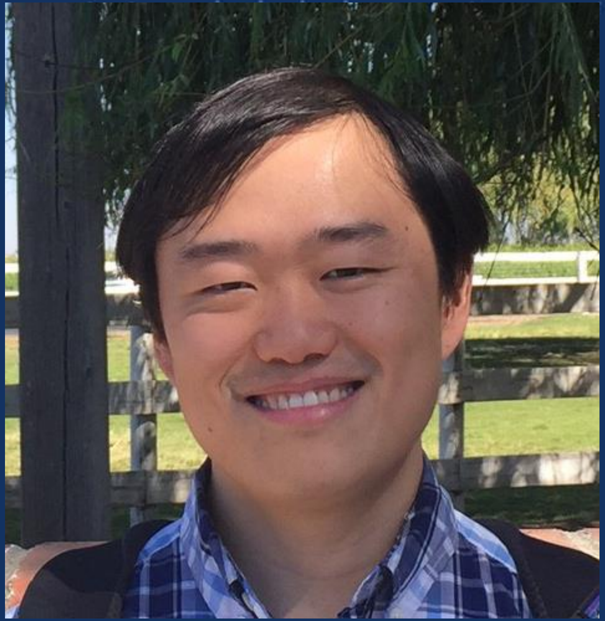
Towards a unified framework of Neural and Symbolic Decision Making

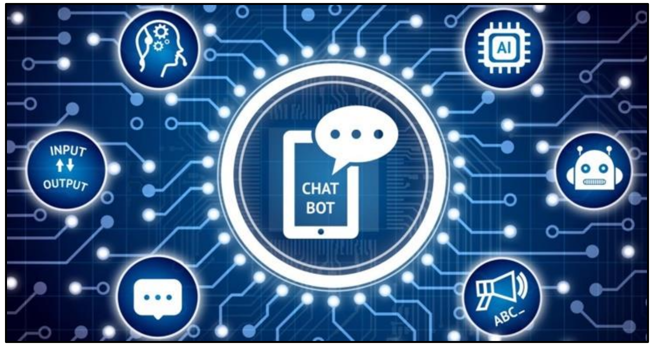
Yuandong Tian
Research Scientist Director

Meta AI (FAIR)



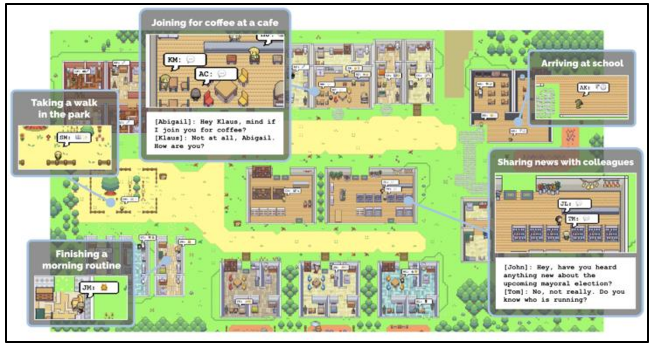
facebook Artificial Intelligence

Large Language Models (LLMs)



Conversational AI

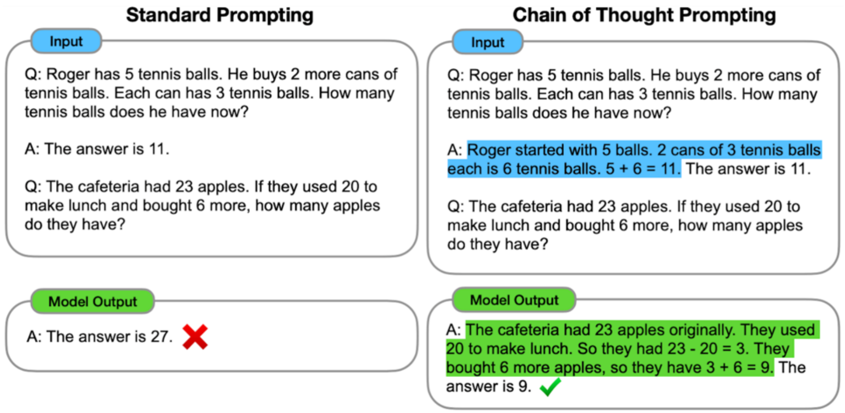
Content Generation


Joining for coffee at a cafe
KM:
AC:
Arriving at school
AK:
Taking a walk in the park
SM:
Sharing news with colleagues
JL:
TM:
Finishing a morning routine
JM:

[Abigail]: Hey Klaus, mind if I join you for coffee?
[Klaus]: Not at all, Abigail. How are you?

[John]: Hey, have you heard anything new about the upcoming mayoral election?
[Tom]: No, not really. Do you know who is running?

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?

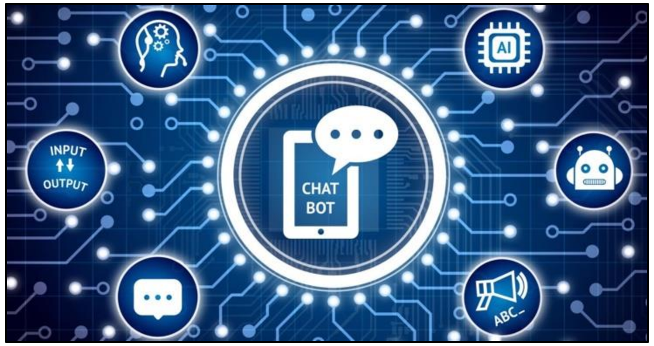
Model Output

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

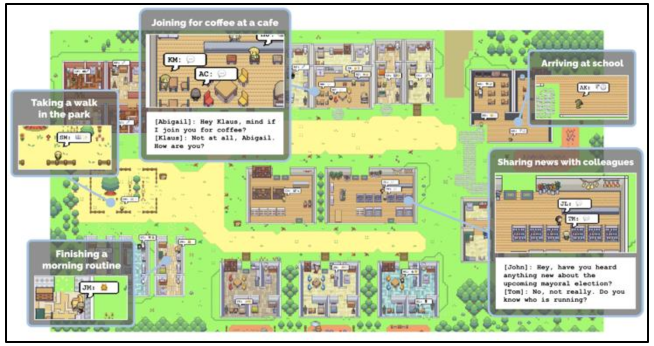
Planning

Large Language Models (LLMs)



Conversational AI

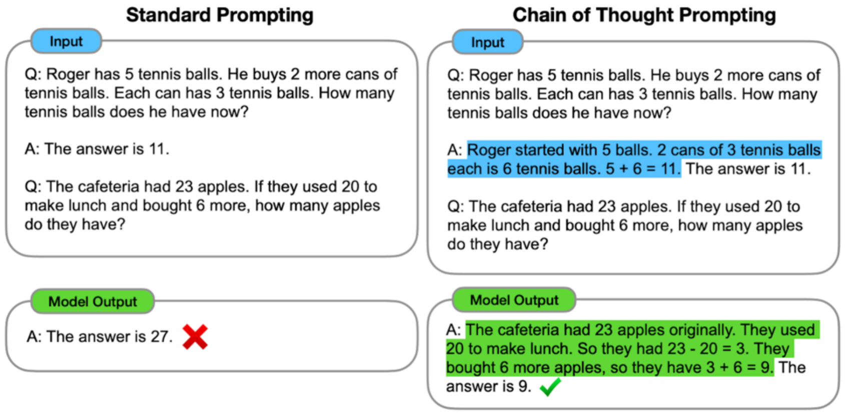
Content Generation


Joining for coffee at a cafe
Arriving at school
Taking a walk in the park
Sharing news with colleagues
Finishing a morning routine

[Abigail]: Hey Klaus, mind if I join you for coffee?
[Klaus]: Not at all, Abigail. How are you?

[John]: Hey, have you heard anything new about the upcoming mayoral election?
[Tom]: No, not really. Do you know who is running?

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?

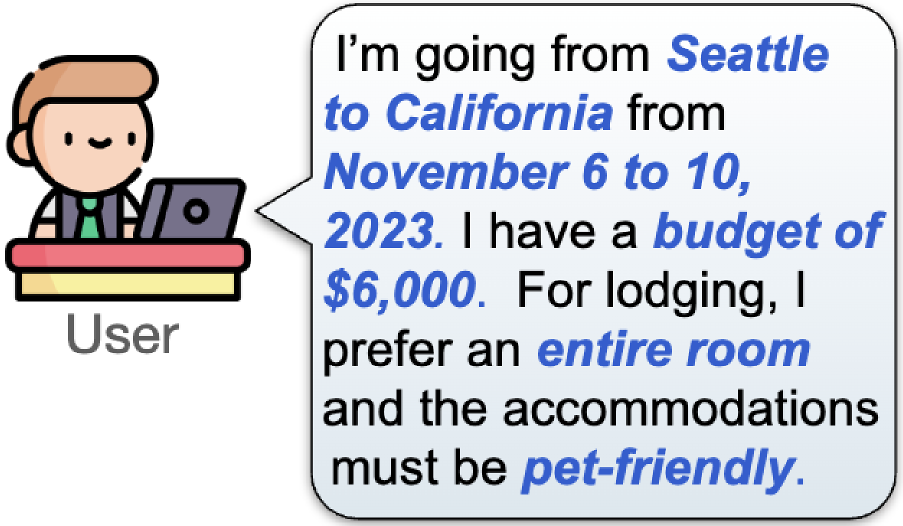
Model Output

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?



