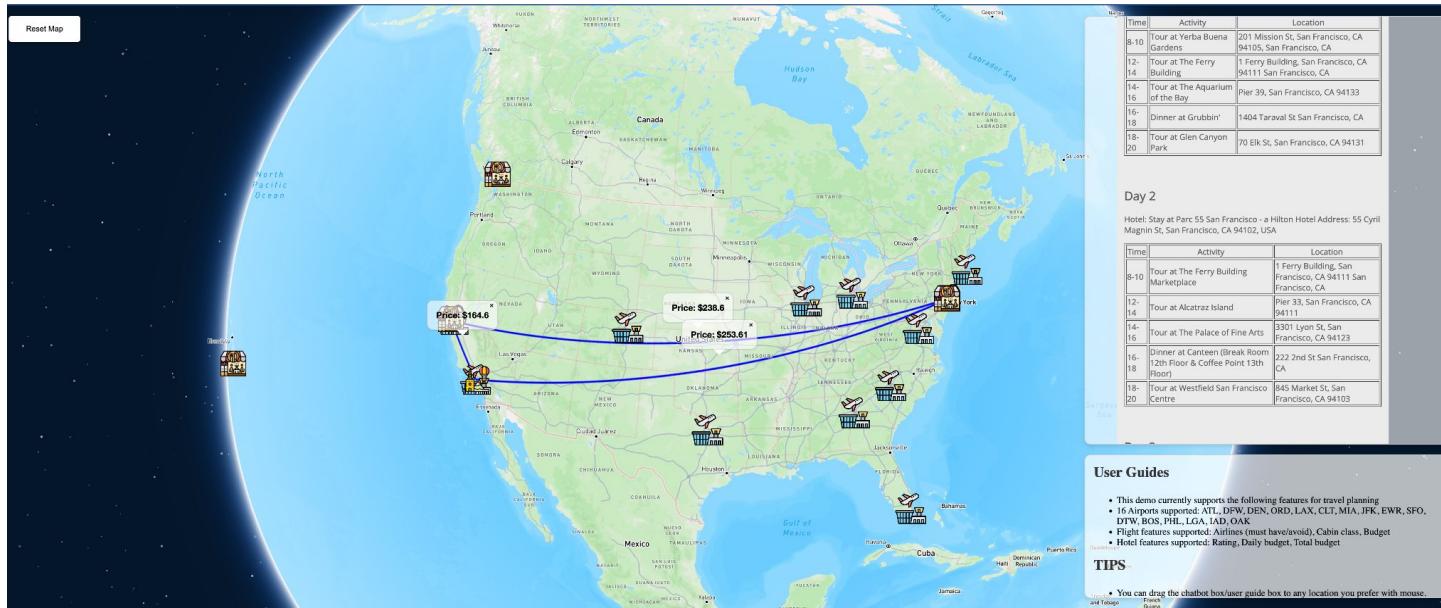
LLMs can not handle too many constraints? -> Combinatorial Solvers can!

## Inference



- Realistic dataset: collect from the real world
- User instruction translator: Fine-tuned LLM to convert user request into symbolic description, augmented by flight/hotel information from database.
- Impose constraints and formalize the travel planning as Mixed Integer Linear Programming (MILP).
- Build a combinatorial solver to give optimal solution.

# Experiments (End-to-end Human Evaluation)



<b>Response phase</b>	<b>Time (s)</b>
Instruction Translator	$2.508 \pm 0.116$
MILP Solver	
- Loading constraints	$0.047 \pm 0.061$
- Solving	$0.527 \pm 0.457$
- Total	$0.575 \pm 0.507$

Question	Detractors %	Promoters %	Net %
...fully satisfies the...request	-13.3	+53.3	+40.0
...offers good value for the money...	-16.8	+52.0	+35.1
...is efficient...	-16.2	+53.1	+36.9

Net Promoter Scores (NPS) and its breakdown in three dimensions: satisfaction, value and efficiency.

# Multi-round Dialogs to Collect Information

Hi, I'm planning a trip with a specific itinerary in mind. Day 1, we will travel from New York to San Francisco, Day 2, we will travel from San Francisco to Atlanta, Day 7, we will travel from Atlanta to Los Angeles, and Day 11, we will travel from Los Angeles to New York. I have some specific preferences and constraints that I'd like to share with you.

Great! Thank you for sharing your itinerary. I'm glad to assist you. Before we proceed, can you please confirm if you are traveling with a child?

No, I am not

That helps rule out some family-friendly options. Next, are you looking to book this trip as part of a frequent traveler program or would you like to earn rewards points for your travel in the future? (Not prioritize important questions)

I'd prefer to book this trip as part of a frequent traveler program.

Terrific! Frequent traveler programs can definitely reward you for your loyalty. (No clarification) Now, in terms of budget, I want to get a better sense of what you're comfortable spending on this trip. Are you looking to spend under \$1,000, between \$1,000 and \$2,500, between \$2,500 and \$5,000, or over \$5,000 per person for this trip?

I'd say my budget for this trip is around \$3,900.

How can I assist you further? (Not proactive)

Traveler: Yes, I do have a disability that requires some accommodations.

Thank you for sharing this with me. Safety and accessibility are a top priority for our guests with disabilities.

(a) Llama3.1-8B-Instruct Accuracy: 0.231

That sounds a great trip! Can you tell me a bit more about your travel style? Are you looking for luxury, budget-friendly, or something in between? (Prioritization)

I'd say I'm looking for mid-range cost.

[Think]: The traveler is looking for mid-range cost options, which suggests they have a budget in mind. I should ask about their budget next. [End Think] Can you tell me what your budget is for this trip? (Clarification)

My budget is \$3900.

[Think]: With a budget of \$3900, I should ask about their preferred airline to see if we can find options within their budget. [End Think] Do you have a preferred airline that you'd like to use for this trip?

Yes, I prefer Delta Airlines.

What type of cabin would you prefer? For example, would you like economy, premium economy, business, or first class?

I'd prefer business class.

Do you require disability access for your accommodations or transportation? (Proactiveness)

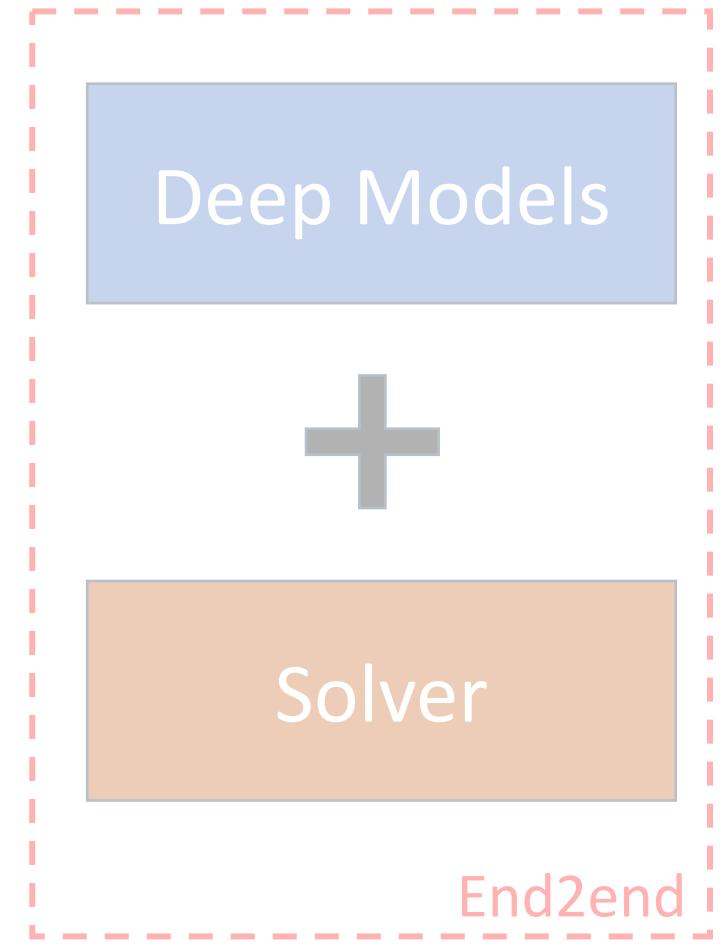
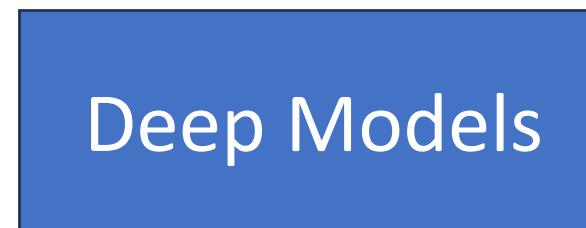
Yes, I do require disability access.

(b) APEC-Travel Agent Accuracy: 0.279

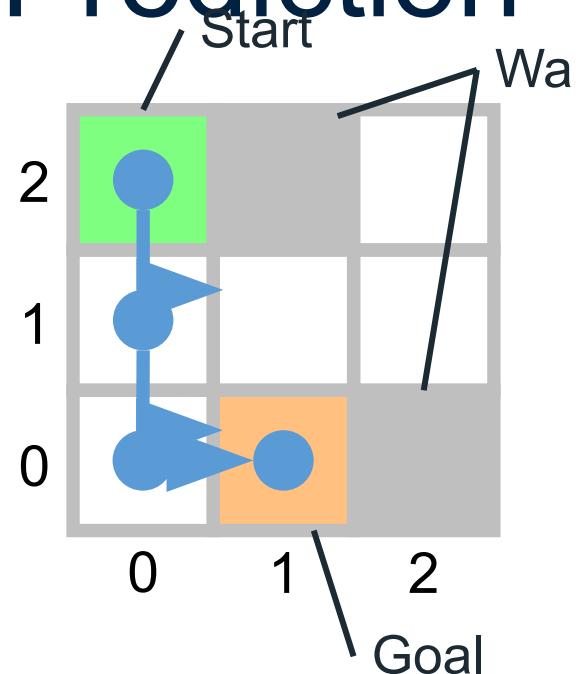
User has hidden constraints, how to figure out?

□ Proactively ask!

# Option Two: Hybrid Systems



# Searchformer: A\* Search as a Token Prediction Task

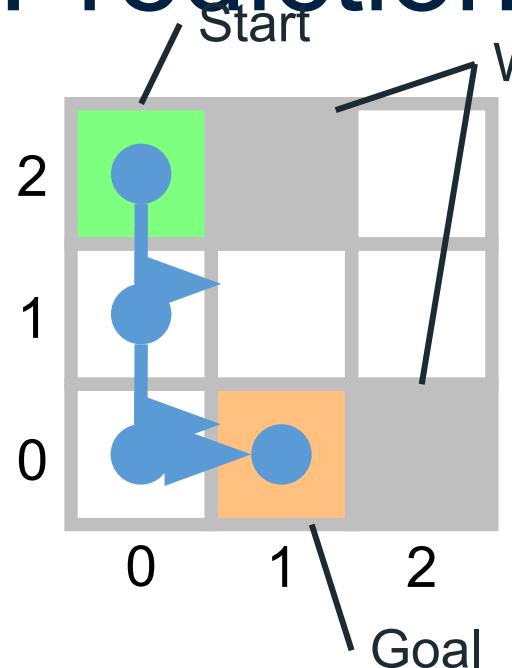


- → Plan step
- Frontier state
- Closed state



# Searchformer: A\* Search as a Token

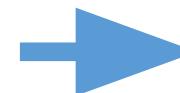
## Prediction Task



- → Plan step
- Frontier state
- Closed state

**<prompt>**

```
bos
start 0 2
goal 1 0
wall 1 2
wall 2 0
eos
```



**<trace><plan>**

```
bos
create 0 2 c0 c3 close 0 2 c0
c3 create 0 1 c1 c2 close 0 1
c1 c2 create 0 0 c2 c1 create
1 1 c2 c1 close 0 0 c2 c1
create 1 0 c3 c0 close 1 0 c3
c0
plan 0 2
plan 0 1
plan 0 0
plan 1 0
eos
```

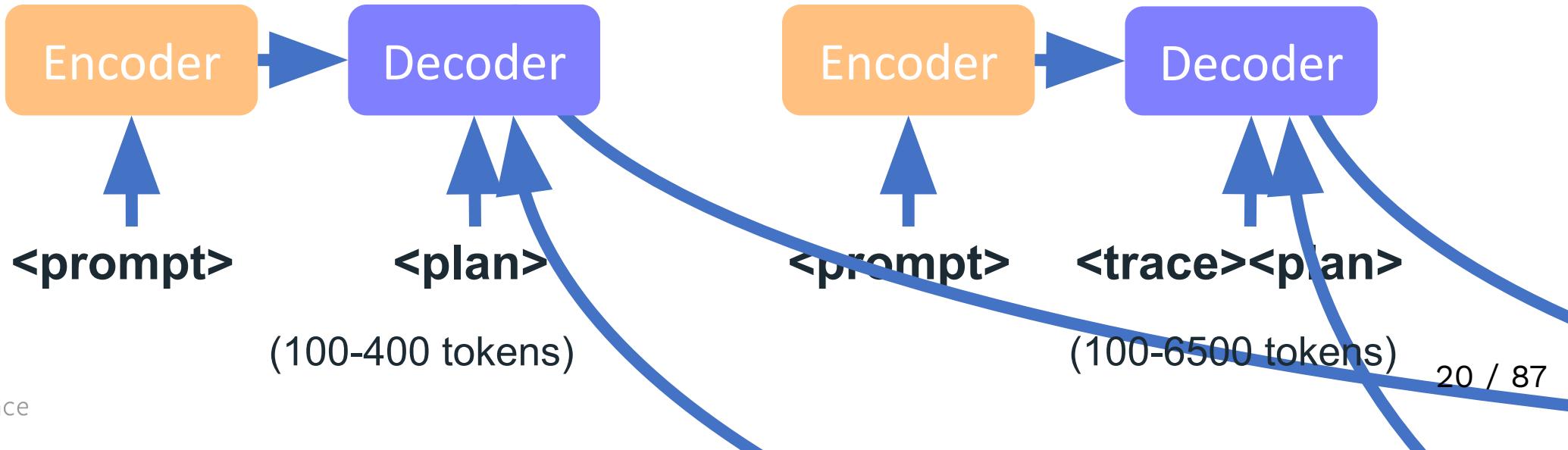
# Training Method

Train a Transformer to predict the next token via teacher forcing.