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

[J. Xie et al, *TravelPlanner: A Benchmark for Real-World Planning with Language Agents*, ICML’24 (**Spotlight**)]

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 …

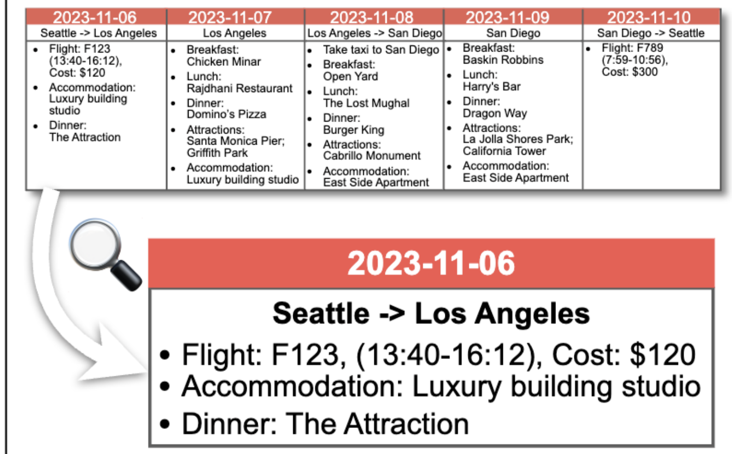
User Needs (Hard Constraints)

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	Los Angeles	Los Angeles -> San Diego	San Diego	San Diego -> Seattle
• Flight: F123 (13:40-16:12), Cost: $120 • Accommodation: Luxury building studio • Dinner: The Attraction	• Breakfast: Chicken Minar • Lunch: Rajdhani Restaurant • Dinner: Domino's Pizza • Attractions: Santa Monica Pier; Griffith Park • Accommodation: Luxury building studio	• Take taxi to San Diego • Breakfast: Open Yard • Lunch: The Lost Mughal • Dinner: Burger King • Attractions: Cabrillo Monument • Accommodation: East Side Apartment	• Breakfast: Baskin Robbins • Lunch: Harry's Bar • Dinner: Dragon Way • Attractions: La Jolla Shores Park; California Tower • Accommodation: East Side Apartment	• Flight: F789 (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

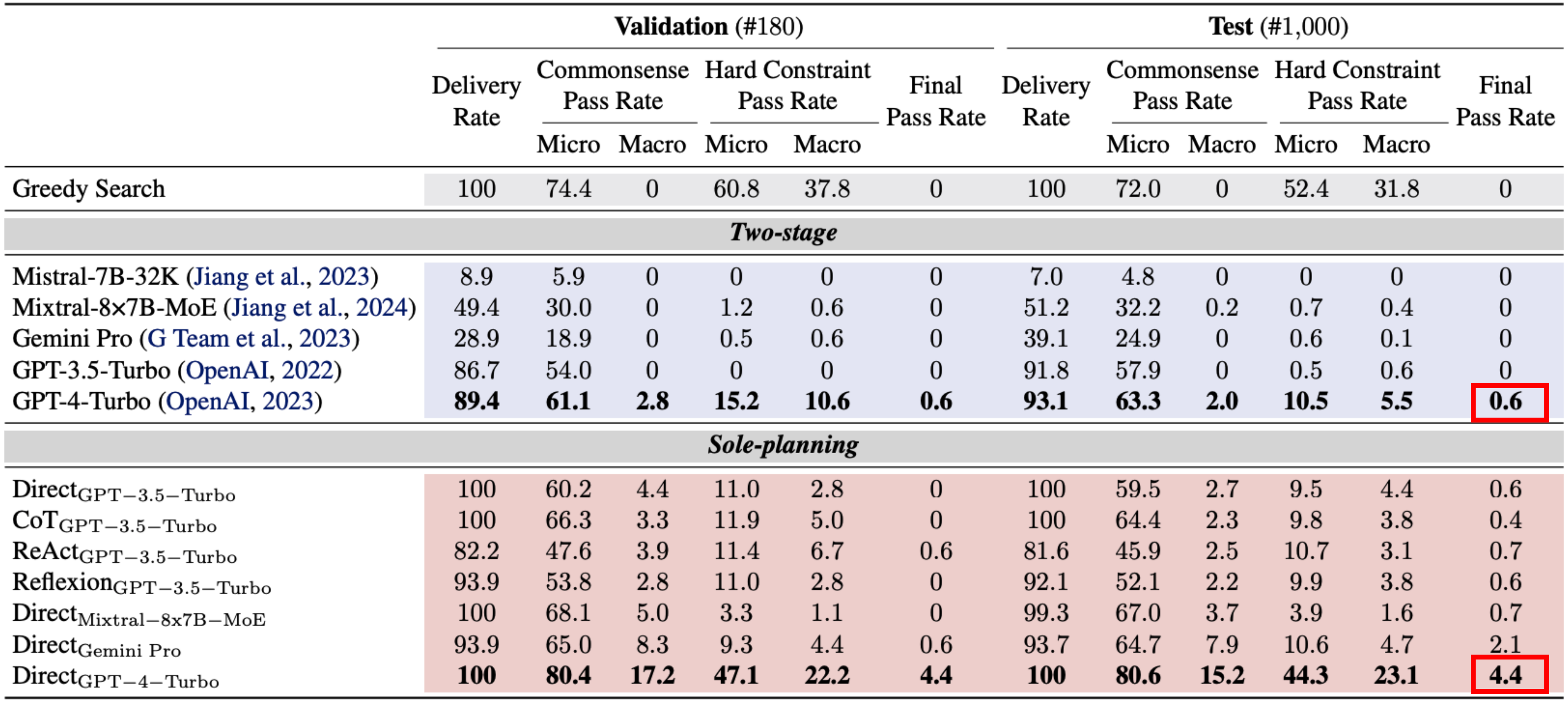
[J. Xie et al, *TravelPlanner: A Benchmark for Real-World Planning with Language Agents*, ICML'24 (**Spotlight**)]

Using SoTA LLMs for Travel Planning (not great)

Even SoTA LLMs struggle for such hard planning problems

First tool use,
Then plan the travel

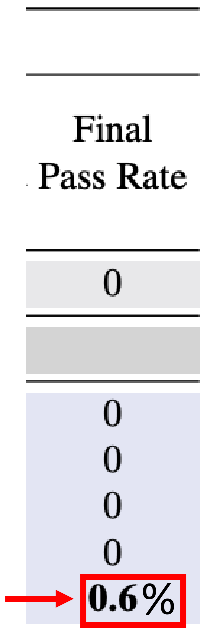
Ground-truth tool use,
Then plan the travel



	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	
Greedy Search	100	74.4	0	60.8	37.8	0	100	72.0	0	52.4	31.8	0
Two-stage												
Mistral-7B-32K (Jiang et al., 2023)	8.9	5.9	0	0	0	0	7.0	4.8	0	0	0	0
Mixtral-8×7B-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
GPT-4-Turbo (OpenAI, 2023)	**89.4**	**61.1**	**2.8**	**15.2**	**10.6**	**0.6**	**93.1**	**63.3**	**2.0**	**10.5**	**5.5**	**0.6**
Sole-planning												
Direct$_{\text{GPT-3.5-Turbo}}$	100	60.2	4.4	11.0	2.8	0	100	59.5	2.7	9.5	4.4	0.6
CoT$_{\text{GPT-3.5-Turbo}}$	100	66.3	3.3	11.9	5.0	0	100	64.4	2.3	9.8	3.8	0.4
ReAct$_{\text{GPT-3.5-Turbo}}$	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$_{\text{GPT-3.5-Turbo}}$	93.9	53.8	2.8	11.0	2.8	0	92.1	52.1	2.2	9.9	3.8	0.6
Direct$_{\text{Mixtral-8x7B-MoE}}$	100	68.1	5.0	3.3	1.1	0	99.3	67.0	3.7	3.9	1.6	0.7
Direct$_{\text{Gemini Pro}}$	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$_{\text{GPT-4-Turbo}}$	**100**	**80.4**	**17.2**	**47.1**	**22.2**	**4.4**	**100**	**80.6**	**15.2**	**44.3**	**23.1**	**4.4**

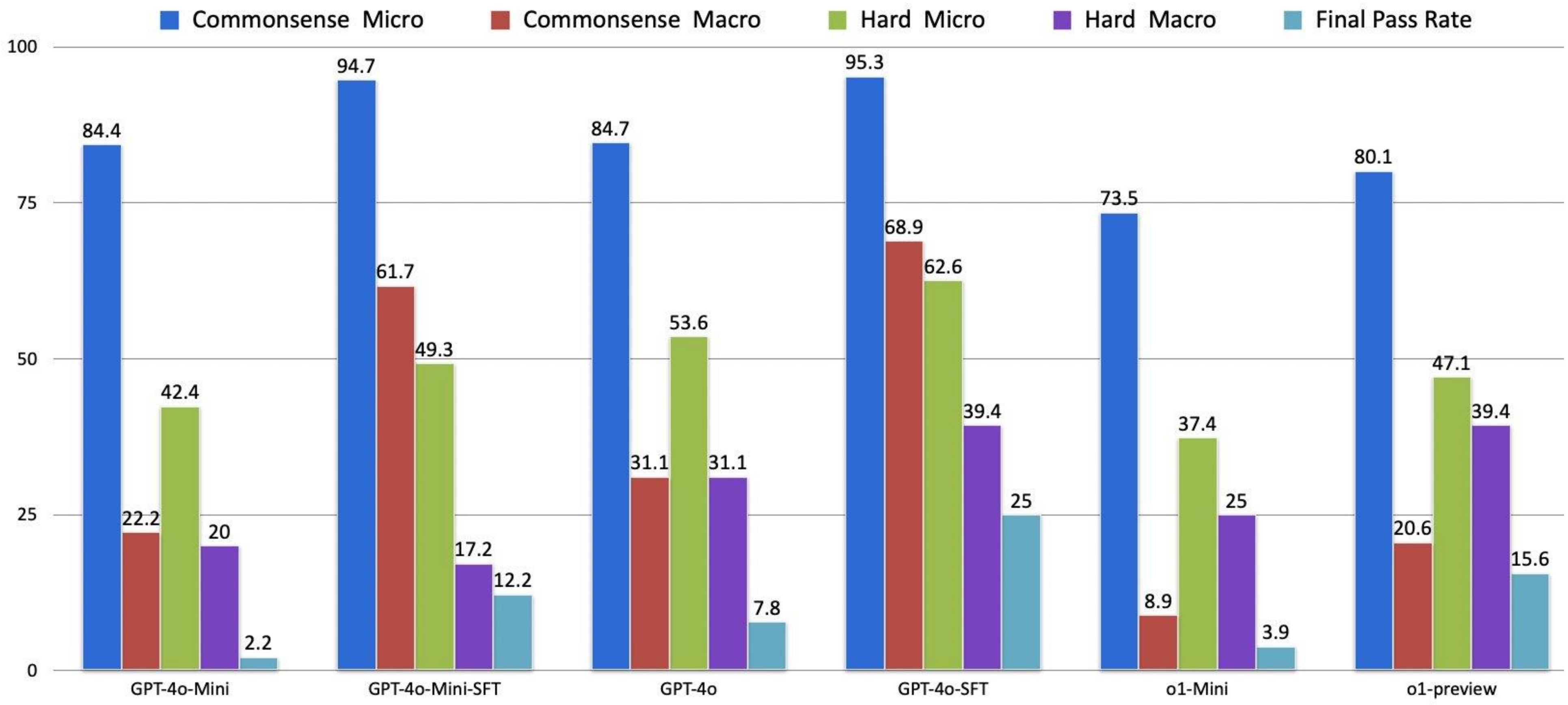
[J. Xie et al, *TravelPlanner: A Benchmark for Real-World Planning with Language Agents*, ICML'24 (**Spotlight**)]

How about o1?



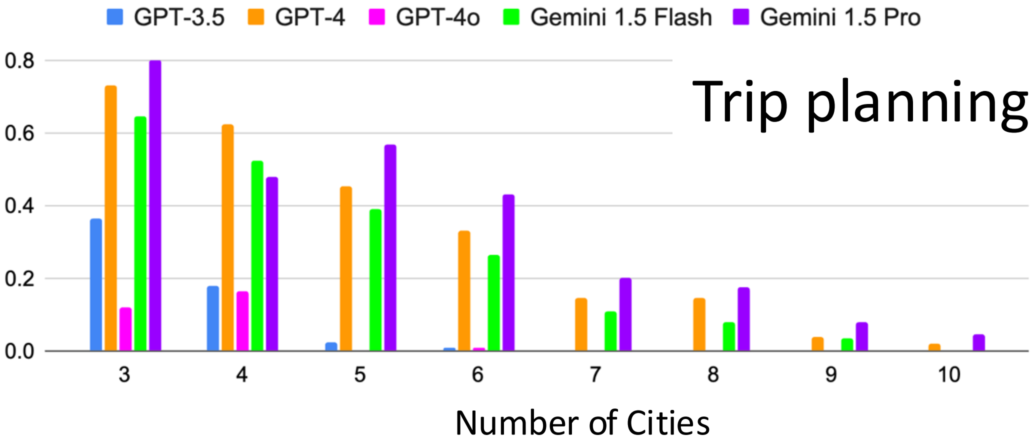
Final Pass Rate
0
0
0
0
0
0.6%

GPT-4-turbo → **0.6%**

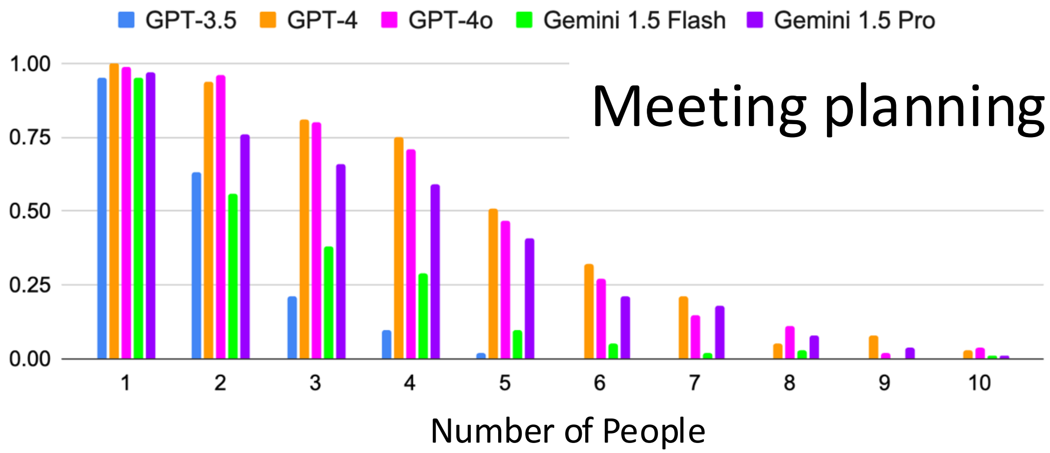


[J. Xie et al, *TravelPlanner: A Benchmark for Real-World Planning with Language Agents*, ICML'24 (**Spotlight**)]

LLM planning is still a hard problem

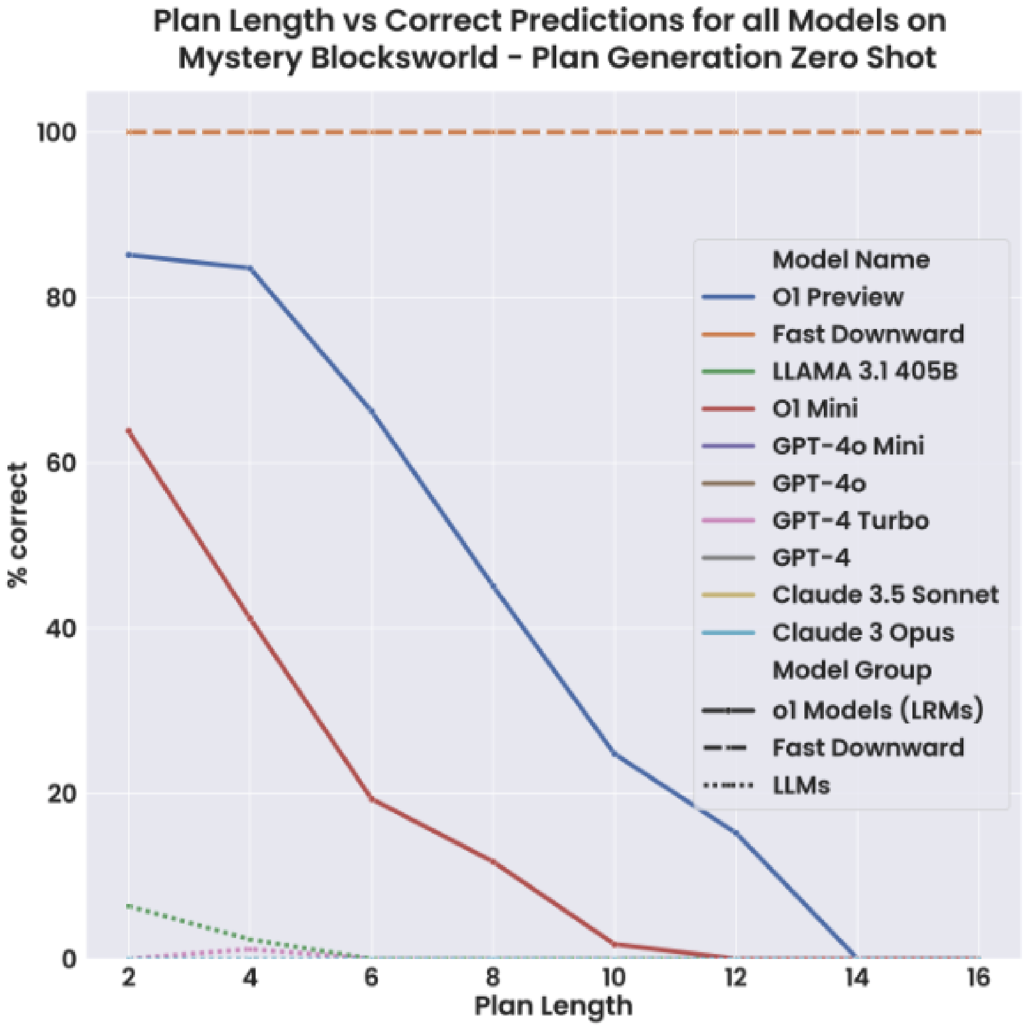


Trip planning



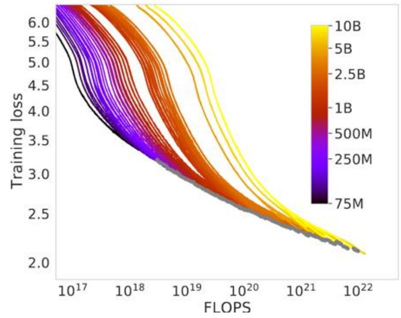
Meeting planning

[H. S. Zheng et al, *NATURAL PLAN: Benchmarking LLMs on Natural Language Planning*, arXiv'24]

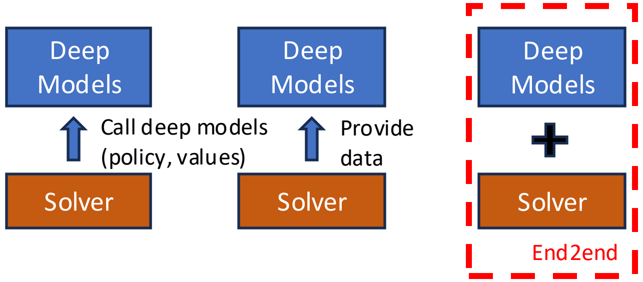


What are the Solutions?

What are the Solutions?



Option **One**: Scaling Law


Deep Models

Call deep models (policy, values)