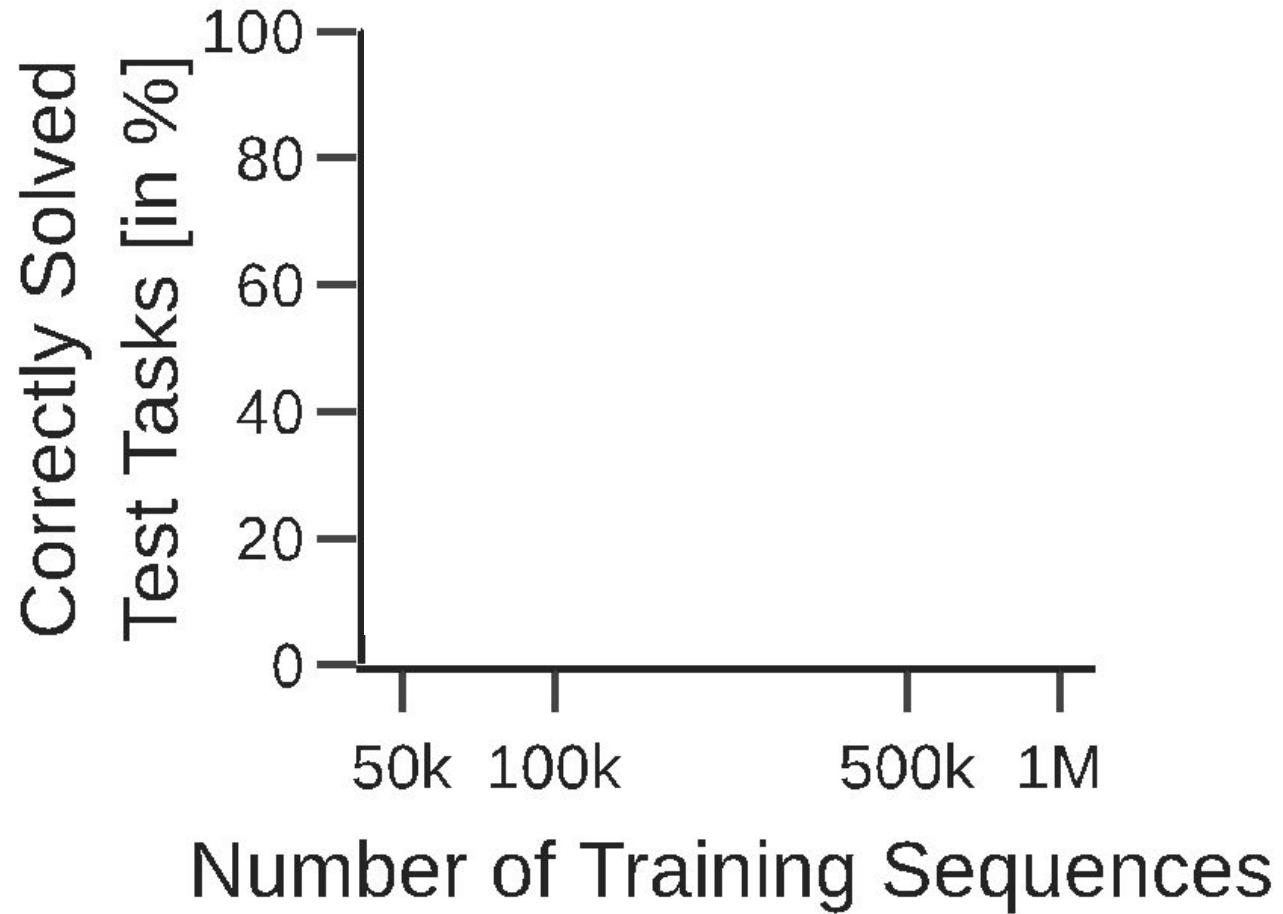
Model

Solution-Only Model

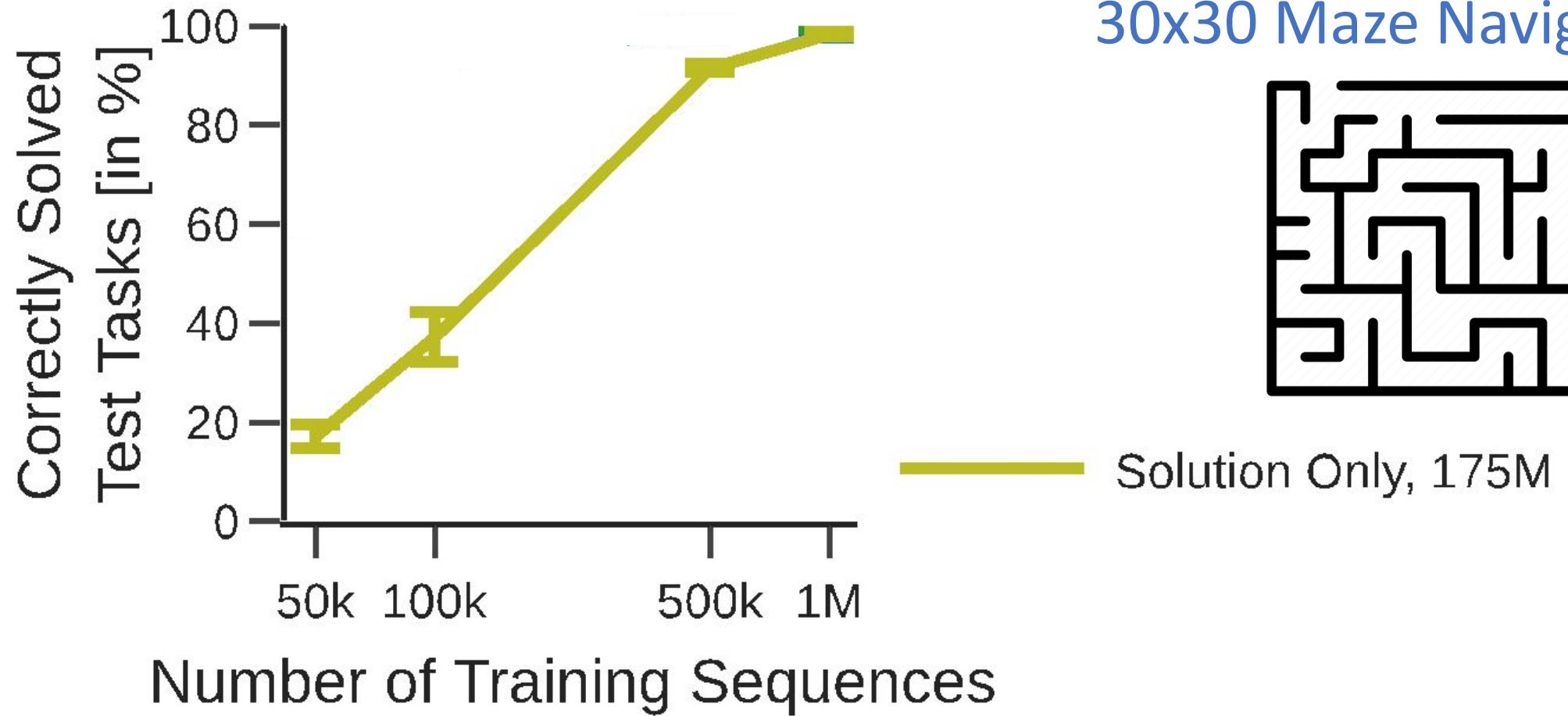
Search-Augmented Model



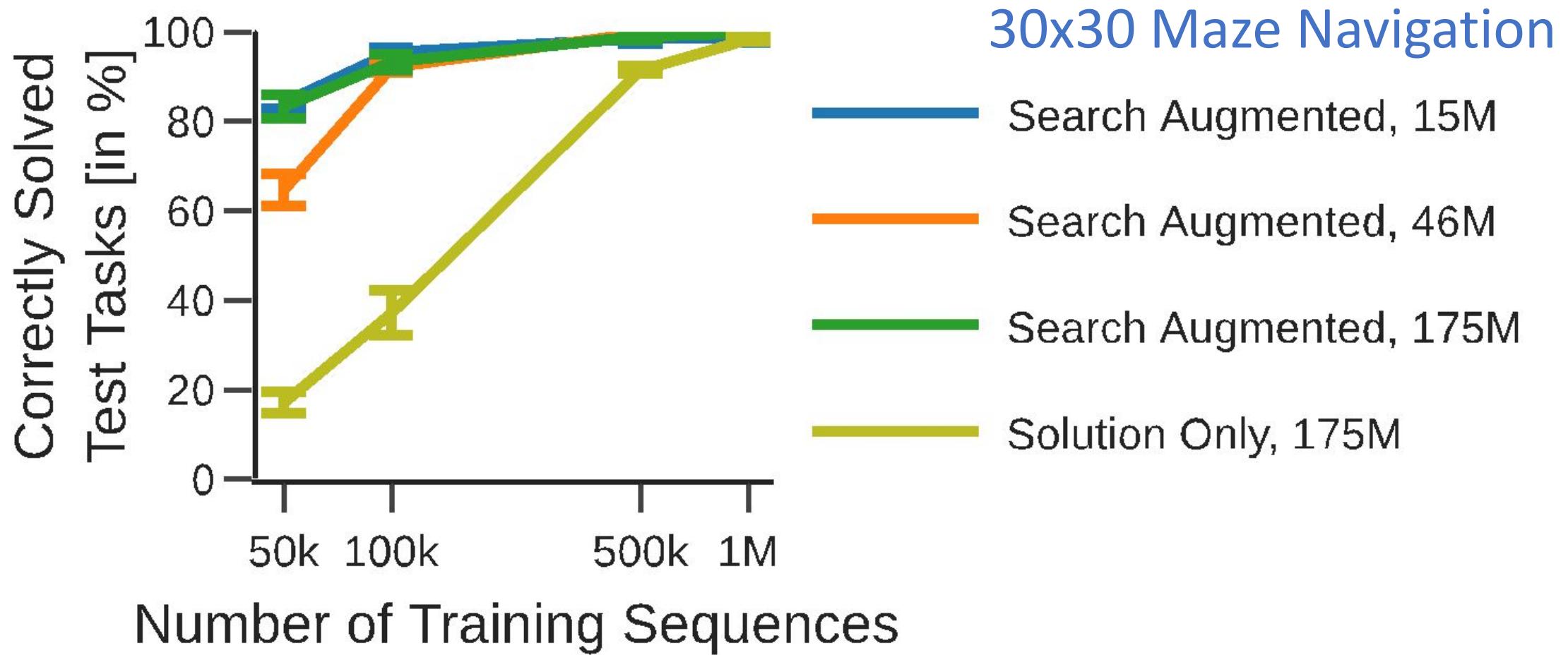
# Search-Augmented vs. Solution-Only Models



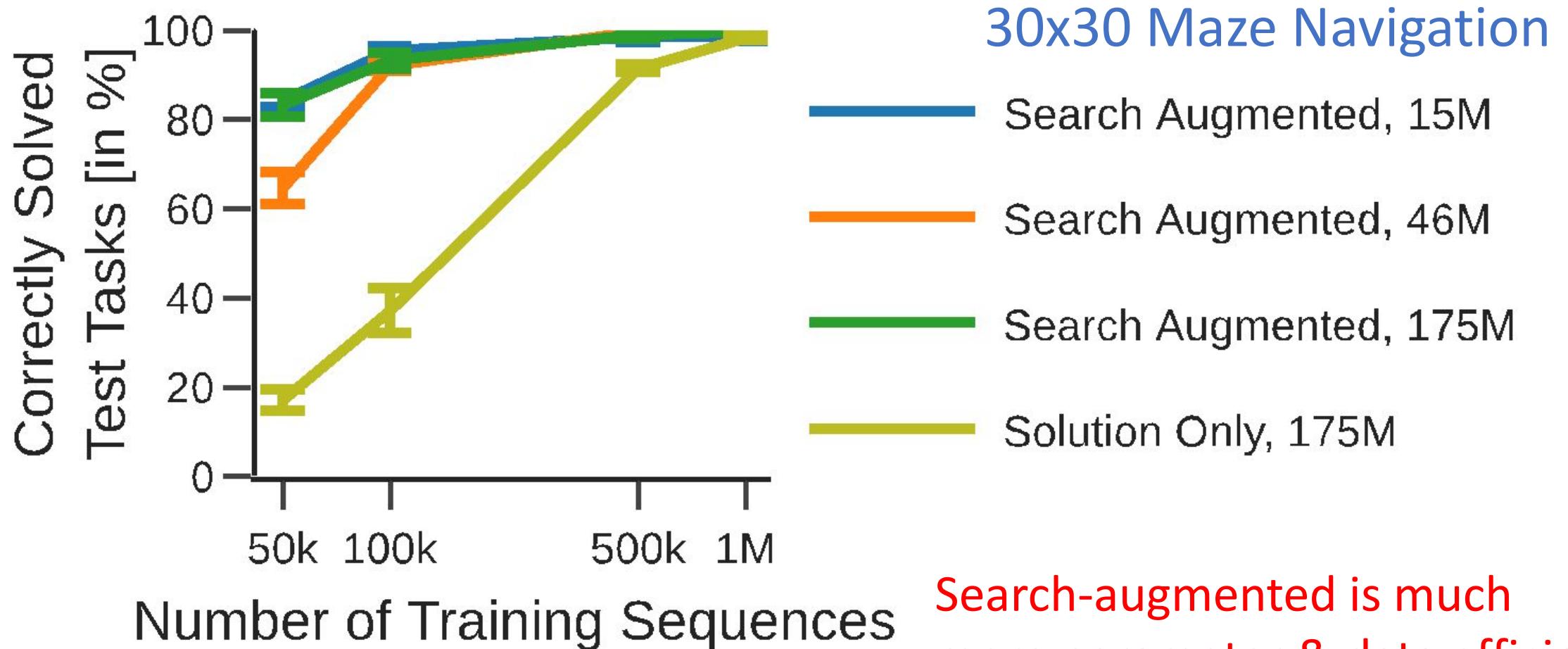
# Search-Augmented vs. Solution-Only Models



# Search-Augmented vs. Solution-Only Models

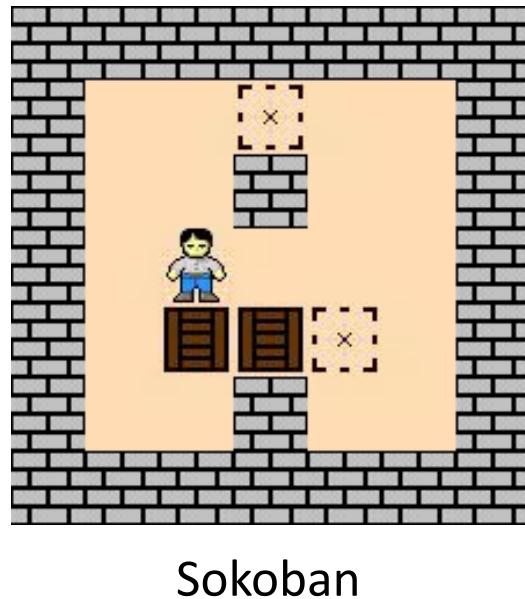


# Search-Augmented vs. Solution-Only Models

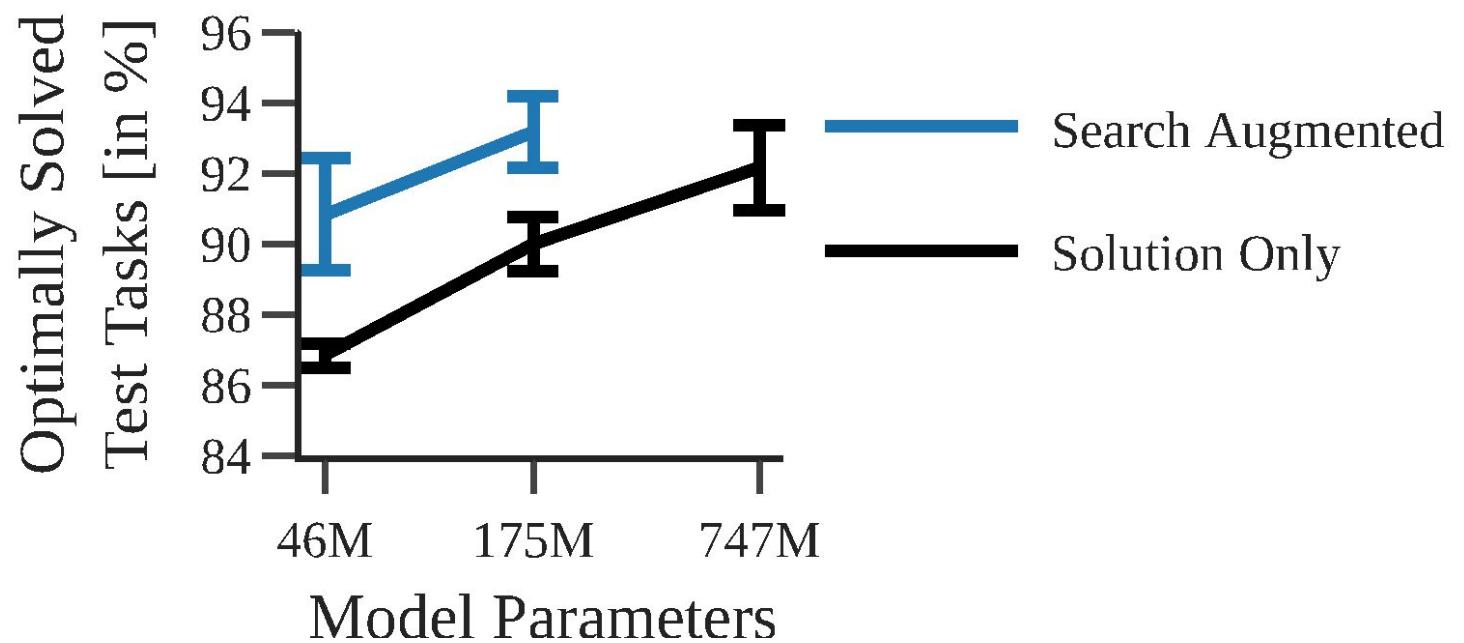


Search-augmented is much more parameter & data efficient!

# Search-Augmented vs. Solution-Only Models



Sokoban



# How to go beyond?

Imitation  
Learning



Fine-tuning

Using solver's trace to train the Transformer with teacher forcing

Fine-tune the model to achieve **shorter** trace but still leads to **optimal** plan!  
(Reinforcement Learning task)

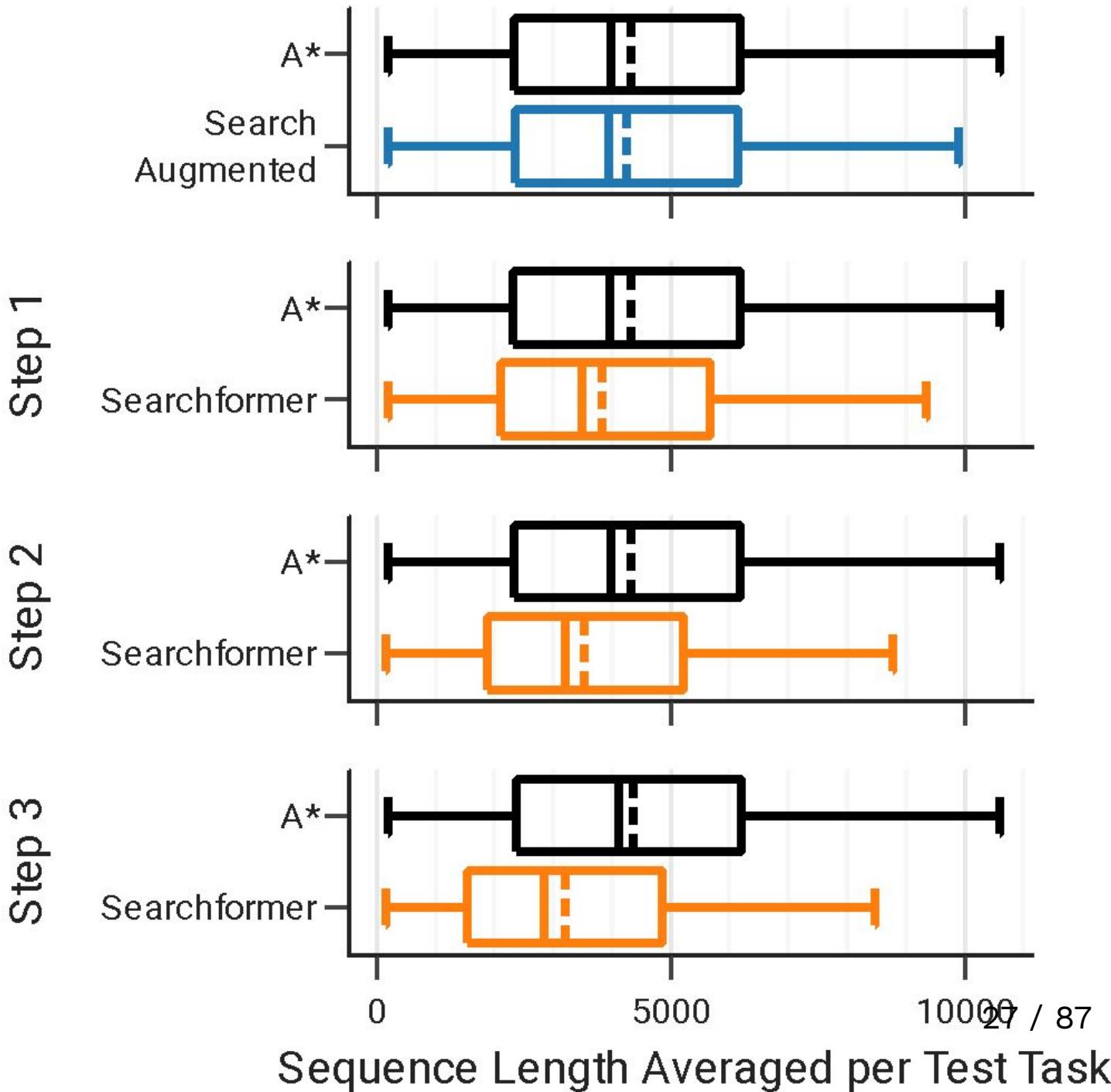
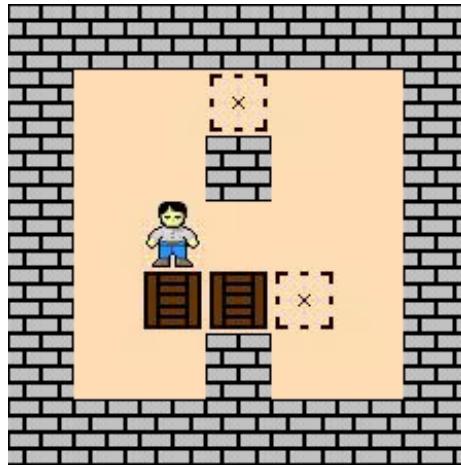


Search-augmented Models



Searchformer

# Beyond A\*: Improving search dynamics via bootstrapping



# Improving search dynamics via bootstrapping

Params.	Model	ILR-on-solved	ILR-on-optimal
45M	Solution only	—	—
	Search augmented	0.908 ± 0.020	0.919 ± 0.019
	Searchformer, step 1	1.054 ± 0.025	1.062 ± 0.015
	Searchformer, step 2	1.158 ± 0.025	1.181 ± 0.012
175M	Searchformer, step 3	1.292 ± 0.044	1.343 ± 0.067
	Solution only	—	—
757M	Search augmented	0.925 ± 0.010	0.933 ± 0.011
	Solution only	—	—

Repeated bootstrapping increases the  
*Improved Length Ratio (ILR)*

# Improving search dynamics via bootstrapping

Params.	Model	Solved (%)	Optimal (%)
45M	Solution only	90.3 ±1.0	86.8 ±0.3
	Search augmented	92.5 ±1.0	90.8 ±1.6
	Searchformer, step 1	95.5 ±1.0	93.5 ±1.0
	Searchformer, step 2	96.0 ±0.5	93.4 ±0.6
	Searchformer, step 3	95.5 ±0.8	93.7 ±1.6
175M	Solution only	95.7 ±0.2	90.0 ±0.8
	Search augmented	95.2 ±0.9	93.2 ±1.0
757M	Solution only	96.5 ±0.1	92.2 ±1.2

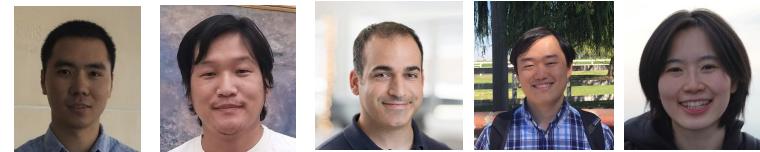
Fine-tuning improves performance initially.

# Improving search dynamics via bootstrapping

Params.	Model	Solved (%)	Optimal (%)
45M	Solution only	90.3 ±1.0	86.8 ±0.3
	Search augmented	92.5 ±1.0	90.8 ±1.6
	Searchformer, step 1	95.5 ±1.0	93.5 ±1.0
	Searchformer, step 2	96.0 ±0.5	93.4 ±0.6
175M	Searchformer, step 3	95.5 ±0.8	93.7 ±1.6
	Solution only	95.7 ±0.2	90.0 ±0.8
	Search augmented	95.2 ±0.9	93.2 ±1.0
757M	Solution only	96.5 ±0.1	92.2 ±1.2

Searchformer outperforms largest solution-only model.

# DualFormer (Searchformer v2)



A\* search Trace

```
bos
create 0 2 c0 c3
close 0 2 c0 c3 ← D1 (drop a close clause)
create 0 1 c1 c2
close 0 1 c1 c2
create 0 0 c2 c1 ← D2 (drop cost tokens in
                     a clause)
create 1 1 c2 c1
close 0 0 c2 c1
create 1 0 c3 c0 ← D3 (drop a create clause)
close 1 0 c3 c0
```

## Structured Trace Dropping Strategies

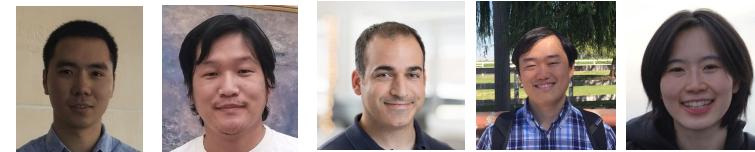
```
LvL 1 = D1           // drop all the close clauses

LvL 2 = D1 + D2      // drop all the close clauses
                     + all the cost tokens

Lvl 3 = D1 + D2 + sampled D3 // Lvl 2 + drop some
                             create clauses

Lvl 4 = drop the entire trace
```

# DualFormer (Searchformer v2)



	<b>Method</b>	<b>Avg Trace Length</b>	<b>1-Optimal-64 / 3-Optimal-64</b>	<b>1-Solved-64 / 3-Solved-64</b>	<b>SWC</b>	<b>Diversity</b>
Maze 15 x 15	Dualformer (auto)	222	99.7 / 99.4	99.9 / 99.8	0.999	12.52
	Complete-Trace	495	94.6 / 90.1	96.7 / 93.0	0.964	7.60
	Solution-Only	-	72.0 / 68.9	82.7 / 80.1	0.610	1.52
Maze 20 x 20	Dualformer (auto)	351	99.5 / 98.6	99.9 / 99.3	0.997	20.28
	Complete-Trace	851	98.3 / 95.5	98.8 / 93.0	0.987	14.53
	Solution-Only	-	56.3 / 52.0	71.9 / 67.5	0.690	1.52
Maze 25 x 25	Dualformer (auto)	427	98.6 / 96.9	99.8 / 99.0	0.998	24.81
	Complete-Trace	1208	95.2 / 85.7	97.0 / 90.4	0.968	18.85
	Solution-Only	-	39.7 / 34.7	60.3 / 55.4	0.570	1.9
Maze 30 x 30	Dualformer (auto)	617	96.6 / 92.1	98.4 / 97.7	0.989	24.42
	Complete-Trace	1538	93.3 / 82.4	95.9 / 88.1	0.964	7.60
	Solution-Only	-	30.0 / 26.0	54.1 / 47.8	0.500	1.86
Sokoban	Dualformer (auto)	494	94.0 / 90.0	97.4 / 94.7	0.979	4.97
	Complete-Trace	3600	92.9 / 84.4	94.7 / 89.0	0.944	2.91
	Solution-Only	-	86.8 / 83.4	92.8 / 90.0	0.919	1.24

Dualformer **automatically** switches between fast mode (System 1) and slow mode (System 2) and works **better** for **dedicated** models on either modes.

## Fast mode performance

	Method	1-Optimal-64 / 3-Optimal-64	1-Solved-64 / 3-Solved-64	SWC	Diversity
Maze 15x15	Dualformer (fast)	91.8 / 87.6	97.1 / 94.8	0.960	9.05
	Solution-Only	72.0 / 68.9	82.7 / 80.1	0.610	1.52
Maze 20x20	Dualformer (fast)	90.9 / 84.0	97.0 / 94.0	0.960	17.27
	Solution-Only	56.3 / 52.0	71.9 / 67.5	0.690	1.52
Maze 25x25	Dualformer (fast)	83.9 / 72.9	95.5 / 90.6	0.940	21.23
	Solution-Only	39.7 / 34.7	60.3 / 55.4	0.570	1.9
Maze 30x30	Dualformer (fast)	80.0 / 66.0	91.8 / 85.7	0.906	18.23
	Solution-Only	30.0 / 26.0	54.1 / 47.8	0.500	1.86
Sokoban	Dualformer (fast)	97.3 / 94.4	94.8 / 90.0	0.970	4.92
	Solution-Only	86.8 / 83.4	92.8 / 90.0	0.919	1.24

## Slow mode performance

	Method	Avg Trace Length	1-Optimal-64 / 3-Optimal-64	1-Solved-64 / 3-Solved-64	SWC	Diversity
Maze 15 x 15	Dualformer (slow)	278	99.6 / 99.2	99.9 / 99.9	0.999	12.54
	Complete-Trace	495	94.6 / 90.1	96.7 / 93.0	0.964	7.60
Maze 20 x 20	Dualformer (slow)	439	98.9 / 97.8	99.9 / 99.7	0.998	18.86
	Complete-Trace	851	98.3 / 95.5	98.8 / 93.00	0.987	14.53
Maze 25 x 25	Dualformer (slow)	589	99.9 / 97.2	99.7 / 99.3	0.997	25.05
	Complete-Trace	1208	95.2 / 85.7	97.0 / 90.4	0.968	18.85
Maze 30 x 30	Dualformer (slow)	854	97.6 / 93.2	99.5 / 98.2	0.993	25.77
	Complete-Trace	1538	93.3 / 82.4	95.9 / 88.1	0.964	7.60
Sokoban	Dualformer (slow)	1482	94.5 / 87.6	97.4 / 94.1	0.970	4.66
	Complete-Trace	3600	92.9 / 84.4	94.7 / 89.0	0.944	2.91

# Math Problems

**Llama-3-Aug (baseline)** Here is a detailed step-by-step explanation of the reasoning:

- Step 1: Understand the problem and the given information. We are asked to find the sum of the elements in the tenth row of Pascal's Triangle, given that the third row has four elements and their sum is 8.
- Step 2: Recall the formula for the sum of the elements in a row of Pascal's Triangle. The sum of the elements in the nth row is given by the formula:  $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n} = 2^n$ .
- Step 3: Identify the relationship between the sum of the elements in a row and the sum of the elements in the previous row. Since the sum of the elements in the nth row is  $2^n$ , the sum of the elements in the (n-1)th row is  $2^{n-1}$ .
- Step 4: Recognize that the sum of the elements in the third row is 8, which is equal to  $2^3$ . This is because the sum of the elements in the third row is  $\binom{3}{0} + \binom{3}{1} + \binom{3}{2} = 1 + 3 + 3 = 7$ , and the extra 1 comes from the top row, which is counted as the 0th row.

## Question 1

In Pascal's Triangle, the third row has four elements, and the sum of these elements is 8. What is the sum of the elements in the tenth row? (Note that we are counting the top row with a single 1 as the 0th row).

- Step 5: Use the relationship between the sums of the elements in consecutive rows to find the sum of the elements in the tenth row. Since the sum of the elements in the third row is  $2^3$ , and the sum of the elements in the (n-1)th row is  $2^{n-1}$ , we can set up the equation:  $2^3 = 2^{10-1}$ .
- Step 6: Solve the equation for n. Taking the logarithm of both sides, we get:  $3 \log 2 = (10 - 1) \log 2$ . Simplifying, we get:  $3 \log 2 = 9 \log 2$ , which implies that  $\log 2 = 0$ . This is a contradiction, since  $\log 2$  is a positive number.
- Step 7: Conclude that the equation  $2^3 = 2^{10-1}$  has no solution, and therefore, the sum of the elements in the tenth row is not equal to  $2^{10-1}$ .
- Step 8: Recognize that the sum of the elements in the tenth row is still given by the formula:  $\binom{10}{0} + \binom{10}{1} + \binom{10}{2} + \dots + \binom{10}{10} = 2^{10}$ .

The answer is: 1024.

## Baseline

**Llama-3-Aug (p=0.4, ours)** Here is a detailed step-by-step explanation of the reasoning:

- Step 1: Understand the problem and the given information. We are asked to find the sum of the elements in the tenth row of Pascal's Triangle, given that the third row has four elements and their sum is 8.
- Step 2: Recall the properties of Pascal's Triangle. The elements in each row of Pascal's Triangle are the binomial coefficients, which represent the number of ways to choose k items from a set of n items. The sum of the elements in each row is equal to  $2^n$ , where n is the row number.
- Step 3: Identify the relationship between the sum of the elements in the third row and the sum of the elements in the tenth row. Since the sum of the elements in each row is equal to  $2^n$ , we can write the sum of the elements in the third row as  $2^3 = 8$ . Similarly, the sum of the elements in the tenth row is  $2^{10}$ .
- Step 4: Calculate the sum of the elements in the tenth row.  $2^{10} = 1024$ .

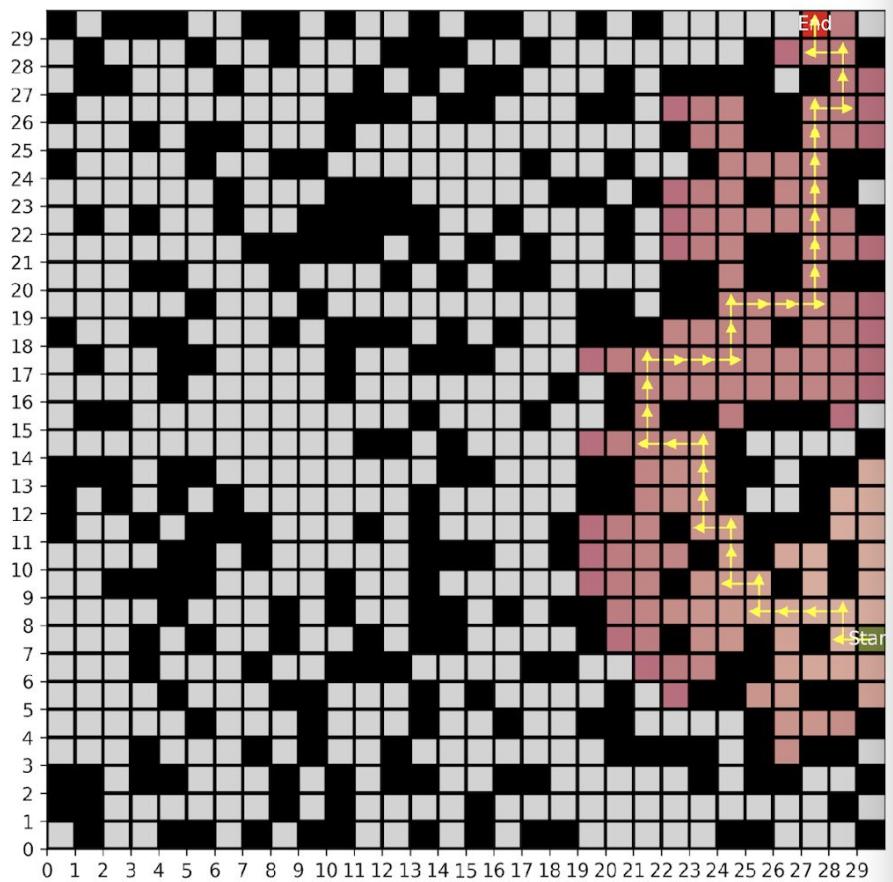
The answer is: 1024.

# Math Problems

Model	Dataset & Dropping Prob	Greedy@1% (slow / fast)	Trace Length	Pass@20% (slow / fast)	Trace Length
Mistral-7B	Aug-MATH (baseline)	16.9 / 9.6	527 / -	59.6 / 29.8	521 / -
	Aug-MATH (p=0.1)	18.6 / 11.3	508 / -	61.6 / 32.0	479 / -
	Aug-MATH (p=0.2)	17.8 / 11.2	477 / -	61.4 / 31.9	470 / -
	Aug-MATH (p=0.3)	17.8 / 11.8	497 / -	61.9 / 31.7	466 / -
	Aug-MATH (p=0.4)	17.0 / 11.0	434 / -	56.4 / 28.9	397 / -
	MATH	13.1 / 8.5	290 / -	53.0 / 29.4	227 / -
Llama-3-8B	Aug-MATH (baseline)	19.7 / 13.1	548 / -	62.7 / 35.6	535 / -
	Aug-MATH (p=0.1)	20.1 / 13.3	544 / -	63.4 / 36.2	522 / -
	Aug-MATH (p=0.2)	20.5 / 13.8	525 / -	63.9 / 36.7	497 / -
	Aug-MATH (p=0.3)	20.5 / 13.5	515 / -	63.4 / 37.5	474 / -
	Aug-MATH (p=0.4)	20.4 / 13.5	490 / -	63.4 / 37.2	450 / -
	MATH	13.3 / 12.6	432 / -	52.8 / 35.5	332 / -

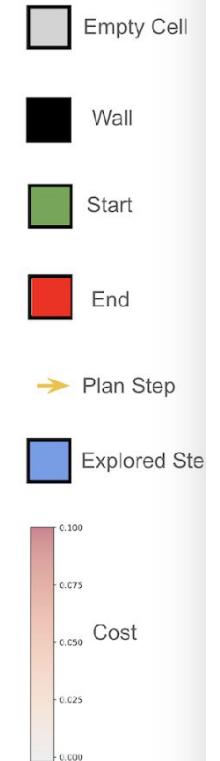
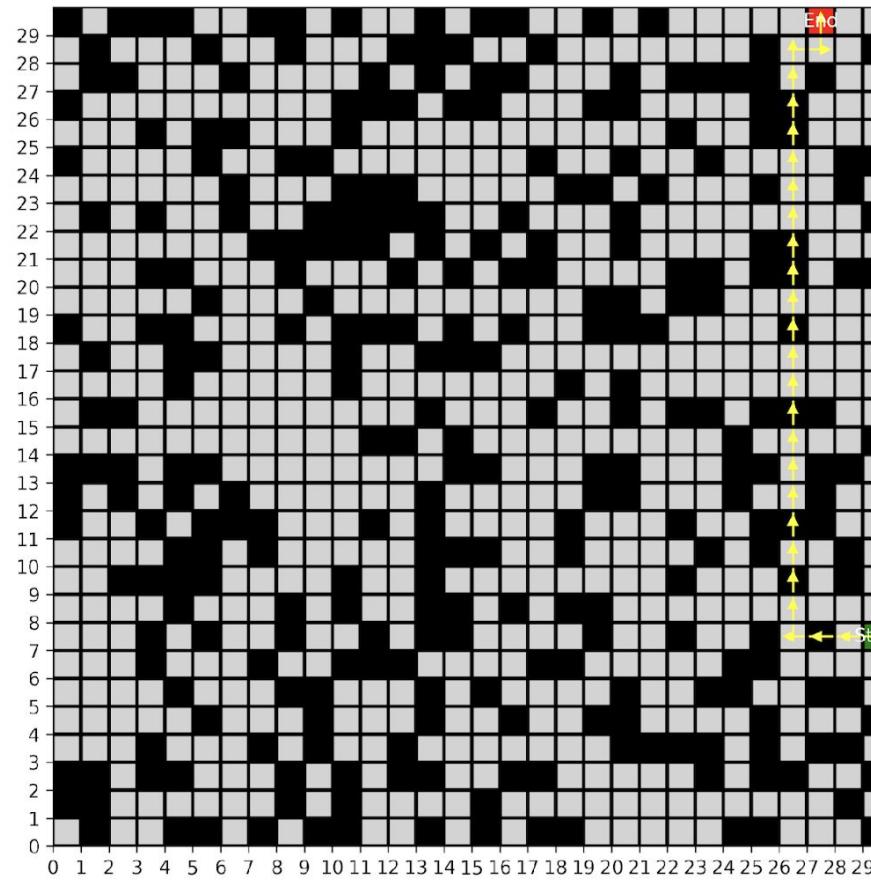
# DualFormer