Solver

Deep Models

Provide data

Solver

Deep Models

+

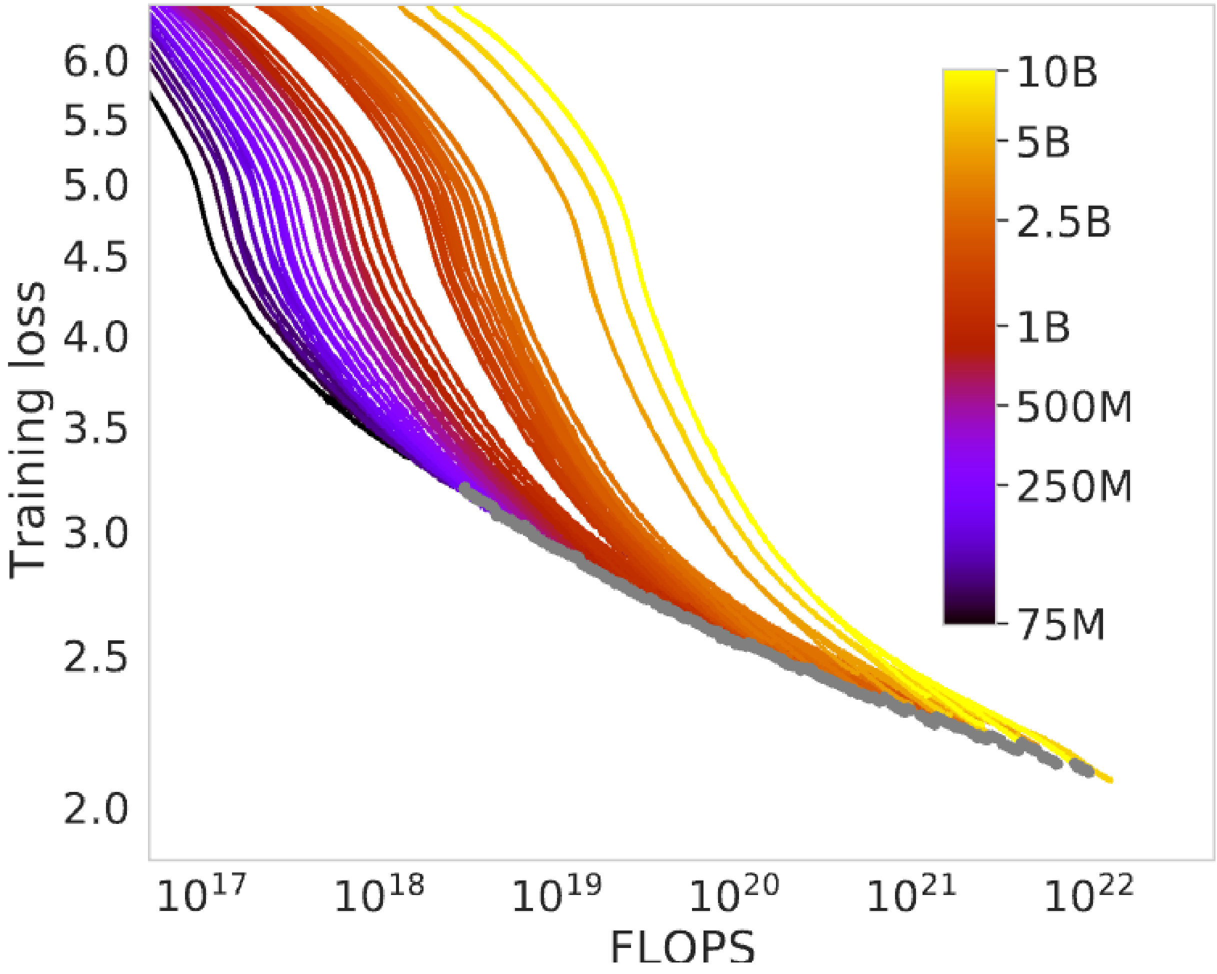
Solver

End2end

Option **Two**: Hybrid System

Option **Three**: Emerging Symbolic Structure from Neural network

Option **One**: The Scaling Law



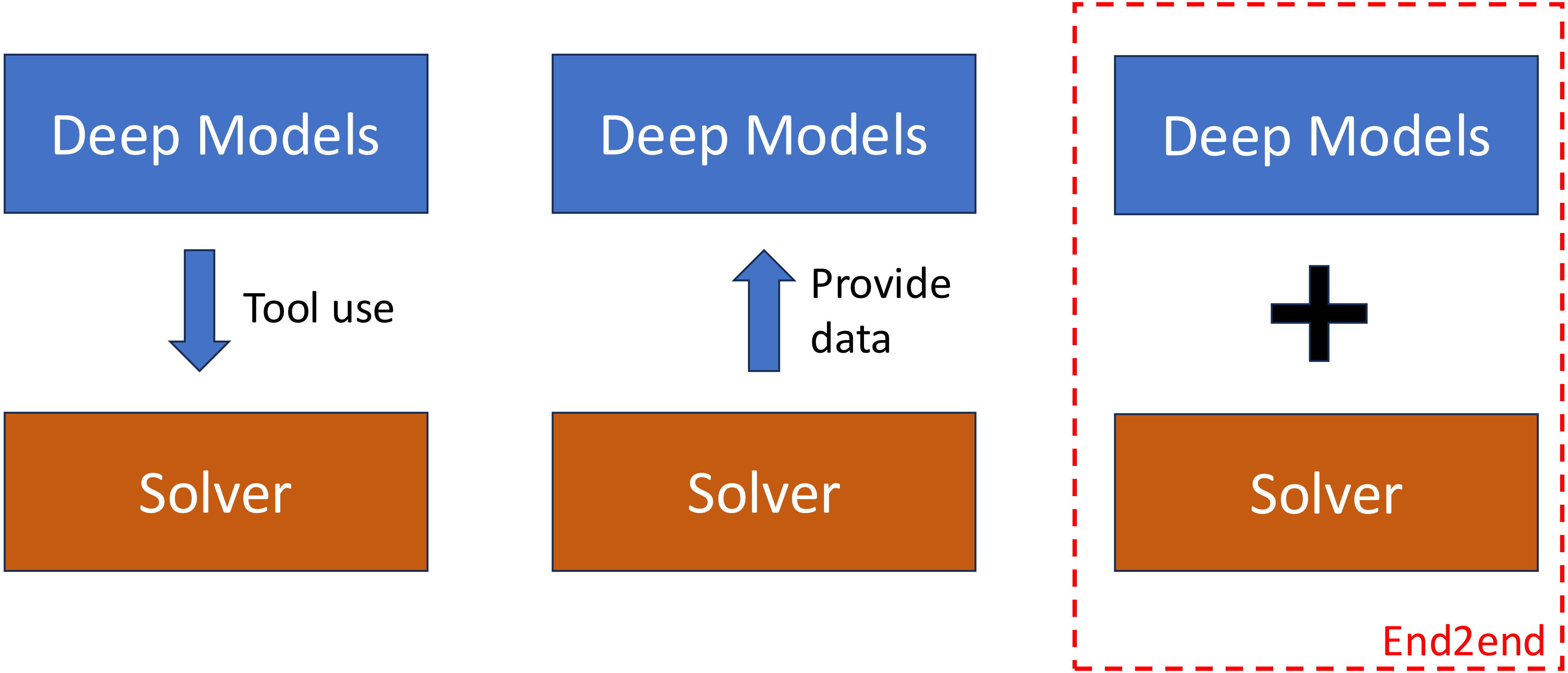
More data
More compute
Larger models

Very expensive

Does that work for reasoning/planning?

[J. Hoffmann*, S. Borgeaud*, A. Mensch* et al, *Training Compute-Optimal Large Language Models*]

Option **Two**: Hybrid Systems



Deep Models

Tool use

Solver

Deep Models

Provide data

Solver

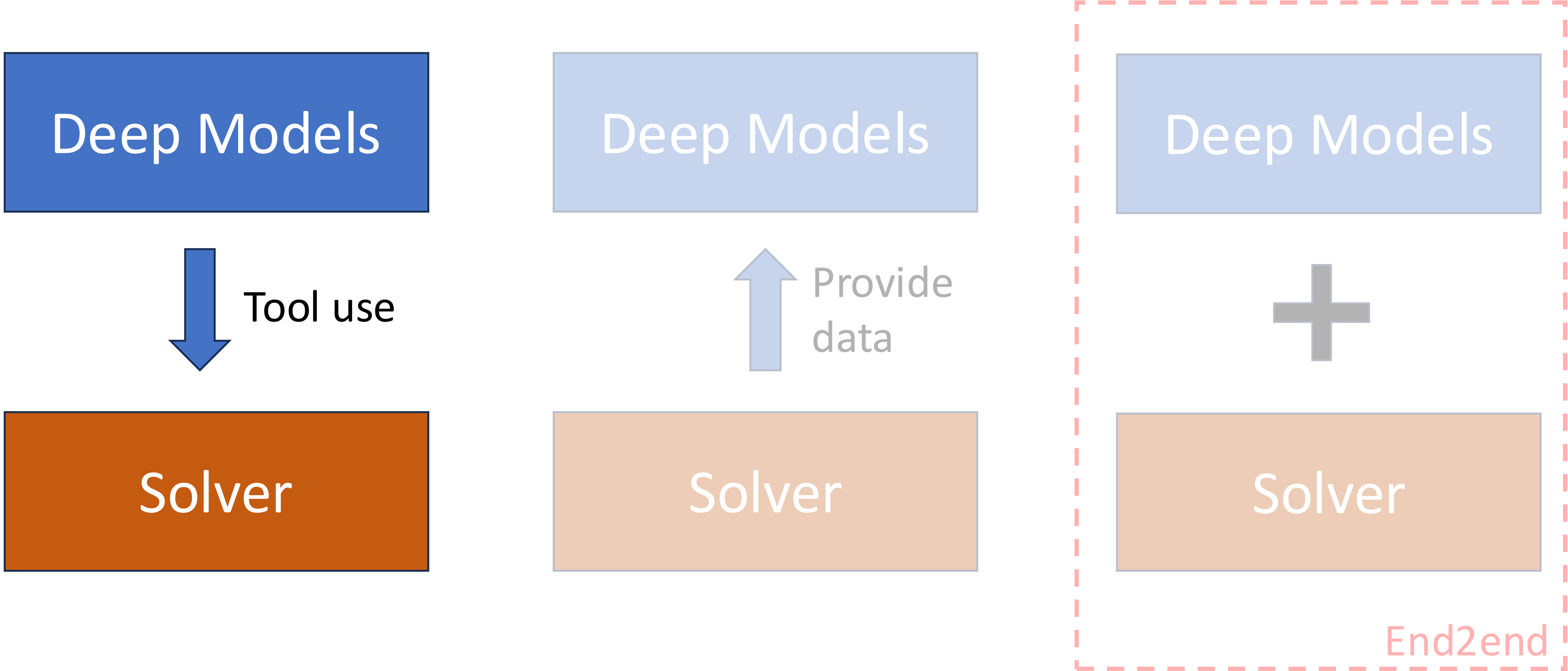
Deep Models

+

Solver

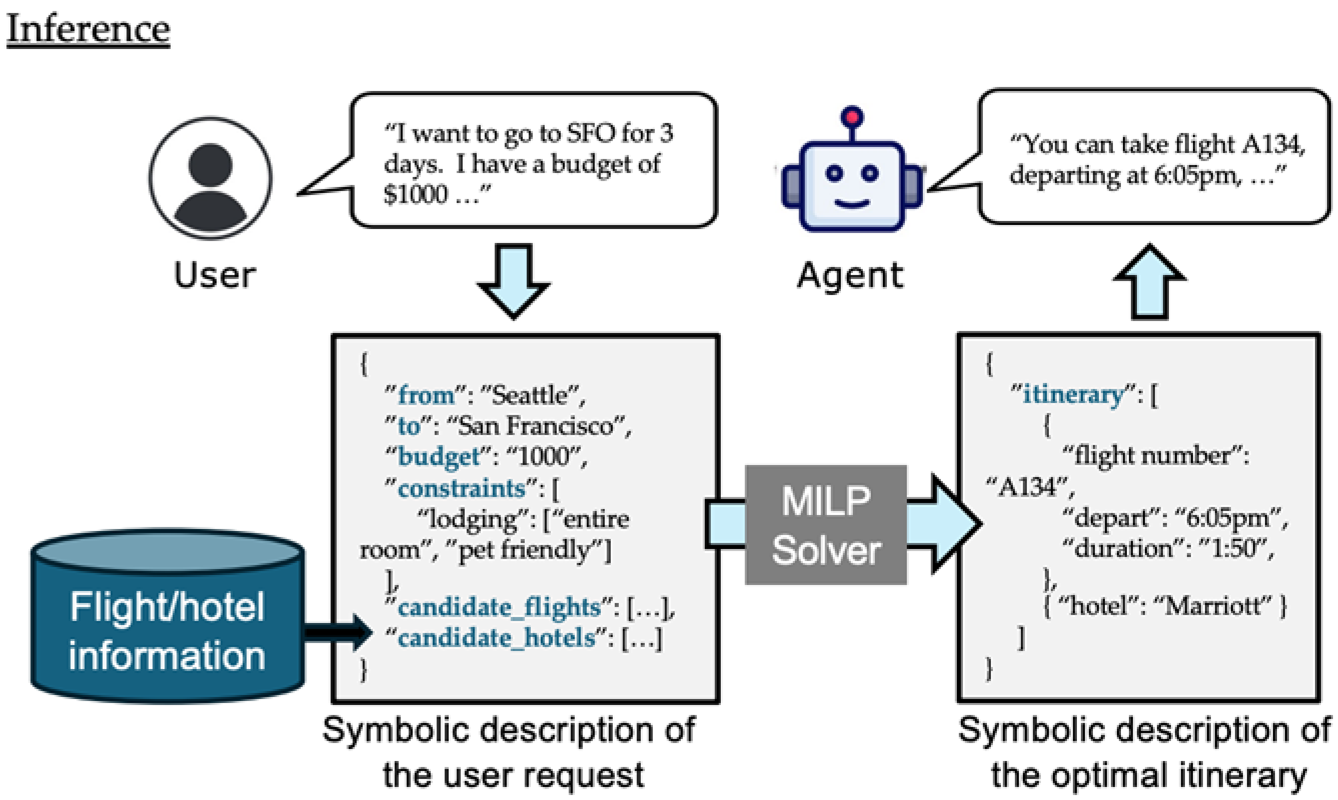
End2end

Option **Two**: Hybrid Systems



Language-Driven Guaranteed Travel Planning

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


Inference
User
"I want to go to SFO for 3 days. I have a budget of $1000 …"
Agent
"You can take flight A134, departing at 6:05pm, …"
Flight/hotel information
MILP Solver

```
{
   ”from”: ”Seattle”,
   ”to”: “San Francisco”,
   “budget”: “1000”,
   ”constraints“: [
      “lodging”: [“entire
room”, ”pet friendly”]
   ],
   ”candidate_flights”: […],
   “candidate_hotels”: […]
}
```