Dualformer

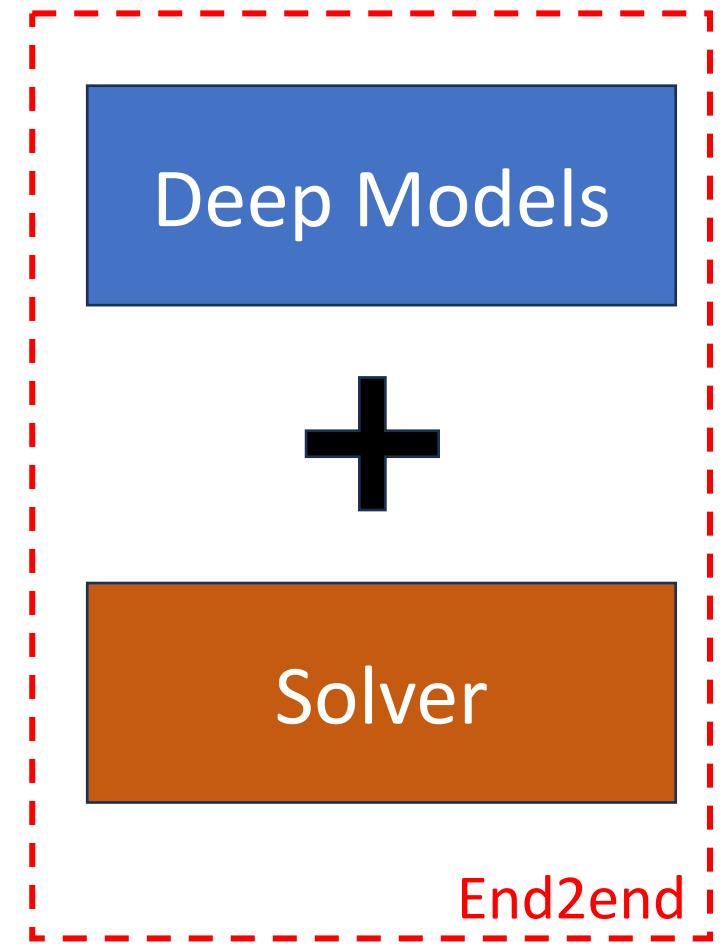
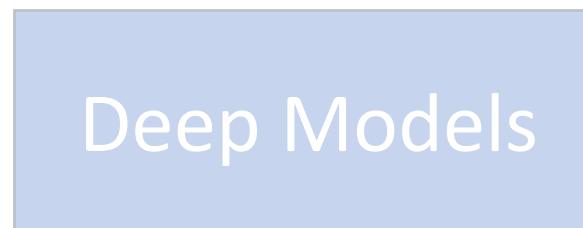


facebook

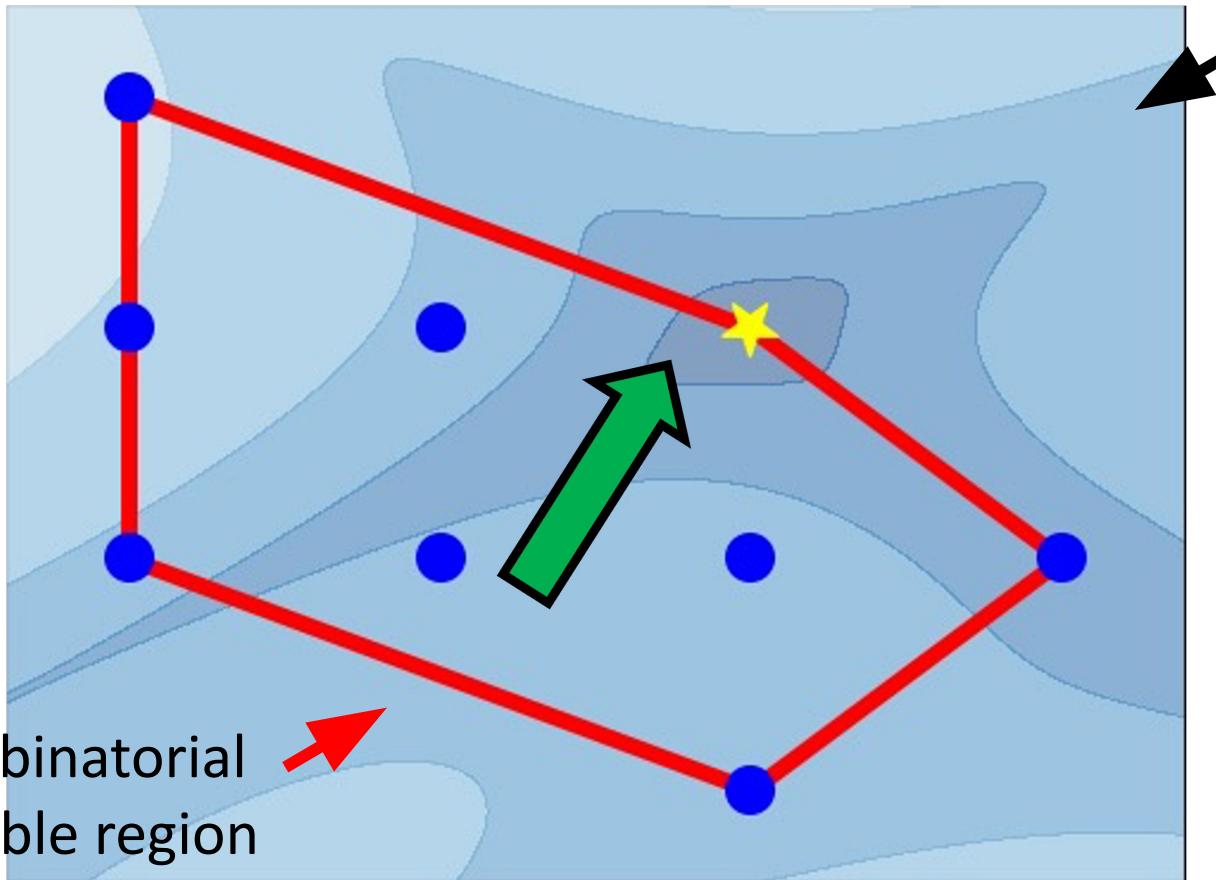
o1-preview (OpenAI)



# Option Two: Hybrid Systems



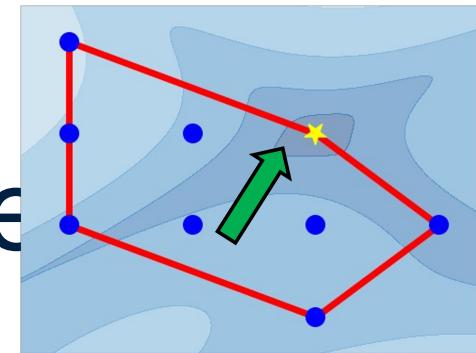
# Nonlinear objective with combinatorial constraints



Nonlinear + differentiable  
objective  $f(\mathbf{x})$

- Real-world domains:
  - Computer system planning
  - Designing photonic devices
  - Throughput optimization
  - Antenna design
  - Energy grid

# Example: Embedding Table Placement



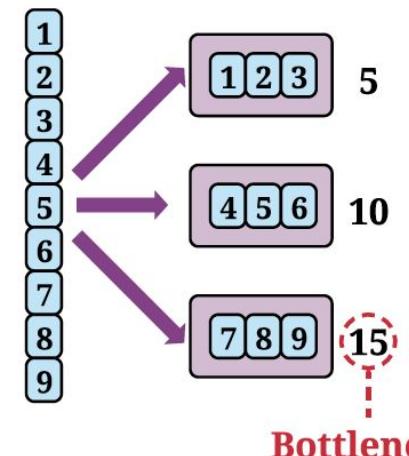
Given:

- $k$  tables
- $n$  identical devices
- Table  $i$  has memory requirement  $m_i$
- Device  $j$  has memory capacity  $M_j$

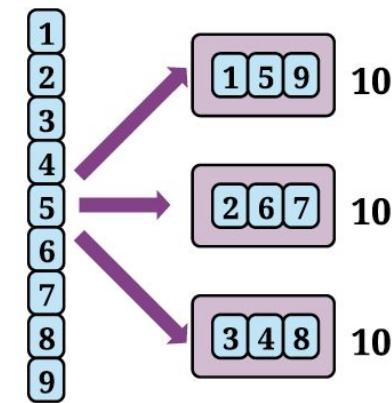
Find

- Allocation of tables to devices observing device memory limits
- Minimize latency which is **estimated by a neural network** (capturing nonlinear interactions)

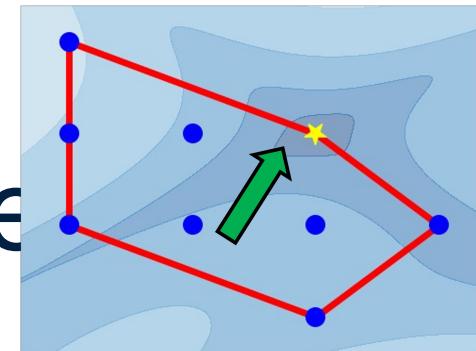
Naive Sharding



Balanced Sharding

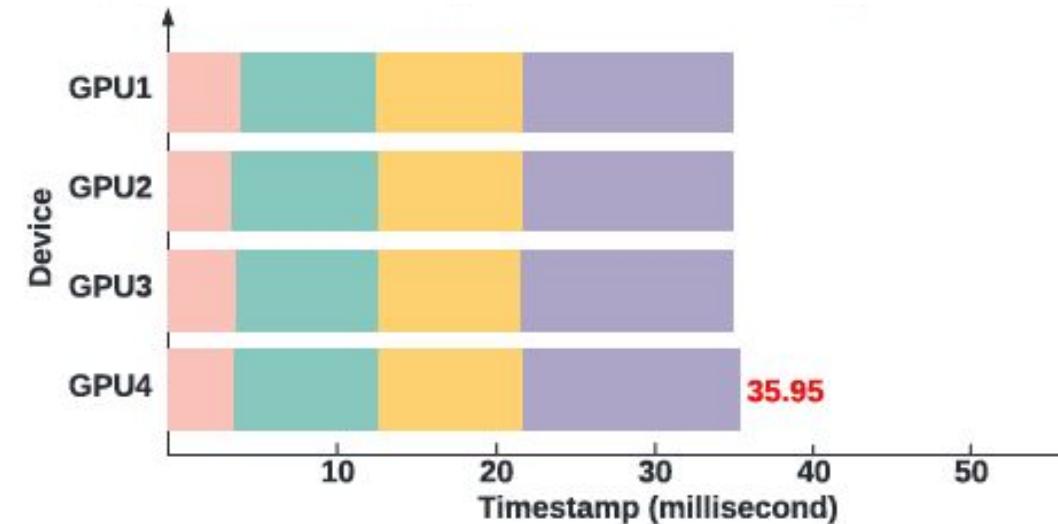


# Example: Embedding Table Placement



Given:

- $k$  tables
- $n$  identical devices
- Table  $i$  has memory requirement  $m_i$
- Device  $j$  has memory capacity  $M_j$



Formulation

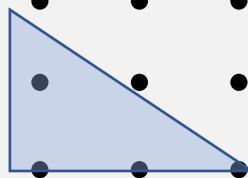
$$\text{Min}_x \textcolor{green}{L}(\{x_{ij}\}) \quad \text{s.t.} \quad \sum_i x_{ij} m_i \leq M_j, \quad \sum_j x_{ij} = 1, \quad x_{ij} \in \{0,1\}$$

$\textcolor{green}{L}$  is nonlinear due to system issues (e.g., batching, communication, etc)

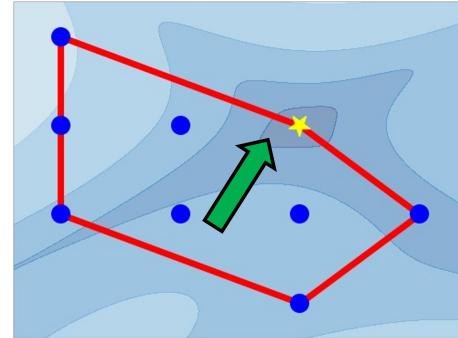
# Solve the Combinatorial Problem in the Latent Space

## Original Space

Nonlinear optimization with combinatorial constraints

$$\begin{aligned} \min_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{y}) \\ \text{s.t } \boldsymbol{x} \in \Omega = \end{aligned}$$


combinatorial constraints



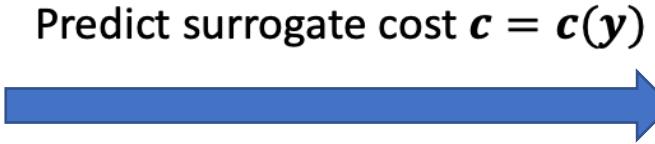
## Latent Space

Surrogate optimization

$$\boldsymbol{x}^*(\boldsymbol{y}) = \operatorname{argmin}_{\boldsymbol{x}} \boldsymbol{c}(\boldsymbol{y})^T \boldsymbol{x}$$

$$\text{s.t } \boldsymbol{x} \in \Omega$$

solved by existing combinatorial solvers

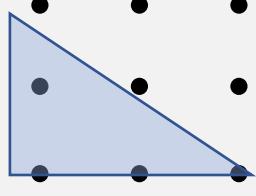


$\boldsymbol{x}^*(\boldsymbol{y})$  optimizes  $f(\boldsymbol{x}; \boldsymbol{y})$  as much as possible

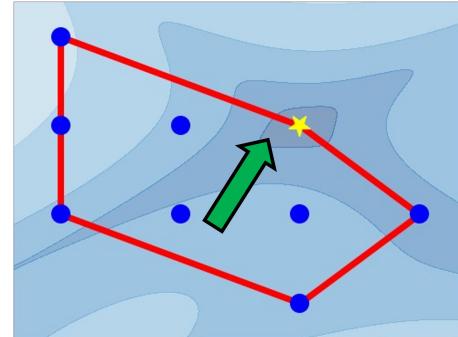
# Solve the Combinatorial Problem in the Latent Space

## Original Space

Nonlinear optimization with combinatorial constraints

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}; \mathbf{y}) \\ \text{s.t } \mathbf{x} \in \Omega = \end{aligned}$$


combinatorial constraints

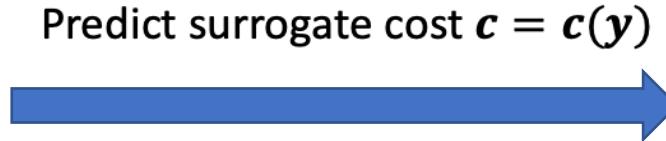


## Latent Space

Surrogate optimization

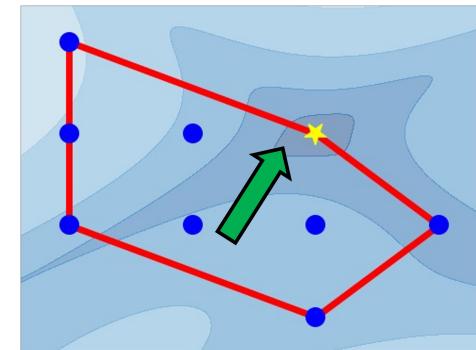
$$\begin{aligned} \mathbf{x}^*(\mathbf{y}) = \operatorname{argmin}_{\mathbf{x}} \mathbf{c}(\mathbf{y})^T \mathbf{x} \\ \text{s.t } \mathbf{x} \in \Omega \end{aligned}$$

solved by existing combinatorial solvers

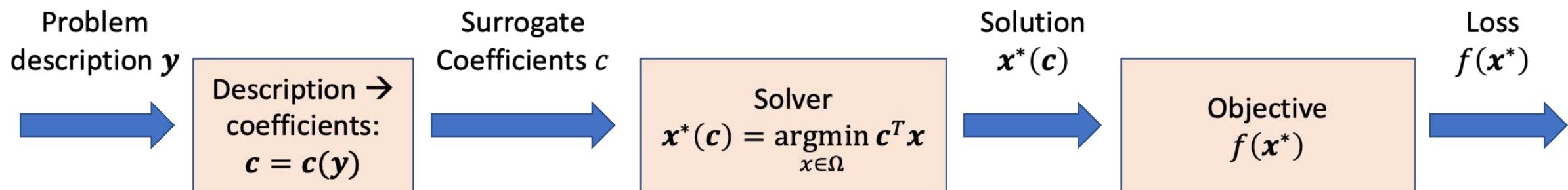


$\mathbf{x}^*(\mathbf{y})$  optimizes  $f(\mathbf{x}; \mathbf{y})$  as much as possible