Symbolic description of the user request

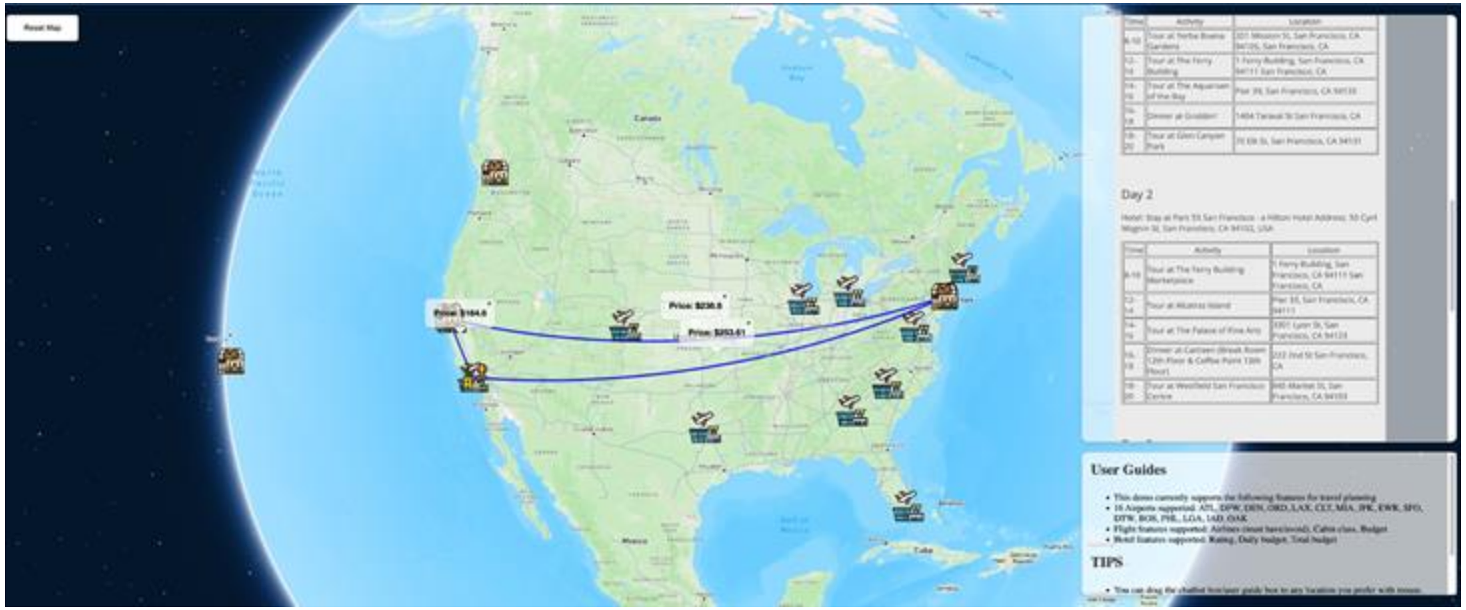
```
{
   ”itinerary”: [
      {
         “flight number”:
“A134”,
         “depart”: “6:05pm”,
         “duration”: ”1:50”,
      },
      { “hotel”: “Marriott” }
   ]
}
```

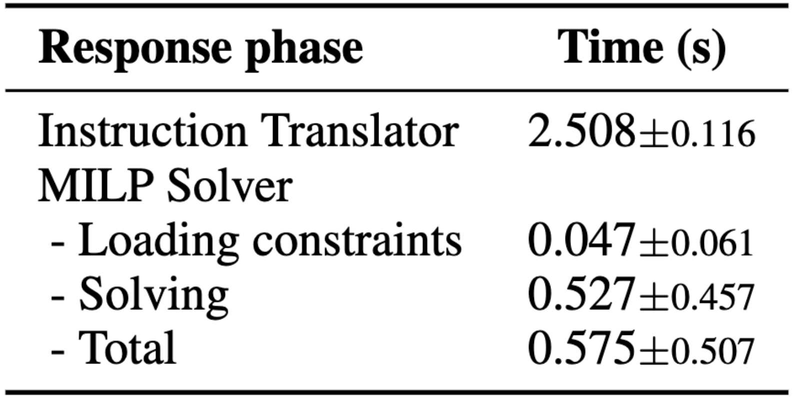
Symbolic description of the optimal itinerary

- 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.

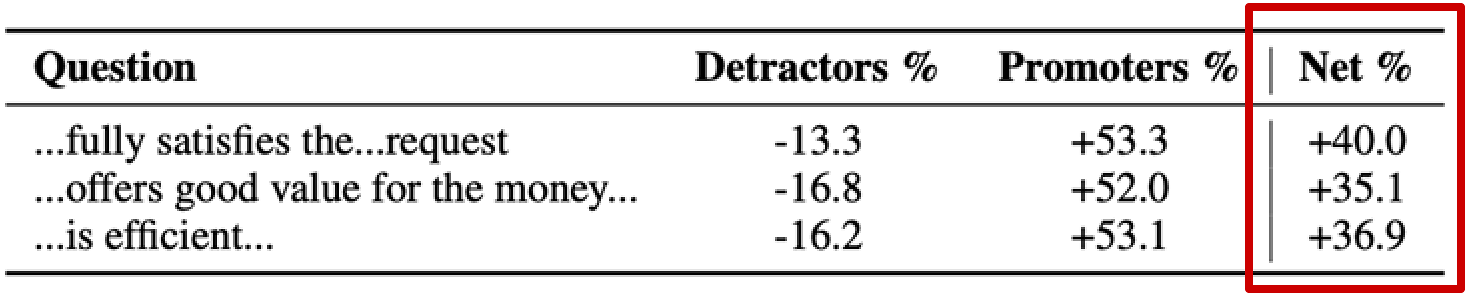
Ju et al, *To the Globe (TTG): Towards Language-Driven Guaranteed Travel Planning (EMNLP’24 Demo)*

Experiments (End-to-end Human Evaluation)





Response phase	Time (s)
Instruction Translator	2.508±0.116
MILP Solver	
- Loading constraints	0.047±0.061
- Solving	0.527±0.457
- Total	0.575±0.507

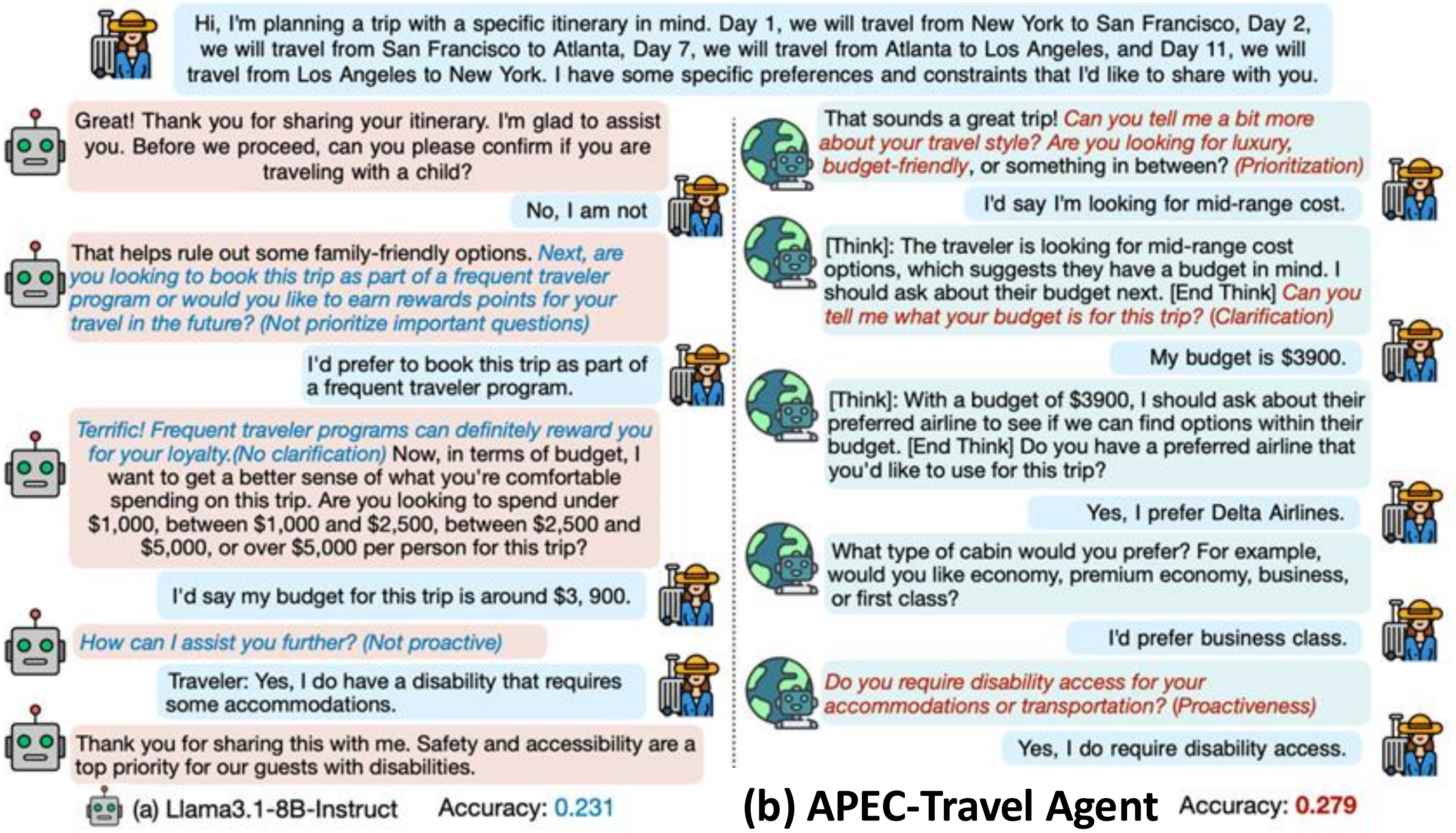


Question	Detractors %	Promoters %	Net %
...fully satisfies the...request	-13.3	+53.3	+40.0
...offers good value for the money...	-16.8	+52.0	+35.1
...is efficient...	-16.2	+53.1	+36.9