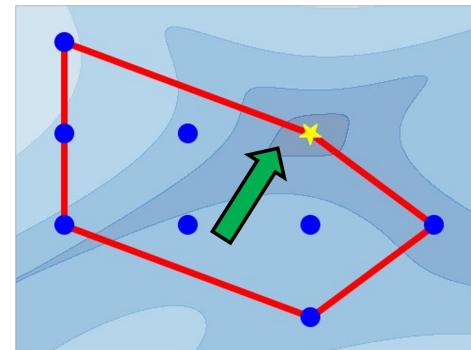
# SurCo: Surrogate combinatorial opt



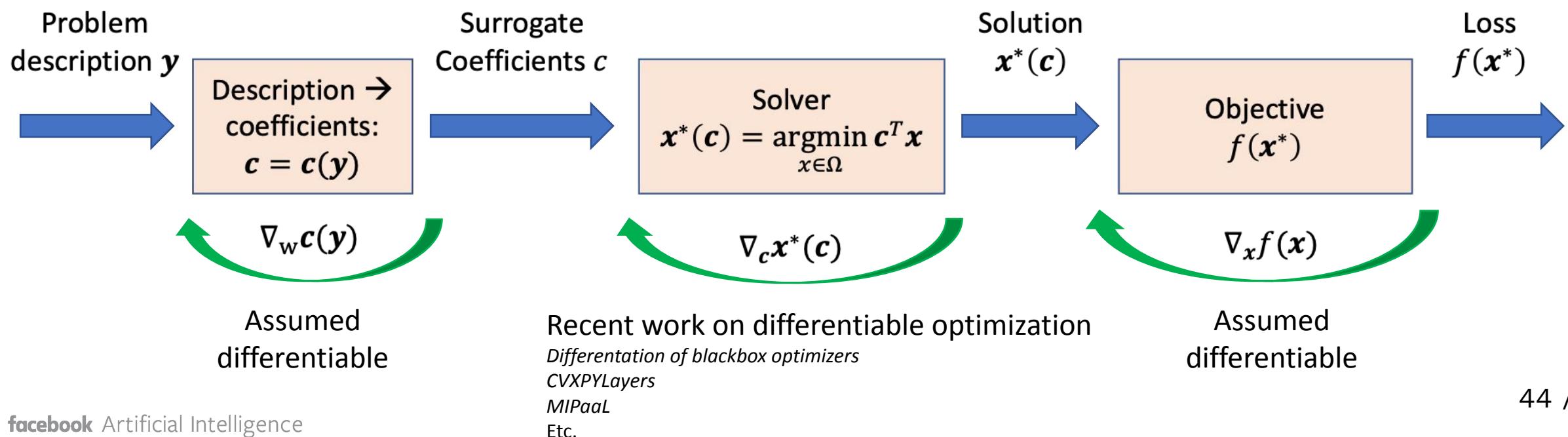
- Use surrogate MILP to solve original problem
- Find linear coefficients  $\mathbf{c}$  such that  $\underset{x \in \Omega}{\operatorname{argmin}} f(x) = \underset{x \in \Omega}{\operatorname{argmin}} \mathbf{c}^T \mathbf{x}$



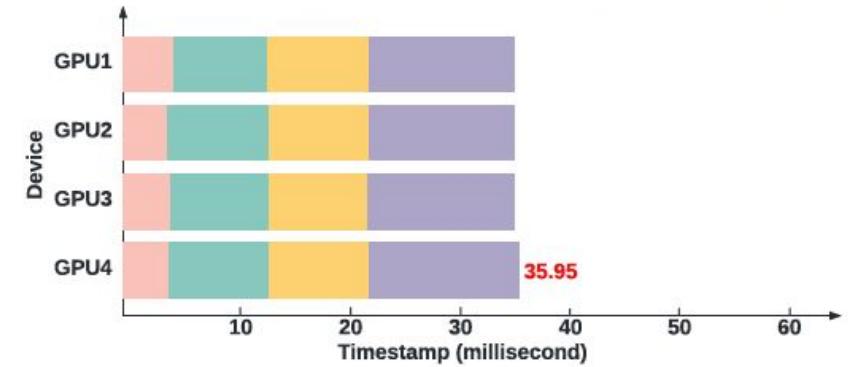
# Gradient-based Optimization



- Use surrogate MILP to solve original problem
- Find linear coefficients  $\mathbf{c}$  such that  $\underset{x \in \Omega}{\operatorname{argmin}} f(x) = \underset{x \in \Omega}{\operatorname{argmin}} \mathbf{c}^T \mathbf{x}$



# Embedding Table Sharding

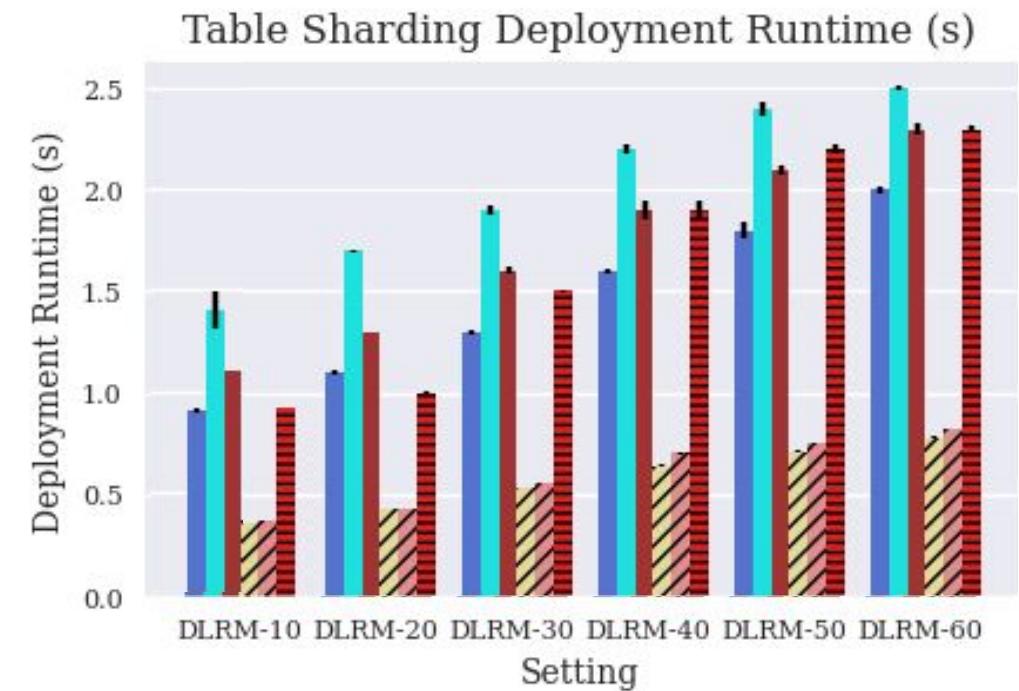
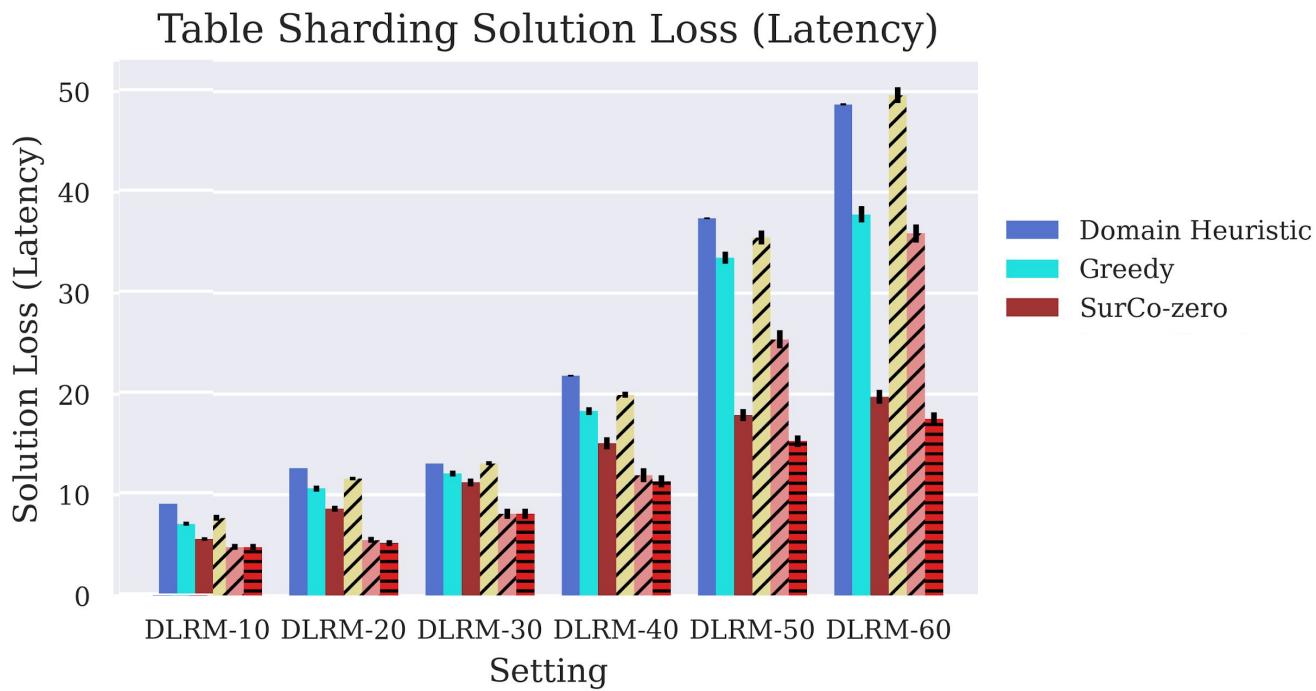


- Public Deep Learning Recommendation Model (DLRM dataset) placing between 10 to 60 tables on 4 GPUs
- Baseline: Greedy
- SoTA: RL approach Dreamshard<sup>1</sup>
- SurCo: Surrogate NN model learned via CVXPYLayers (differentiable LP Solver)

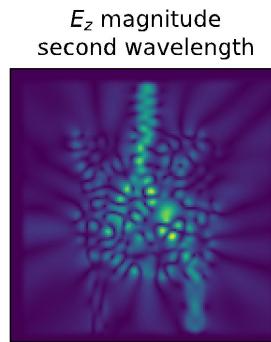
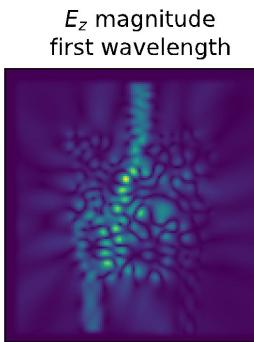
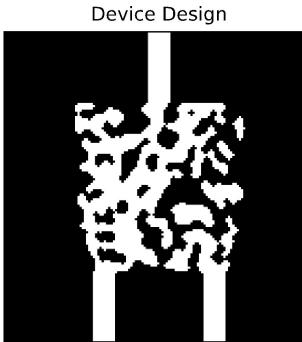
<sup>1</sup>Zha et al. NeurIPS 2022

Dataset: [https://github.com/facebookresearch/dlrm\\_datasets](https://github.com/facebookresearch/dlrm_datasets)

# Results – Table Sharding



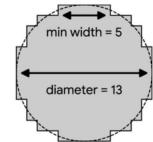
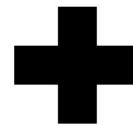
# Inverse Photonic Design



- Design physically-viable devices that take light waves and routes different wavelengths to correct locations

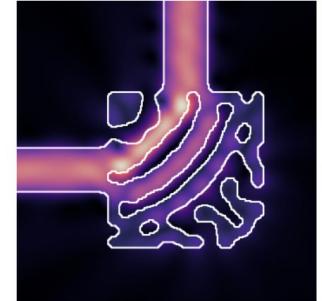
$$\mathcal{L}(S) = \left( \left\| \text{softplus} \left( g \frac{|S|^2 - |S_{\text{cutoff}}|^2}{\min(w_{\text{valid}})} \right) \right\|_2 \right)^2$$

- Device design misspecification loss  $f(\mathbf{x})$  computed by differentiable electromagnetic simulator
- Feasible solution: the design must be the union of brush patterns
  - $\mathbf{x} = \text{binary\_opening}(\mathbf{x}, \text{brush})$
  - $\mathbf{x} = \sim \text{binary\_opening}(\sim \mathbf{x}, \text{brush})$

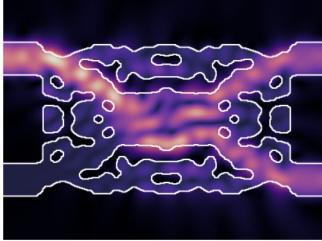


# Inverse Photonic Design

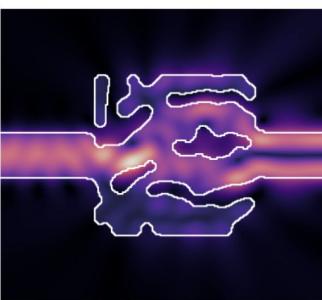
- Dataset: Ceviche Challenges<sup>1</sup>
- Most baselines don't work here due to combinatorial constraints
- SoTA: Brush-based algorithm <sup>1</sup>
- SurCo: Surrogate learned via blackbox differentiation<sup>2</sup> of brush solver



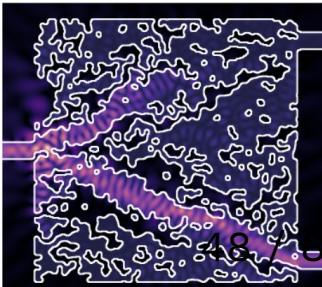
Waveguide  
bend



Beam splitter



Mode converter

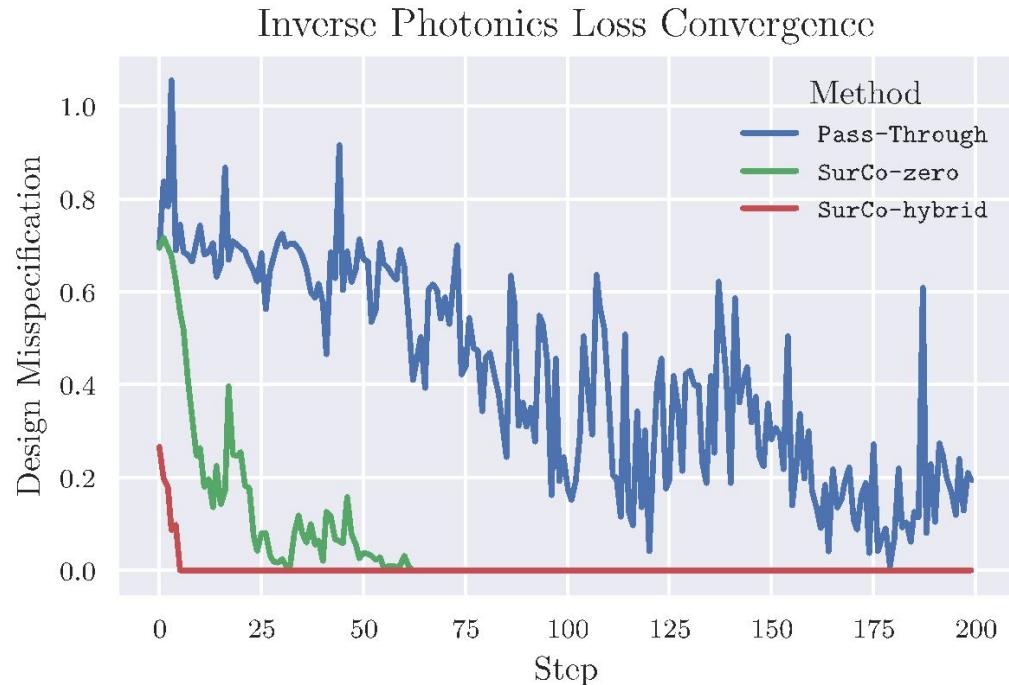


<sup>1</sup>Schubert et al. ACS Photonics 2022

<sup>2</sup>Vlastelica et al. ICLR 2019

Dataset: <https://github.com/google/ceviche-challenges>

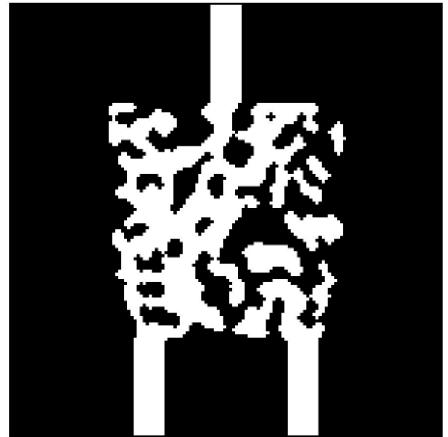
# Inverse photonics Convergence comparison + Solution example



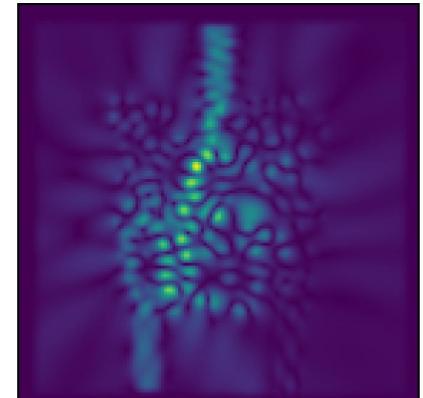
Takeaways:

- SurCo-Zero finds loss-0 solutions quickly
- SurCo-Hybrid uses offline training data to get a head start

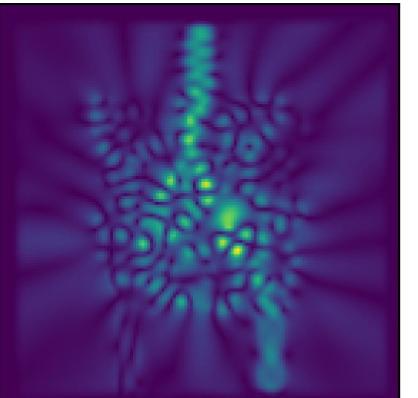
Device Design



$E_z$  magnitude  
first wavelength



$E_z$  magnitude  
second wavelength



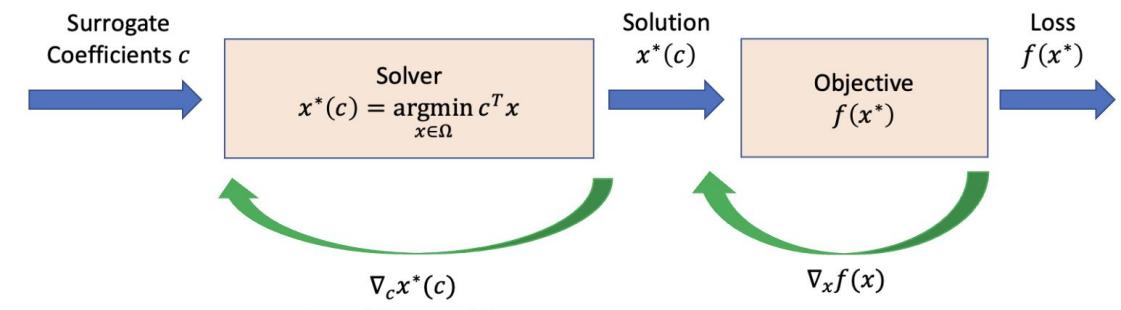
Wavelength division  
multiplexer

# Limitation of SurCo

Recall **SurCo**: Update linear coefficients  $\mathbf{c}$  such that  $x^*(\mathbf{c})$  improves objective  $f(x^*(\mathbf{c}))$

$$\min_{\theta} \mathcal{L}(Y, Z) := \sum_{i=1}^N f(\mathbf{g}_{\theta}(\mathbf{y}_i); \mathbf{z}_i)$$

$$\mathbf{g}_{\theta}(\mathbf{y}) = \arg \min_{\mathbf{x} \in \Omega} \mathbf{x}^\top \mathbf{c}_{\theta}(\mathbf{y})$$



- Requires  $\nabla_x f(x) \rightarrow$  Does not apply to nondifferentiable functions
- Requires  $\nabla_c g_{\theta}(c) \rightarrow$  Solver is backpropagatable

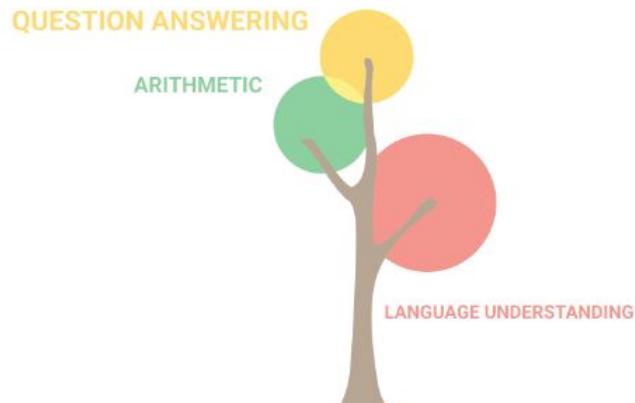
# Option Three: Does Deep Model Actually Converge to Anything Symbolic?