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

Ju et al, *To the Globe (TTG): Towards Language-Driven Guaranteed Travel Planning (EMNLP'24 Demo)*

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!**

[Jiang et al, *Towards Full Delegation: Designing Ideal Agentic Behaviors for Travel Planning]*

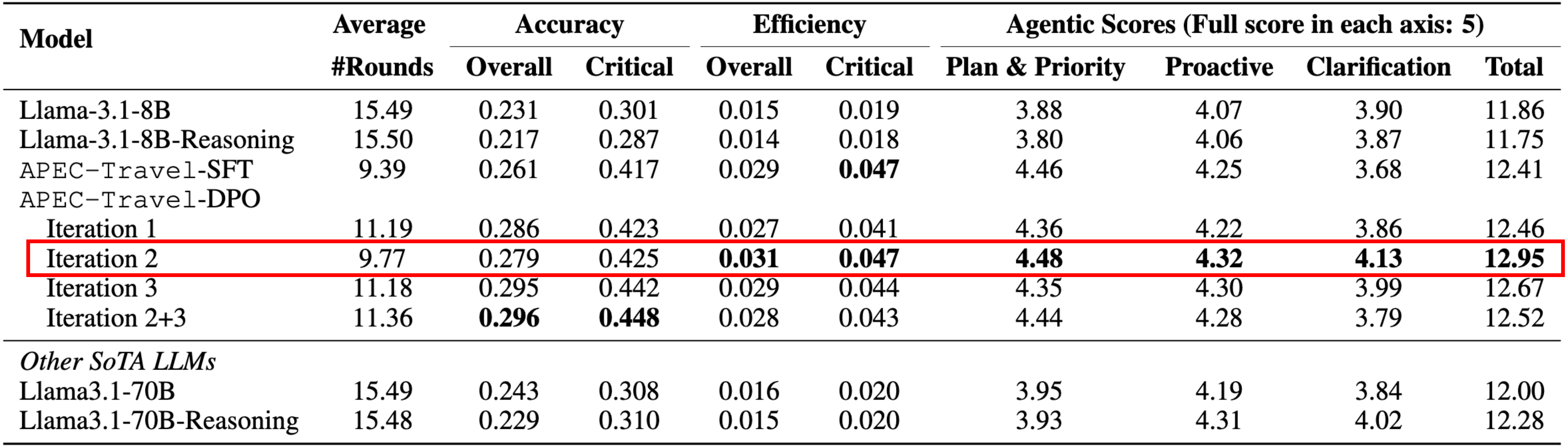
APEC Agent Constitution

LLM agents should be evaluated and optimized not only based on final outcome, but also based on the **procedure** of how agents achieve the goal.

- **Accuracy**. The quality of the final solution that the agent provides (e.g., number of questions that are answered correctly).
- **Proactivity**. Whether the agent proactively collects useful information to solve the task. Such information may be public or private, vague or precise, explicitly provided or inferred from requests.
- **Efficiency**. Whether the agent can achieve its goal with a minimal number of interactions (e.g., number of questions asked, API calls and tool uses).
- **Credibility**. The reliability with which agents achieve positive outcomes (e.g., amount of hallucination and inconsistency).

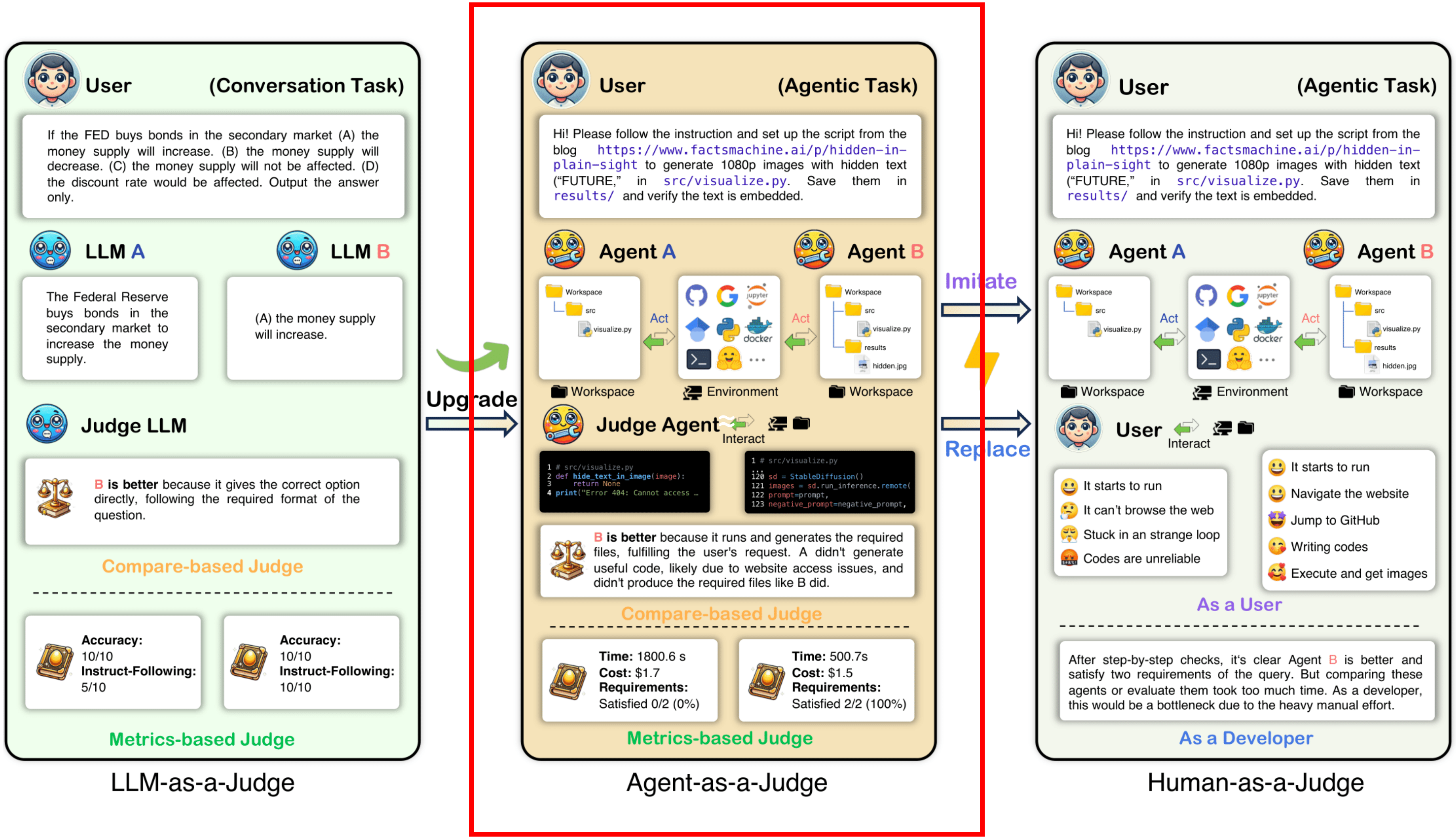
[Jiang et al, *Towards Full Delegation: Designing Ideal Agentic Behaviors for Travel Planning*]

Using Agent Constitution to fine-tune the models



Model	**Average**	**Accuracy**		**Efficiency**		**Agentic Scores (Full score in each axis: 5)**			
	#Rounds	**Overall**	**Critical**	**Overall**	**Critical**	**Plan & Priority**	**Proactive**	**Clarification**	**Total**
Llama-3.1-8B	15.49	0.231	0.301	0.015	0.019	3.88	4.07	3.90	11.86
Llama-3.1-8B-Reasoning	15.50	0.217	0.287	0.014	0.018	3.80	4.06	3.87	11.75
`APEC-Travel`-SFT	9.39	0.261	0.417	0.029	**0.047**	4.46	4.25	3.68	12.41
`APEC-Travel`-DPO									
Iteration 1	11.19	0.286	0.423	0.027	0.041	4.36	4.22	3.86	12.46
Iteration 2	9.77	0.279	0.425	**0.031**	**0.047**	**4.48**	**4.32**	**4.13**	**12.95**
Iteration 3	11.18	0.295	0.442	0.029	0.044	4.35	4.30	3.99	12.67
Iteration 2+3	11.36	**0.296**	**0.448**	0.028	0.043	4.44	4.28	3.79	12.52
Other SoTA LLMs									
Llama3.1-70B	15.49	0.243	0.308	0.016	0.020	3.95	4.19	3.84	12.00
Llama3.1-70B-Reasoning	15.48	0.229	0.310	0.015	0.020	3.93	4.31	4.02	12.28

Agent-as-a-Judge: Evaluate Agents with Agents



If the FED buys bonds in the secondary market (A) the money supply will increase. (B) the money supply will decrease. (C) the money supply will not be affected. (D) the discount rate would be affected. Output the answer only.

The Federal Reserve buys bonds in the secondary market to increase the money supply.

(A) the money supply will increase.

B is better because it gives the correct option directly, following the required format of the question.

Accuracy: 10/10 **Instruct-Following:** 5/10

Accuracy: 10/10 **Instruct-Following:** 10/10

Hi! Please follow the instruction and set up the script from the blog `https://www.factsmachine.ai/p/hidden-in-plain-sight` to generate 1080p images with hidden text ("FUTURE," in `src/visualize.py`. Save them in `results/` and verify the text is embedded.

```
# src/visualize.py
def hide_text_in_image(image):
    return None
print("Error 404: Cannot access …
```

```
# src/visualize.py
...
sd = StableDiffusion()
images = sd.run_inference.remote(
prompt=prompt,
negative_prompt=negative_prompt,
```

B is better because it runs and generates the required files, fulfilling the user's request. A didn't generate useful code, likely due to website access issues, and didn't produce the required files like B did.

Time: 1800.6 s **Cost:** $1.7 **Requirements:** Satisfied 0/2 (0%)

Time: 500.7s **Cost:** $1.5 **Requirements:** Satisfied 2/2 (100%)

Hi! Please follow the instruction and set up the script from the blog `https://www.factsmachine.ai/p/hidden-in-plain-sight` to generate 1080p images with hidden text ("FUTURE," in `src/visualize.py`. Save them in `results/` and verify the text is embedded.