Emerging Symbolic  
Structure

Deep Models

# Debate: Is LLM doing retrieval or true reasoning?



8 billion parameters

LLM shows emergent behaviors!!

# Debate: Is LLM doing retrieval or true reasoning?



Yann LeCun ✅

@ylecun

Do LLMs perform reasoning or approximate retrieval?

There is a continuum between the two, and Auto-Regressive LLMs are largely on the retrieval side.

...



Subbarao Kambhampati (కంభంపాటి సుబ్రావు) ✅

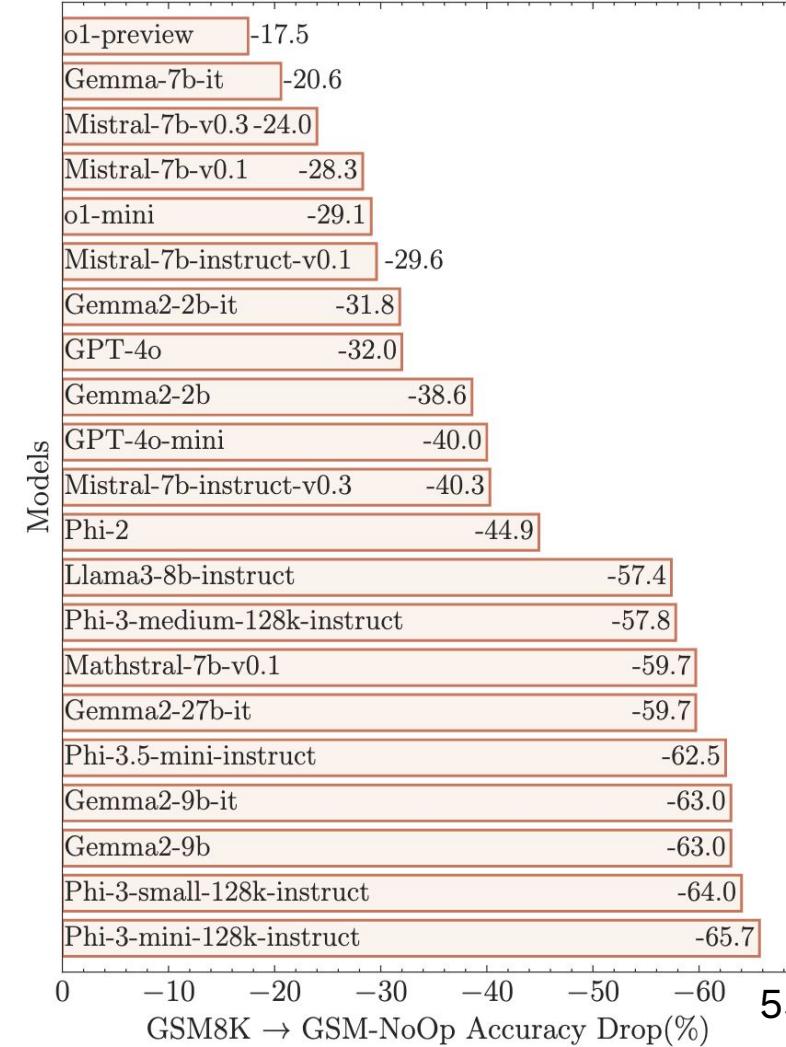
@rao2z

...

Emergent Abilities (noun): The preferred euphemism for what your LLM does, when saying "approximate retrieval" sounds too unsexy.

#AI Aphorisms

LLM is just doing retrievals!!



# Concrete Example: Modular Addition

$$a + b = c \bmod d$$

Does neural network have an *implicit table* to do retrieval?

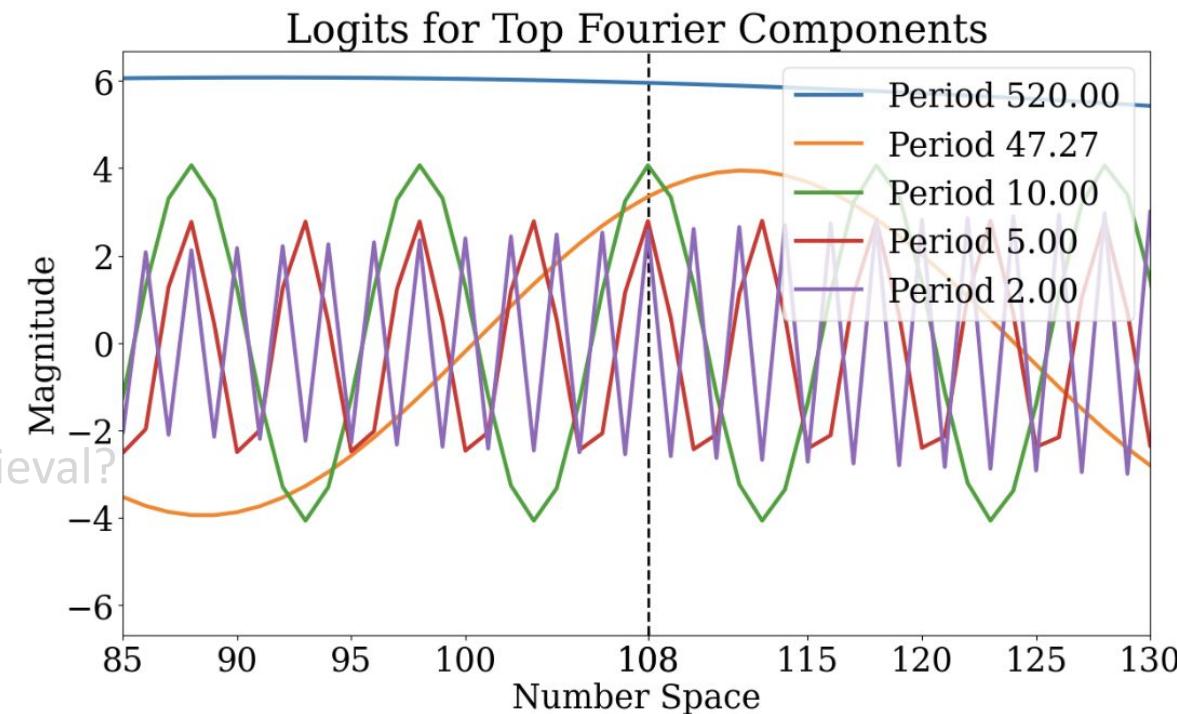
# Concrete Example: Modular Addition

$$a + b = c \bmod d$$

Does neural network have an *implicit table* to do retrieval?

Learned representation = Fourier basis 

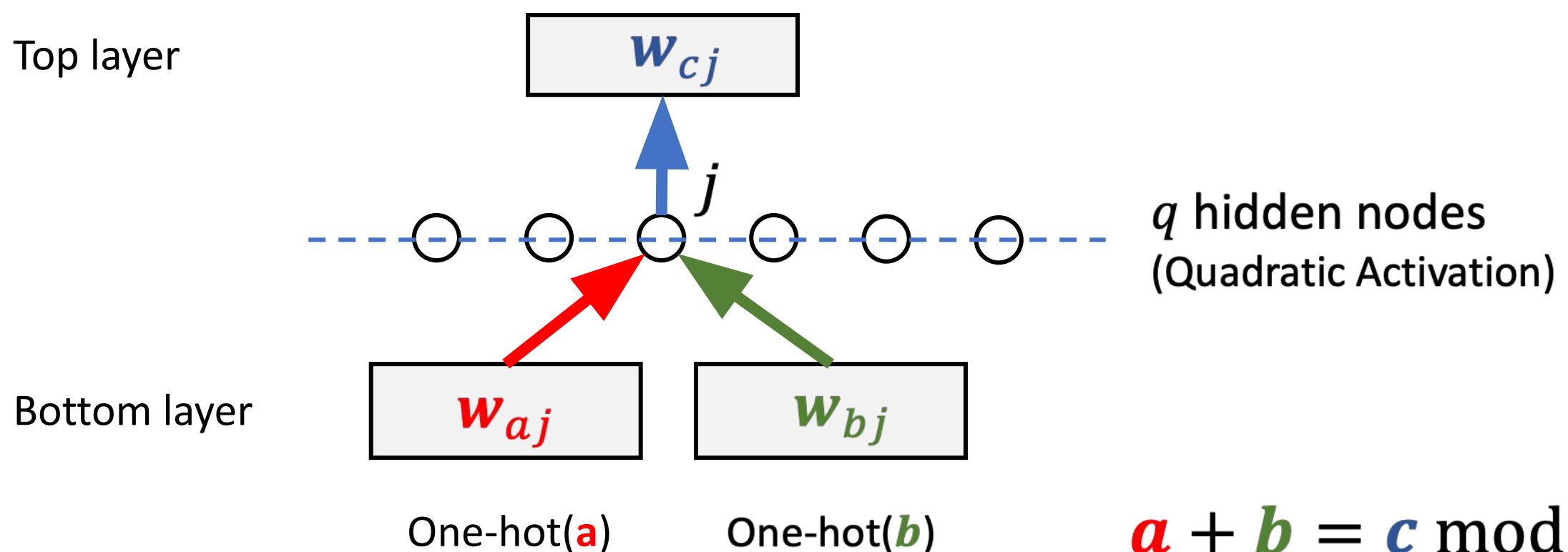
Why? 



(a) Final logits for top Fourier components

# Problem Setup

**MSE Loss:**  $\text{Min} \ \| \text{Output} - \text{one-hot}(\mathbf{c}) \|_2$



$$\mathbf{a} + \mathbf{b} = \mathbf{c} \bmod d$$

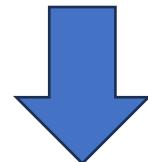
# (Scaled) Fourier Transform

$k$ : frequency

$$z_{akj} = \sum_{m=0}^{d-1} w_{amj} e^{\text{i}mk/d}$$

$\{W_a, W_b, W_c\}$  are real

$$z_{bkj} = \sum_{m=0}^{d-1} w_{bmj} e^{\text{i}mk/d}$$



*Hermitian condition holds*

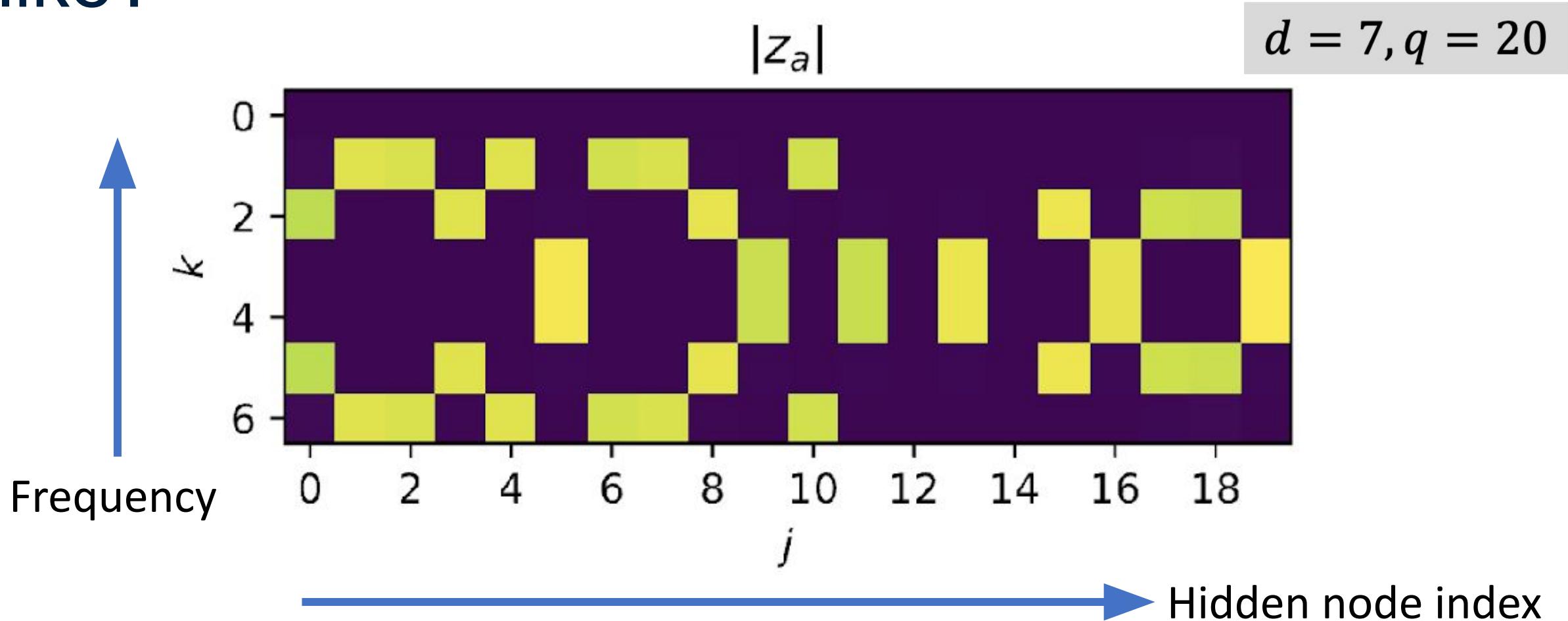
$$z_{ckj} = \sum_{m=0}^{d-1} w_{cmj} e^{\text{i}mk/d}$$

$$z_{akj} = \overline{z_{a,-k,j}}$$

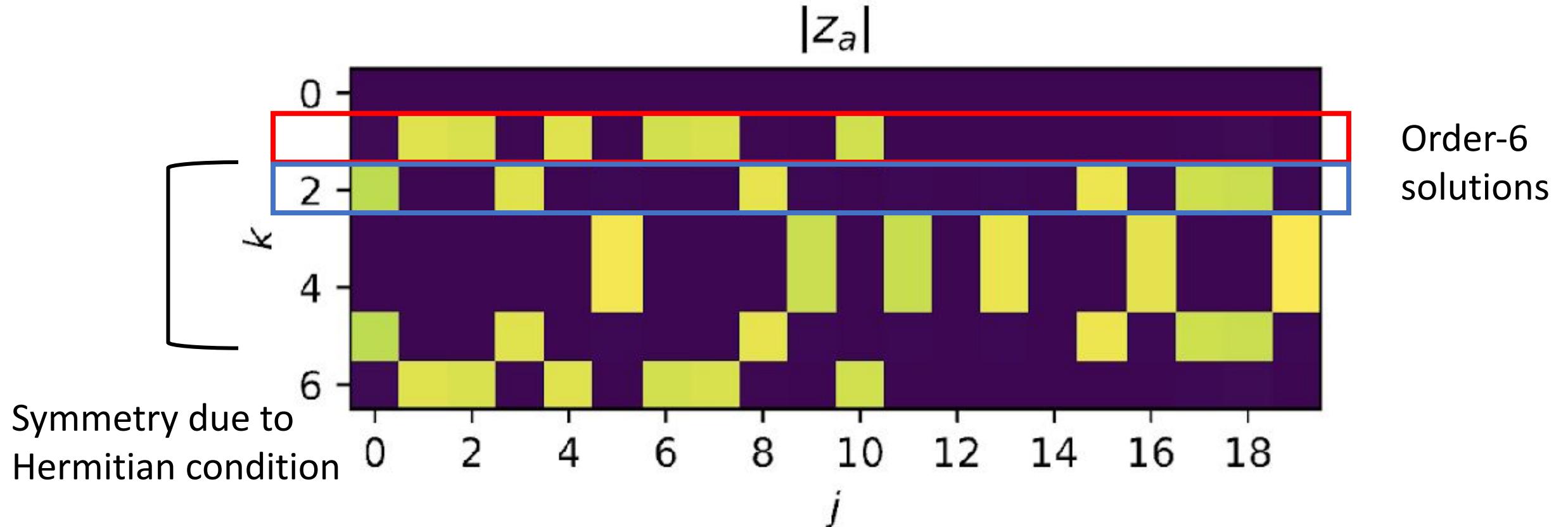
$$z_{bkj} = \overline{z_{b,-k,j}}$$

$$z_{ckj} = \overline{z_{c,-k,j}}$$

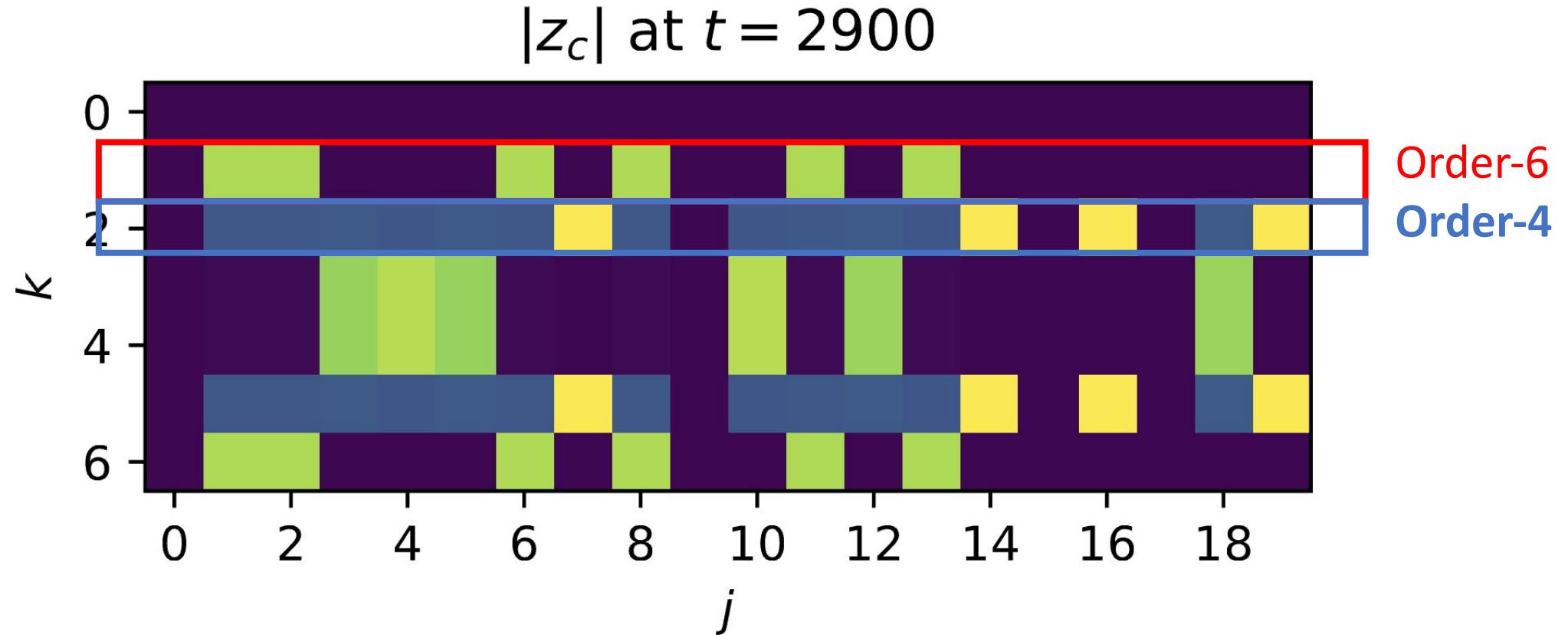
# What a Gradient Descent Solution look like?



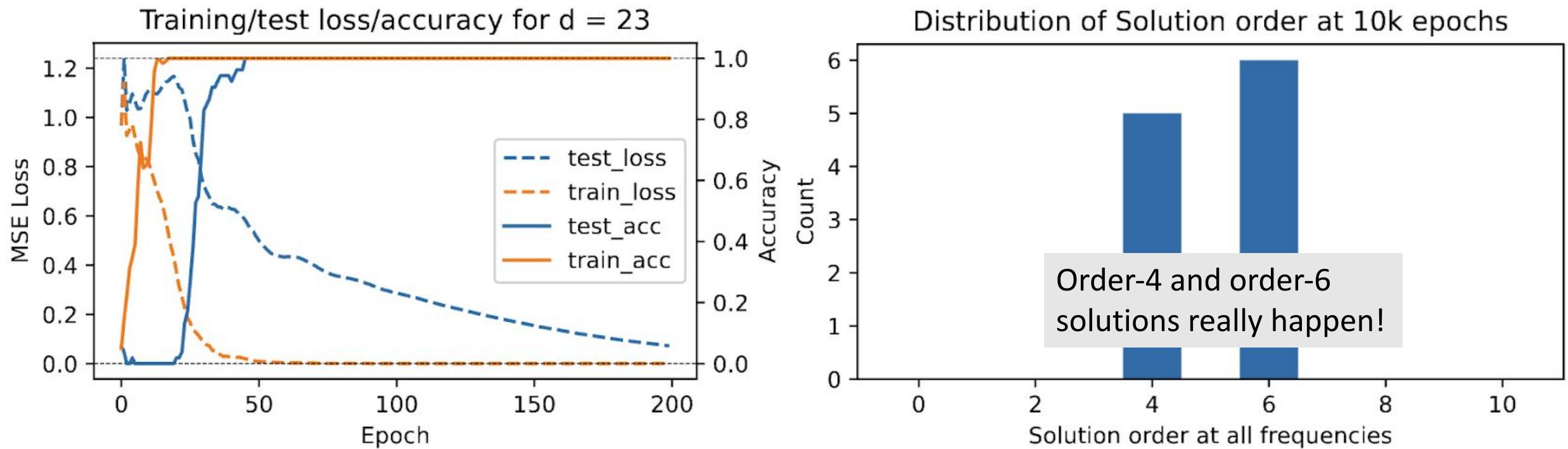
# What a Gradient Descent Solution look like?



# What a Gradient Descent Solution look like?

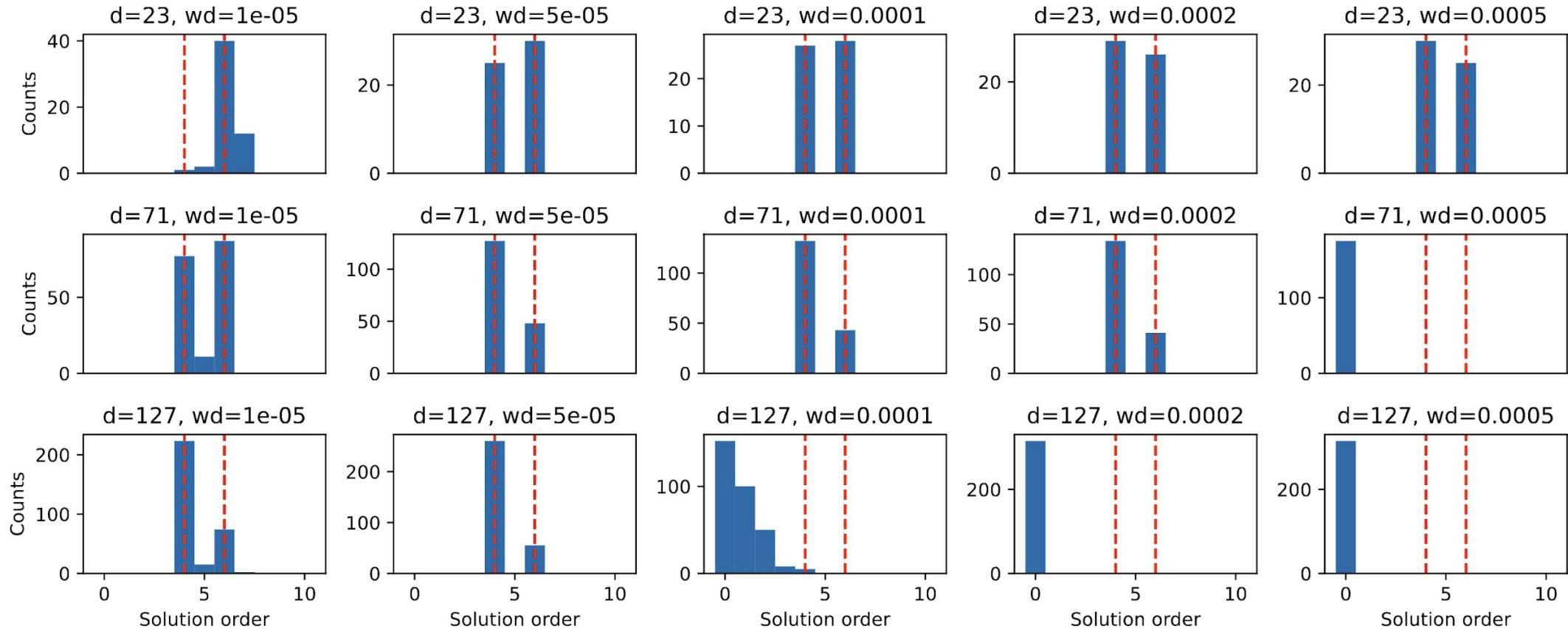


# More Statistics on Gradient Descent Solutions



# Effect of Weight Decay

Stronger weight decay →



# Why?



# Structure of Loss Functions

$$\text{MSE loss } \ell(\mathbf{z}) = d^{-1} \sum_{k \neq 0} \ell_k(\mathbf{z}) + 1 - 1/d$$

$$\ell_k(\mathbf{z}) = -2\mathbf{r}_{kkk} + \sum_{k_1 k_2} |\mathbf{r}_{k_1 k_2 k}|^2 + \frac{1}{4} \left| \sum_{p \in \{a,b\}} \sum_{k'} \mathbf{r}_{p,k',-k',k} \right|^2 + \frac{1}{4} \sum_{m \neq 0} \sum_{p \in \{a,b\}} \left| \sum_{k'} \mathbf{r}_{p,k',m-k',k} \right|^2$$

Term  $\mathbf{r}_{k_1 k_2 k}(\mathbf{z}) := \sum_j z_{ak_1 j} z_{bk_2 j} z_{ckj}$  and  $\mathbf{r}_{pk_1 k_2 k}(\mathbf{z}) := \sum_j z_{pk_1 j} z_{pk_2 j} z_{ckj}$

# Structure of Loss Functions

MSE loss  $\ell(\mathbf{z}) = d^{-1} \sum_{k \neq 0} \ell_k(\mathbf{z}) + 1 - 1/d$

$$\ell_k(\mathbf{z}) = -2\mathbf{r}_{kkk} + \sum_{k_1 k_2} |\mathbf{r}_{k_1 k_2 k}|^2 + \frac{1}{4} \left| \sum_{p \in \{a,b\}} \sum_{k'} \mathbf{r}_{p,k',-k',k} \right|^2 + \frac{1}{4} \sum_{m \neq 0} \sum_{p \in \{a,b\}} \left| \sum_{k'} \mathbf{r}_{p,k',m-k',k} \right|^2$$

Term  $\mathbf{r}_{k_1 k_2 k}(\mathbf{z}) := \sum_j z_{ak_1 j} z_{bk_2 j} z_{ckj}$  and  $\mathbf{r}_{pk_1 k_2 k}(\mathbf{z}) := \sum_j z_{pk_1 j} z_{pk_2 j} z_{ckj}$

Sufficient conditions of Global Optimizers:

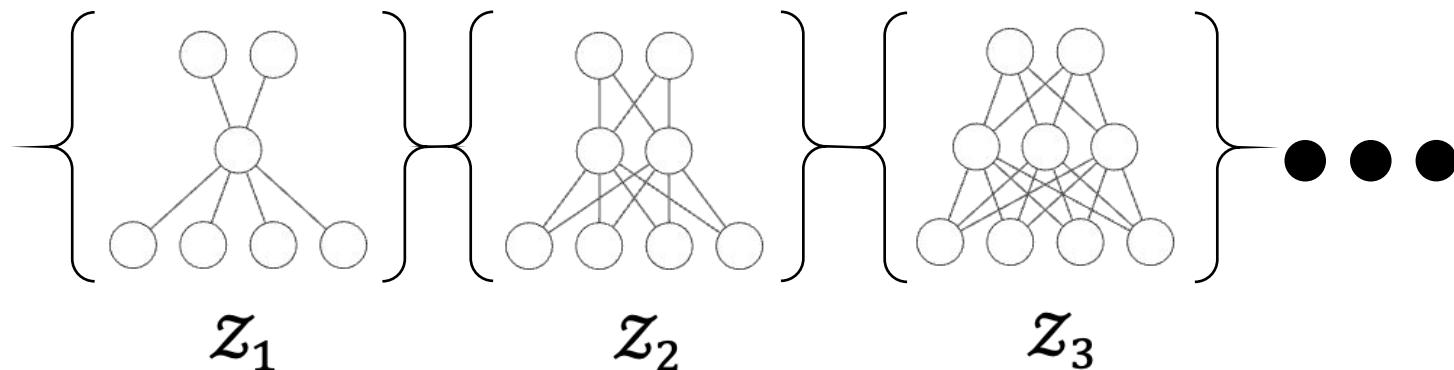

# How to Optimize?

The objective is highly nonlinear !!  
However, nice *algebraic structures* exist!

# How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!

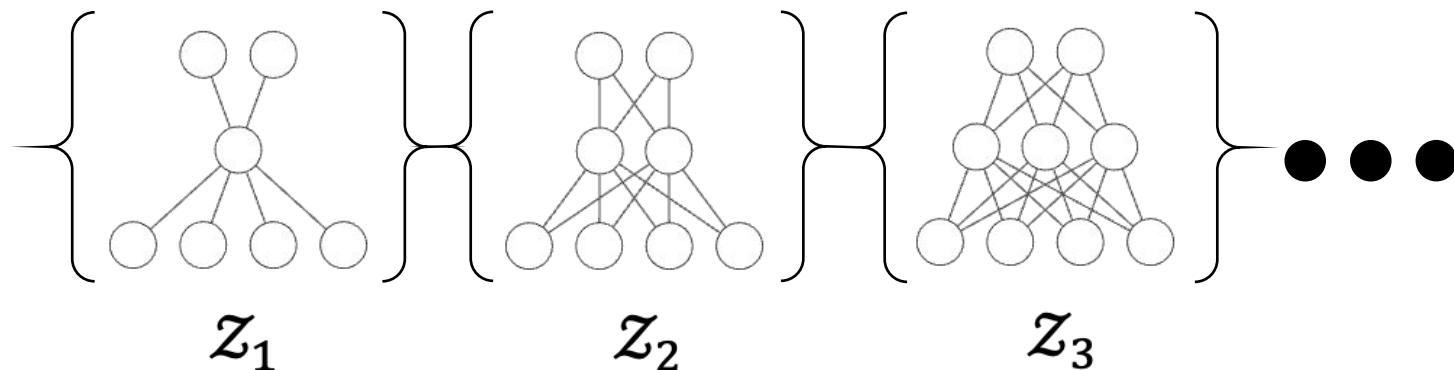


$\mathcal{Z} = \bigcup_{q \geq 0} \mathcal{Z}_q$ : All 2-layer networks with different number of hidden nodes

# How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!



$\mathcal{Z} = \bigcup_{q \geq 0} \mathcal{Z}_q$ : All 2-layer networks with different number of hidden nodes

Ring addition  $+$ : Concatenate hidden nodes

Ring multiplication  $*$ : Kronecker production along the hidden dimensions

$\langle \mathcal{Z}, +, *\rangle$  is a *semi-ring*

# Ring Homomorphism

A function  $r(\mathbf{z}): \mathcal{Z} \mapsto \mathbb{C}$  is a *ring homomorphism*, if

- $r(\mathbf{1}) = 1$
- $r(\mathbf{z}_1 + \mathbf{z}_2) = r(\mathbf{z}_1) + r(\mathbf{z}_2)$
- $r(\mathbf{z}_1 * \mathbf{z}_2) = r(\mathbf{z}_1)r(\mathbf{z}_2)$

# Ring Homomorphism

A function  $r(z): \mathcal{Z} \mapsto \mathbb{C}$  is a *ring homomorphism*, if

- $r(\mathbf{1}) = 1$
- $r(z_1 + z_2) = r(z_1) + r(z_2)$
- $r(z_1 * z_2) = r(z_1)r(z_2)$



$r_{k_1 k_2 k}(z)$  and  $r_{pk_1 k_2 k}(z)$  are **ring homomorphisms!**

# Ring Homomorphism

A function  $r(\mathbf{z}): \mathcal{Z} \mapsto \mathbb{C}$  is a *ring homomorphism*, if

- $r(\mathbf{1}) = 1$
- $r(\mathbf{z}_1 + \mathbf{z}_2) = r(\mathbf{z}_1) + r(\mathbf{z}_2)$
- $r(\mathbf{z}_1 * \mathbf{z}_2) = r(\mathbf{z}_1)r(\mathbf{z}_2)$

🎯  $r_{k_1 k_2 k}(\mathbf{z})$  and  $r_{pk_1 k_2 k}(\mathbf{z})$  are ring homomorphisms!

MSE Loss

$$\ell_k(\mathbf{z}) = -2\mathbf{r}_{\mathbf{k}\mathbf{k}\mathbf{k}} + \sum_{k_1 k_2} |\mathbf{r}_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}}|^2 + \frac{1}{4} \left| \sum_{p \in \{a,b\}} \sum_{k'} \mathbf{r}_{p, k', -k', k} \right|^2 + \frac{1}{4} \sum_{m \neq 0} \sum_{p \in \{a,b\}} \left| \sum_{k'} \mathbf{r}_{p, k', m - k', k} \right|^2$$

# Ring Homomorphism

A function  $r(\mathbf{z}): \mathcal{Z} \mapsto \mathbb{C}$  is a *ring homomorphism*, if

- $r(\mathbf{1}) = 1$
- $r(\mathbf{z}_1 + \mathbf{z}_2) = r(\mathbf{z}_1) + r(\mathbf{z}_2)$
- $r(\mathbf{z}_1 * \mathbf{z}_2) = r(\mathbf{z}_1)r(\mathbf{z}_2)$

🎯  $r_{k_1 k_2 k}(\mathbf{z})$  and  $r_{pk_1 k_2 k}(\mathbf{z})$  are ring homomorphisms!