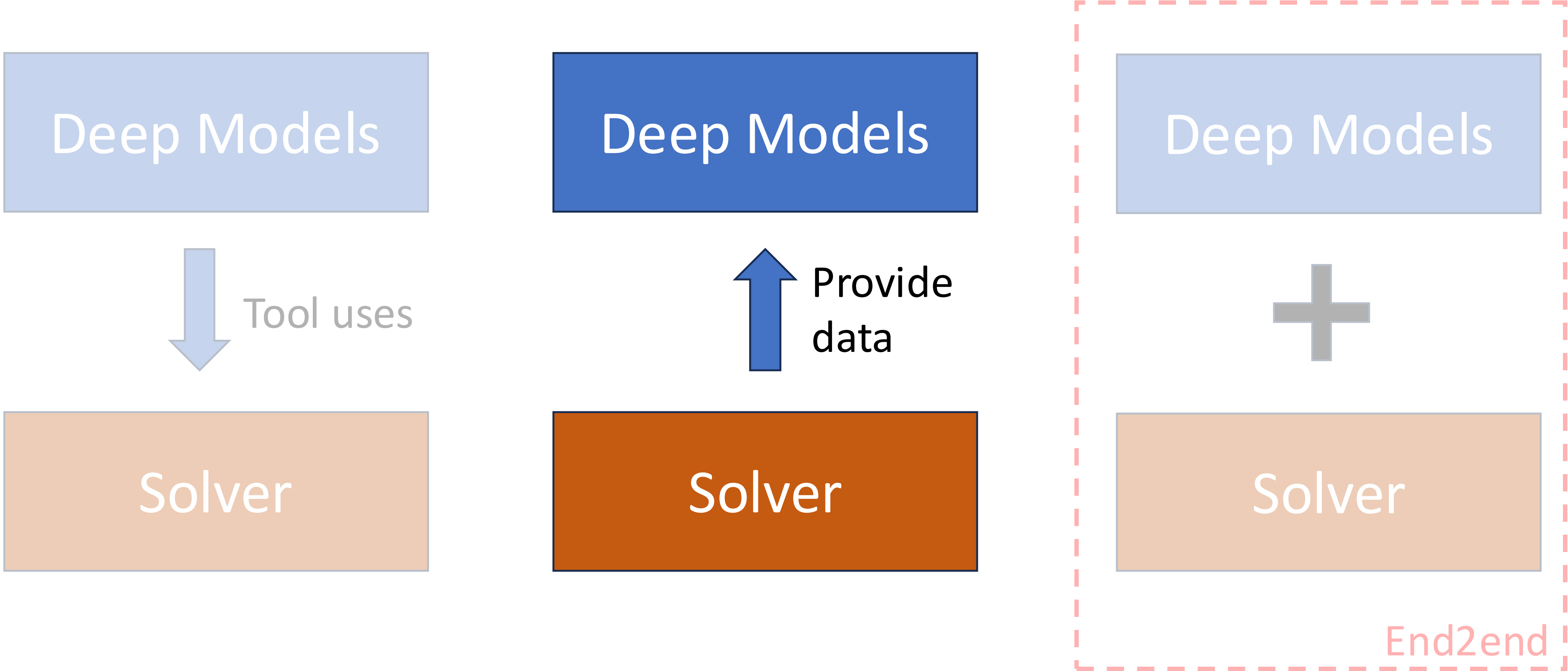
- It starts to run
- It can't browse the web
- Stuck in an strange loop
- Codes are unreliable

- It starts to run
- Navigate the website
- Jump to GitHub
- Writing codes
- Execute and get images

After step-by-step checks, it's clear Agent B is better and satisfy two requirements of the query. But comparing these agents or evaluate them took too much time. As a developer, this would be a bottleneck due to the heavy manual effort.

[M. Zhuge et al, *Agent-as-a-Judge: Evaluate Agents with Agents, arXiv'24*]

Option **Two**: Hybrid Systems



Deep Models

Tool uses

Solver

Deep Models

Provide data

Solver

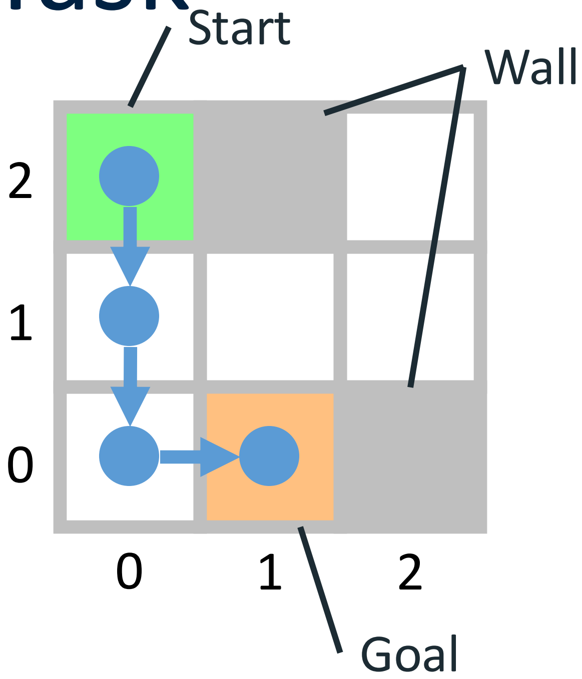
Deep Models

+

Solver

End2end

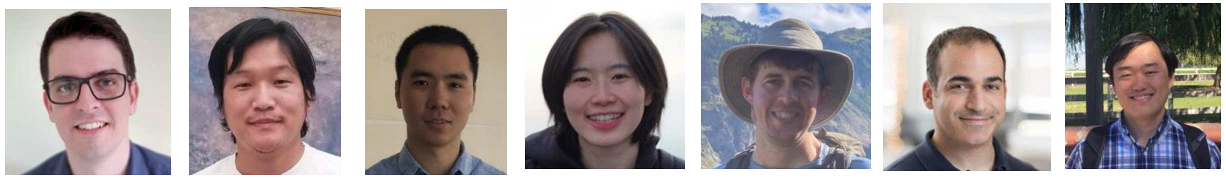
Searchformer: A* Search as a Token Prediction Task



●→ Plan step

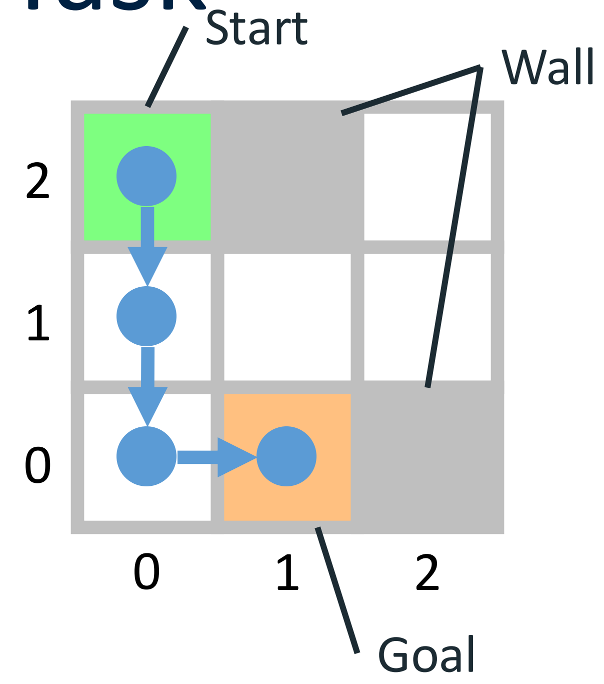
● Frontier state

● Closed state



[L. Lehnert, et al, *Beyond A*: Better Planning with Transformers via Search Dynamics Bootstrapping*, COLM'24]

Searchformer: A* Search as a Token Prediction Task



● → Plan step

● Frontier state

● Closed state

<prompt>

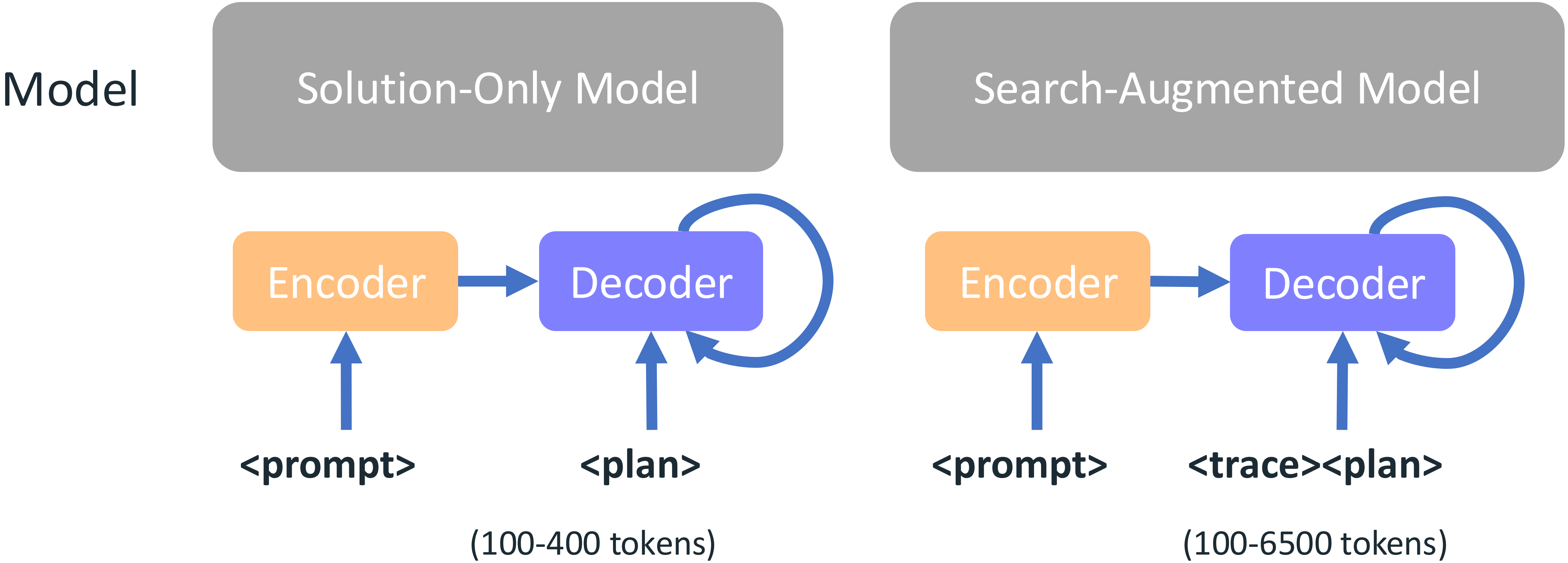
```
bos
start 0 2
goal  1 0
wall  1 2
wall  2 0
eos
```

<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

Encoder

Decoder

<prompt>

<plan>

(100-400 tokens)