MSE Loss

$$\ell_k(\mathbf{z}) = -2\mathbf{r}_{\mathbf{k}\mathbf{k}\mathbf{k}} + \sum_{k_1 k_2} |\mathbf{r}_{k_1 k_2 k}|^2 + \frac{1}{4} \left| \sum_{p \in \{a,b\}} \sum_{k'} \mathbf{r}_{p, k', -k', k} \right|^2 + \frac{1}{4} \sum_{m \neq 0} \sum_{p \in \{a,b\}} \left| \sum_{k'} \mathbf{r}_{p, k', m - k', k} \right|^2$$

Partial solution  $\mathbf{z}_1$  satisfies  $\mathbf{r}_{k_1 k_2 k}(\mathbf{z}_1) = 0$

Partial solution  $\mathbf{z}_2$  satisfies  $\mathbf{r}_{pk', -k', k}(\mathbf{z}_2) = 0$

# Ring Homomorphism

A function  $r(\mathbf{z}): \mathcal{Z} \mapsto \mathbb{C}$  is a *ring homomorphism*, if

- $r(\mathbf{1}) = 1$
- $r(\mathbf{z}_1 + \mathbf{z}_2) = r(\mathbf{z}_1) + r(\mathbf{z}_2)$
- $r(\mathbf{z}_1 * \mathbf{z}_2) = r(\mathbf{z}_1)r(\mathbf{z}_2)$

🎯  $r_{k_1 k_2 k}(\mathbf{z})$  and  $r_{pk_1 k_2 k}(\mathbf{z})$  are ring homomorphisms!

MSE Loss

$$\ell_k(\mathbf{z}) = -2\mathbf{r}_{kkk} + \sum_{k_1 k_2} |\mathbf{r}_{k_1 k_2 k}|^2 + \frac{1}{4} \left| \sum_{p \in \{a,b\}} \sum_{k'} \mathbf{r}_{p,k',-k',k} \right|^2 + \frac{1}{4} \sum_{m \neq 0} \sum_{p \in \{a,b\}} \left| \sum_{k'} \mathbf{r}_{p,k',m-k',k} \right|^2$$

Partial solution  $\mathbf{z}_1$  satisfies  $\mathbf{r}_{k_1 k_2 k}(\mathbf{z}_1) = 0$

Partial solution  $\mathbf{z}_2$  satisfies  $\mathbf{r}_{pk',-k',k}(\mathbf{z}_2) = 0$

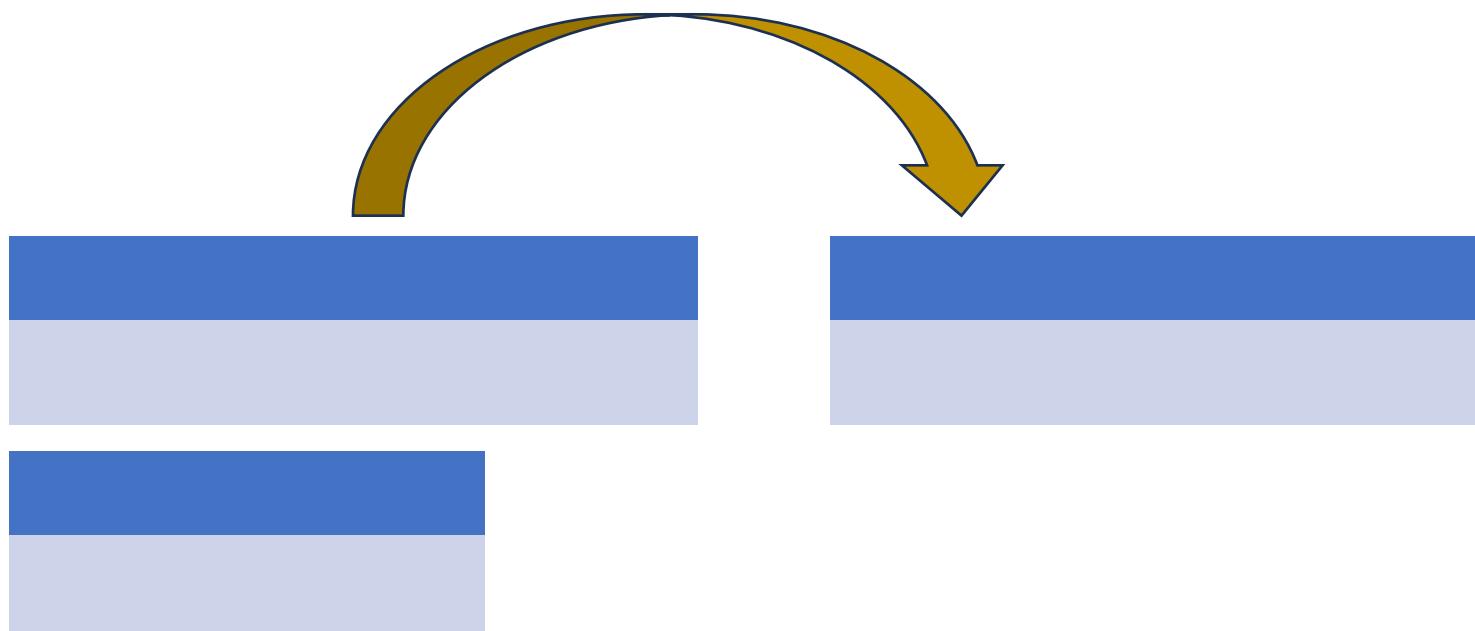
$\left. \begin{array}{l} \text{Partial solution } \mathbf{z}_1 \text{ satisfies } \mathbf{r}_{k_1 k_2 k}(\mathbf{z}_1) = 0 \\ \text{Partial solution } \mathbf{z}_2 \text{ satisfies } \mathbf{r}_{pk',-k',k}(\mathbf{z}_2) = 0 \end{array} \right\} \mathbf{z} = \mathbf{z}_1 * \mathbf{z}_2 \text{ satisfies both } \mathbf{r}_{k_1 k_2 k}(\mathbf{z}) = \mathbf{r}_{pk',-k',k}(\mathbf{z}) = 0$

# Composing Global Optimizers from Partial Ones

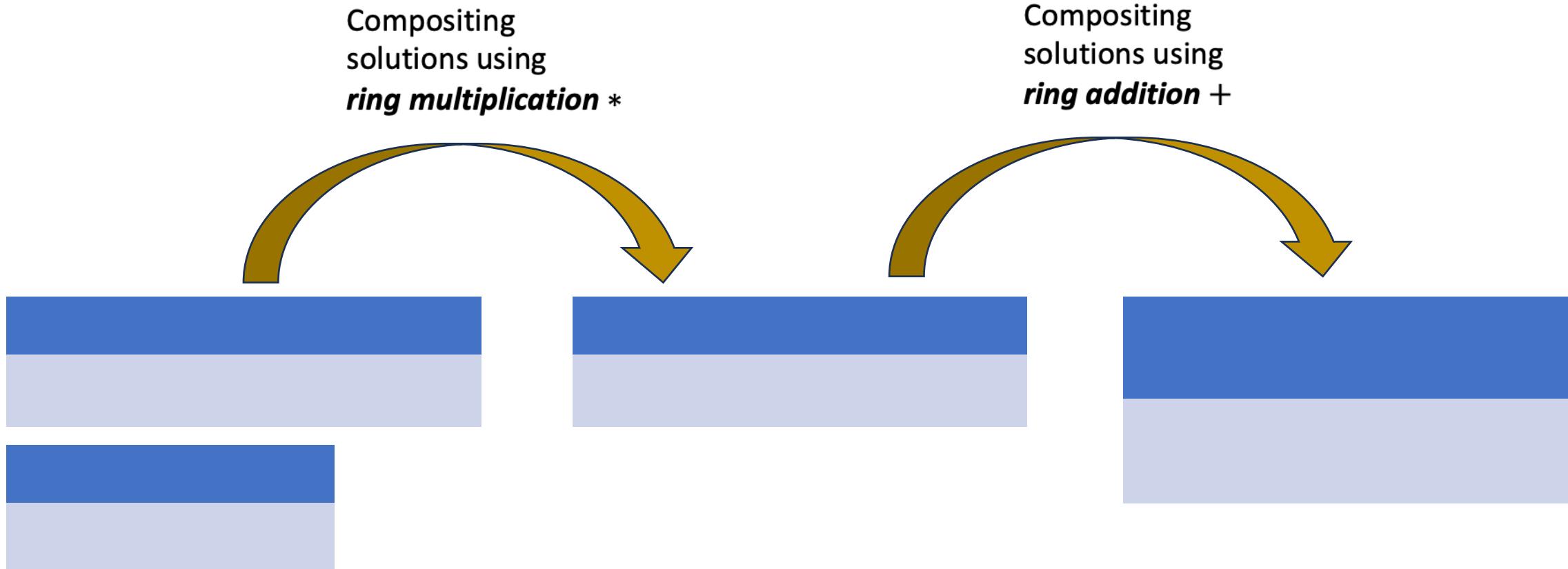


# Composing Global Optimizers from Partial Ones

Compositing  
solutions using  
*ring multiplication* \*



# Composing Global Optimizers from Partial Ones



# Exemplar constructed global optimizers

Order-6  $\mathbf{z}_{F6}$  (2\*3)

$$\mathbf{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \mathbf{z}_{\text{syn}}^{(k)} * \mathbf{z}_{\nu}^{(k)} * \mathbf{y}_k$$

# Exemplar constructed global optimizers

Order-6  $\mathbf{z}_{F6}$  (2\*3)

$$\mathbf{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \mathbf{z}_{\text{syn}}^{(k)} * \mathbf{z}_{\nu}^{(k)} * \mathbf{y}_k$$

Order-4  $\mathbf{z}_{F4/6}$  (2\*2)  
(mixed with order-6)

$$\mathbf{z}_{F4/6} = \frac{1}{\sqrt[3]{6}} \hat{\mathbf{z}}_{F6}^{(k_0)} + \frac{1}{\sqrt[3]{4}} \sum_{k=1, k \neq k_0}^{(d-1)/2} \mathbf{z}_{F4}^{(k)}$$

# Exemplar constructed global optimizers

Order-6  $\mathbf{z}_{F6}$  ( $2^*3$ )

$$\mathbf{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \mathbf{z}_{\text{syn}}^{(k)} * \mathbf{z}_{\nu}^{(k)} * \mathbf{y}_k$$

Order-4  $\mathbf{z}_{F4/6}$  ( $2^*2$ )  
(mixed with order-6)

$$\mathbf{z}_{F4/6} = \frac{1}{\sqrt[3]{6}} \hat{\mathbf{z}}_{F6}^{(k_0)} + \frac{1}{\sqrt[3]{4}} \sum_{k=1, k \neq k_0}^{(d-1)/2} \mathbf{z}_{F4}^{(k)}$$

Perfect memorization  
(order-d per frequency)

$$\mathbf{z}_a = \sum_{j=0}^{d-1} \mathbf{u}_a^j, \quad \mathbf{z}_b = \sum_{j=0}^{d-1} \mathbf{u}_b^j$$

$$\mathbf{z}_M = d^{-2/3} \mathbf{z}_a * \mathbf{z}_b$$

# Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones				others
	order-4	order-6	order-6	order-4	order-6	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn}, \alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$		
23	0.0±0.0	0.00±0.00	5.71±5.71	0.05±0.01	4.80±0.96	47.07±1.88	11.31±1.76	39.80±2.11	1.82±1.82	
71	0.0±0.0	0.00±0.00	0.00±0.00	0.03±0.00	5.02±0.25	72.57±0.70	4.00±1.14	21.14±2.14	2.29±1.07	
127	0.0±0.0	1.50±0.92	0.00±0.00	0.26±0.14	0.93±0.18	82.96±0.39	2.25±0.64	14.13±0.87	0.66±0.66	

$$q = 512, wd = 5 \cdot 10^{-5}$$

# Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable		error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones				others
	order-4	order-6	order-4	order-6	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn}, \alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$			
23	$0.0 \pm 0.0$	$0.00 \pm 0.00$	$5.71 \pm 5.71$	$0.05 \pm 0.01$	$4.80 \pm 0.96$	$47.07 \pm 1.88$	$11.31 \pm 1.76$	$39.80 \pm 2.11$	$1.82 \pm 1.82$	
71	$0.0 \pm 0.0$	$0.00 \pm 0.00$	$0.00 \pm 0.00$	$0.03 \pm 0.00$	$5.02 \pm 0.25$	$72.57 \pm 0.70$	$4.00 \pm 1.14$	$21.14 \pm 2.14$	$2.29 \pm 1.07$	
127	$0.0 \pm 0.0$	$1.50 \pm 0.92$	$0.00 \pm 0.00$	$0.26 \pm 0.14$	$0.93 \pm 0.18$	$82.96 \pm 0.39$	$2.25 \pm 0.64$	$14.13 \pm 0.87$	$0.66 \pm 0.66$	

100% of the per-freq  
solutions are order-4/6

# Gradient Descent solutions matches with construction

d	%not order-4/6			%non-factorable		error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones			
	order-4	order-6	order-4	order-4	order-6	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn}, \alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$	others		
23	0.0±0.0	0.00±0.00	5.71±5.71	0.05±0.01	4.80±0.96	47.07±1.88	11.31±1.76	39.80±2.11	1.82±1.82		
71	0.0±0.0	0.00±0.00	0.00±0.00	0.03±0.00	5.02±0.25	72.57±0.70	4.00±1.14	21.14±2.14	2.29±1.07		
127	0.0±0.0	1.50±0.92	0.00±0.00	0.26±0.14	0.93±0.18	82.96±0.39	2.25±0.64	14.13±0.87	0.66±0.66		

95% of the solutions are  
factorizable into “2\*3” or “2\*2”

# Gradient Descent solutions matches with construction

d	%not order-4/6			%non-factorable order-4			error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones			
	order-4	order-6	order-4	order-4	order-6	order-4	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn}, \alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$	others		
23	0.0±0.0	0.00±0.00	5.71±5.71	0.05±0.01	4.80±0.96	47.07±1.88	11.31±1.76	39.80±2.11	1.82±1.82			
71	0.0±0.0	0.00±0.00	0.00±0.00	0.03±0.00	5.02±0.25	72.57±0.70	4.00±1.14	21.14±2.14	2.29±1.07			
127	0.0±0.0	1.50±0.92	0.00±0.00	0.26±0.14	0.93±0.18	82.96±0.39	2.25±0.64	14.13±0.87	0.66±0.66			

Factorization error is very small

# Gradient Descent solutions matches with construction

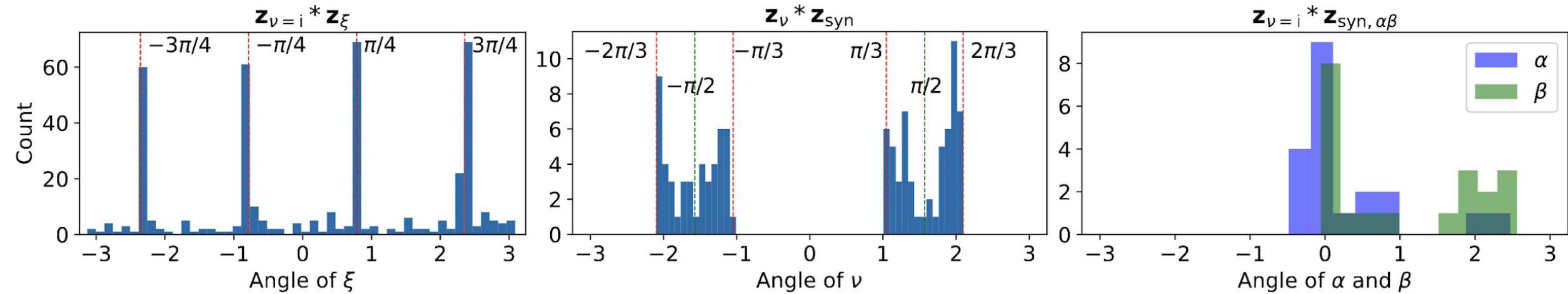
d	%not order-4/6			%non-factorable order-4			error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones			
	order-4	order-6	order-4	order-4	order-6	order-4	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn}, \alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$	others		
23	0.0±0.0	0.00±0.00	5.71±5.71	0.05±0.01	4.80±0.96	47.07±1.88	11.31±1.76	39.80±2.11	1.82±1.82			
71	0.0±0.0	0.00±0.00	0.00±0.00	0.03±0.00	5.02±0.25	72.57±0.70	4.00±1.14	21.14±2.14	2.29±1.07			
127	0.0±0.0	1.50±0.92	0.00±0.00	0.26±0.14	0.93±0.18	82.96±0.39	2.25±0.64	14.13±0.87	0.66±0.66			

98% of the solutions can be factorizable into the constructed forms

# Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ( $\times 10^{-2}$ )		solution distribution (%) in factorable ones			
	order-4	order-6	order-6	order-4	order-6	$z_{\nu=i}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=i}^{(k)} * z_{\text{syn},\alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\text{syn}}^{(k)}$	others
23	0.0±0.0	0.00±0.00	5.71±5.71	0.05±0.01	4.80±0.96	47.07±1.88	11.31±1.76	39.80±2.11	1.82±1.82
	5	72.57±0.70	4.00±1.14	21.14±2.14	2.29±1.07				
	8	82.96±0.39	2.25±0.64	14.13±0.87	0.66±0.66				

Distribution of the parameters in the solutions

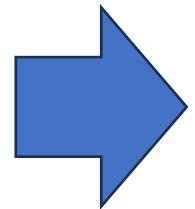


# Possible Implications

Do neural networks end up learning more efficient symbolic representations that we don't know?

Does gradient descent lead to a solution that can be reached by advanced algebraic operations?

Will gradient descent become obsolete, eventually?



# Thanks!