Search-Augmented Model

Encoder

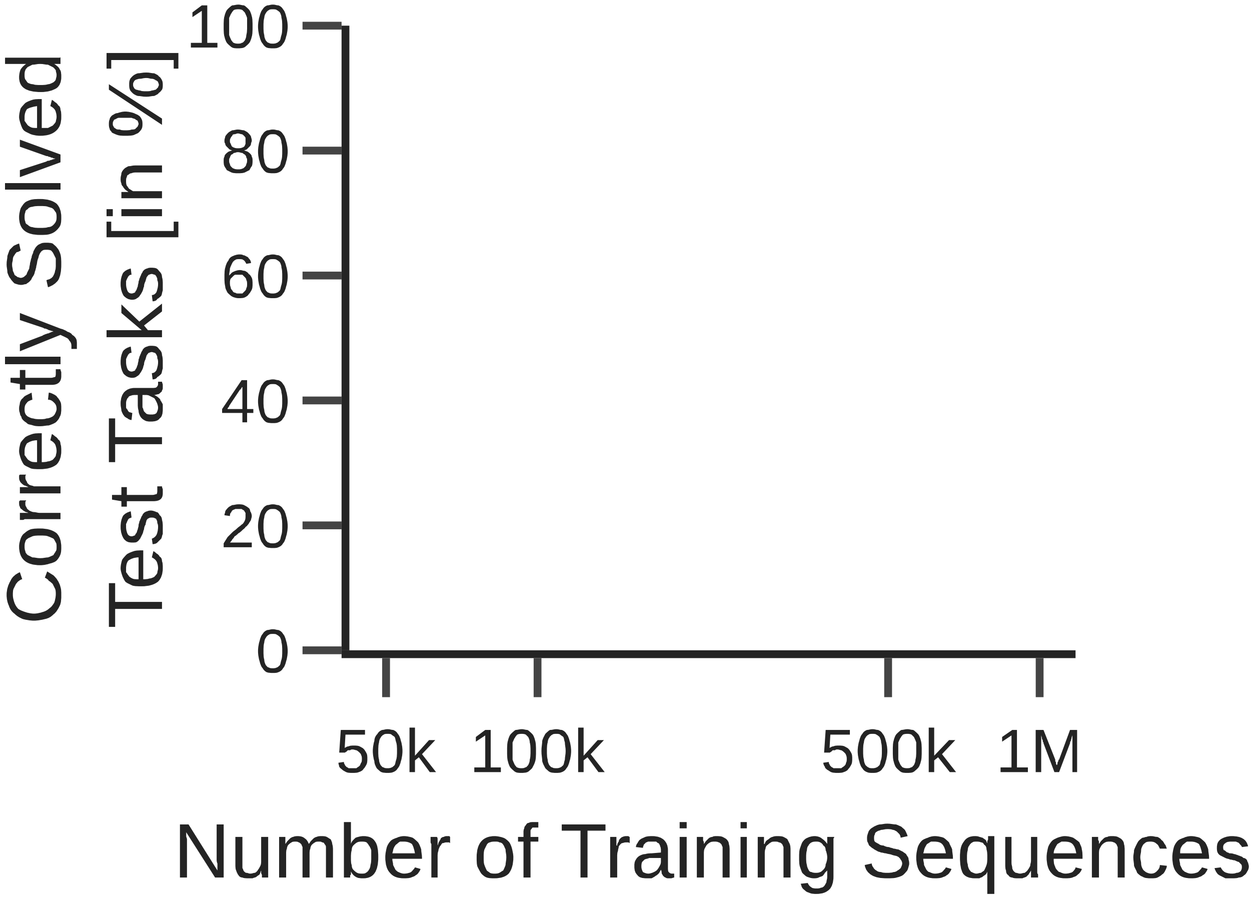
Decoder

<prompt>

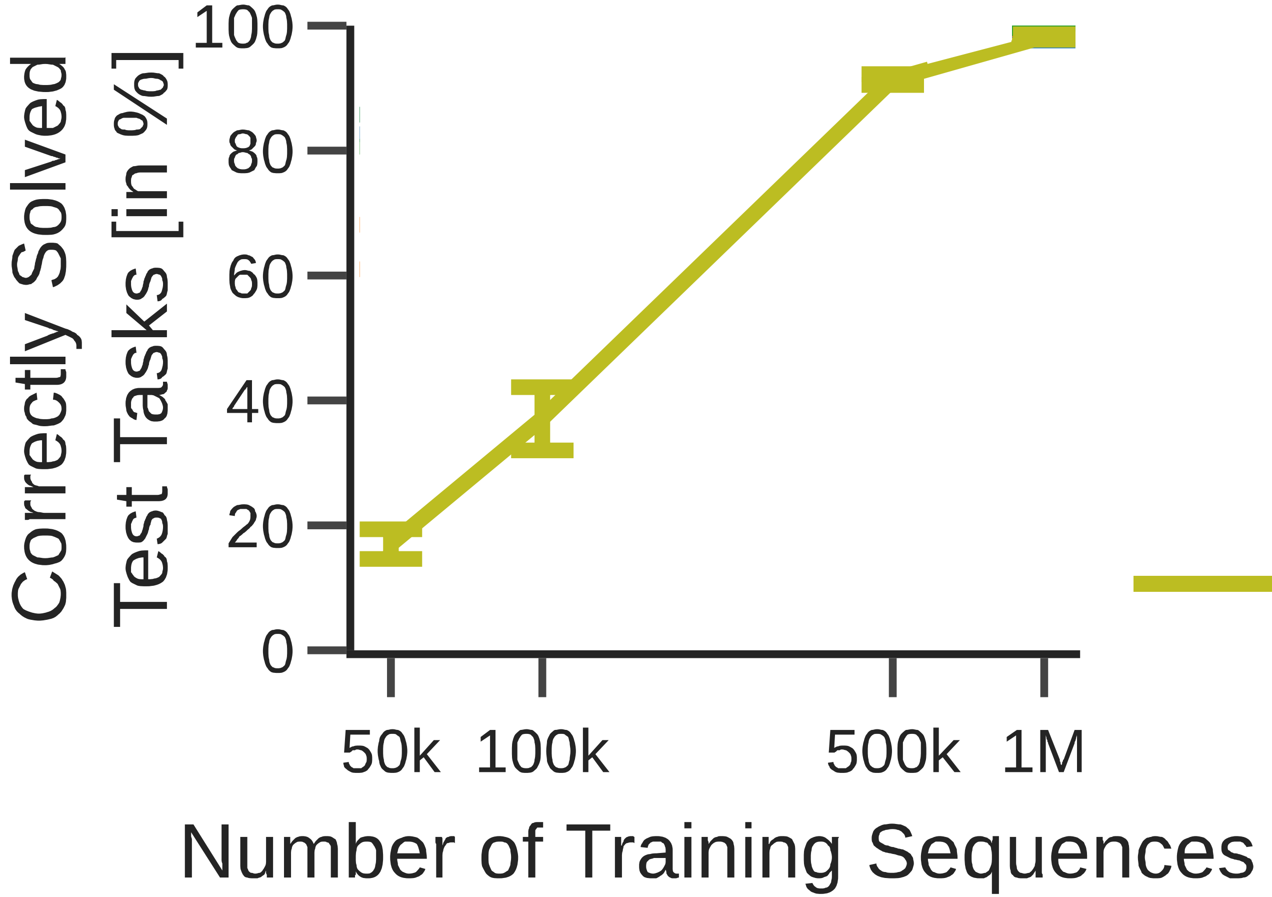
<trace><plan>

(100-6500 tokens)

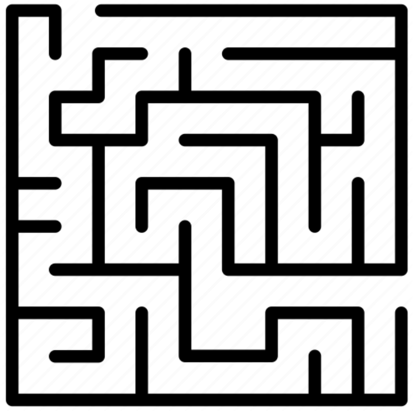
Search-Augmented vs. Solution-Only Models



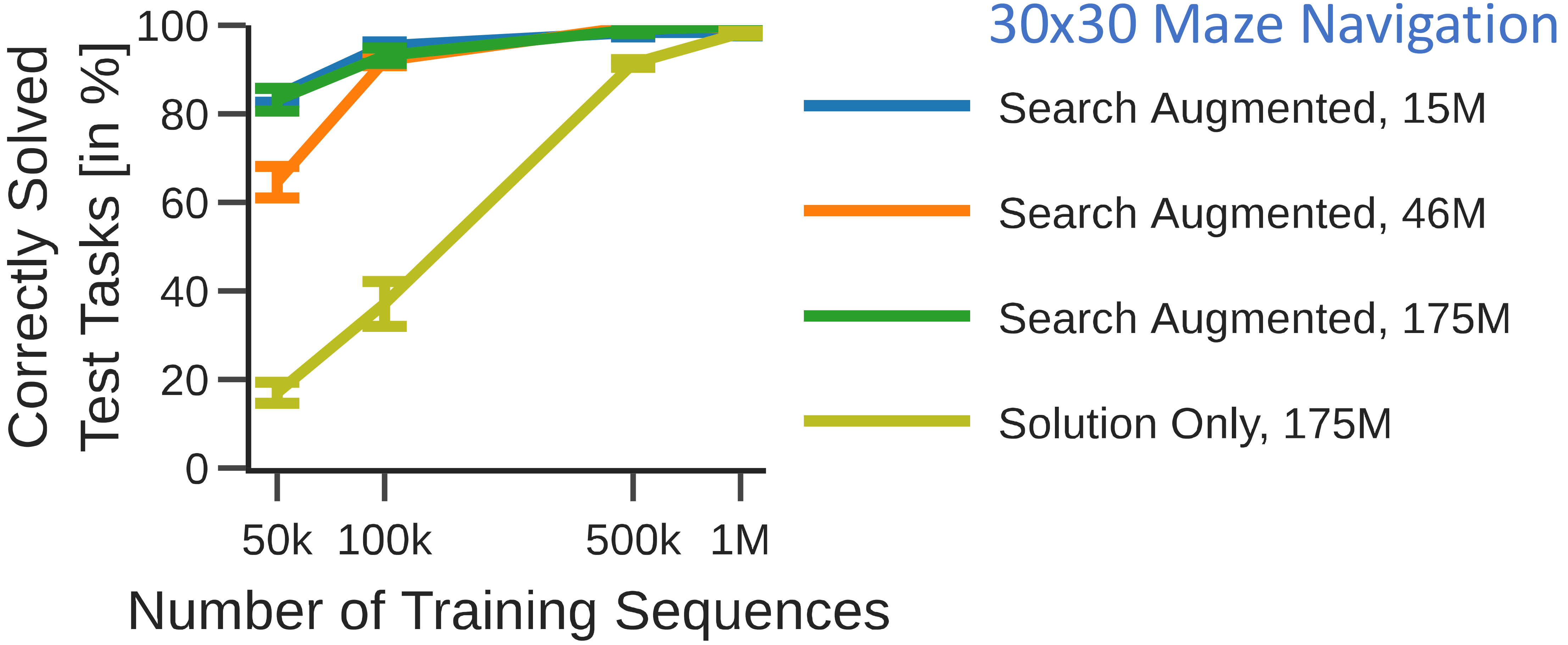
Search-Augmented vs. Solution-Only Models



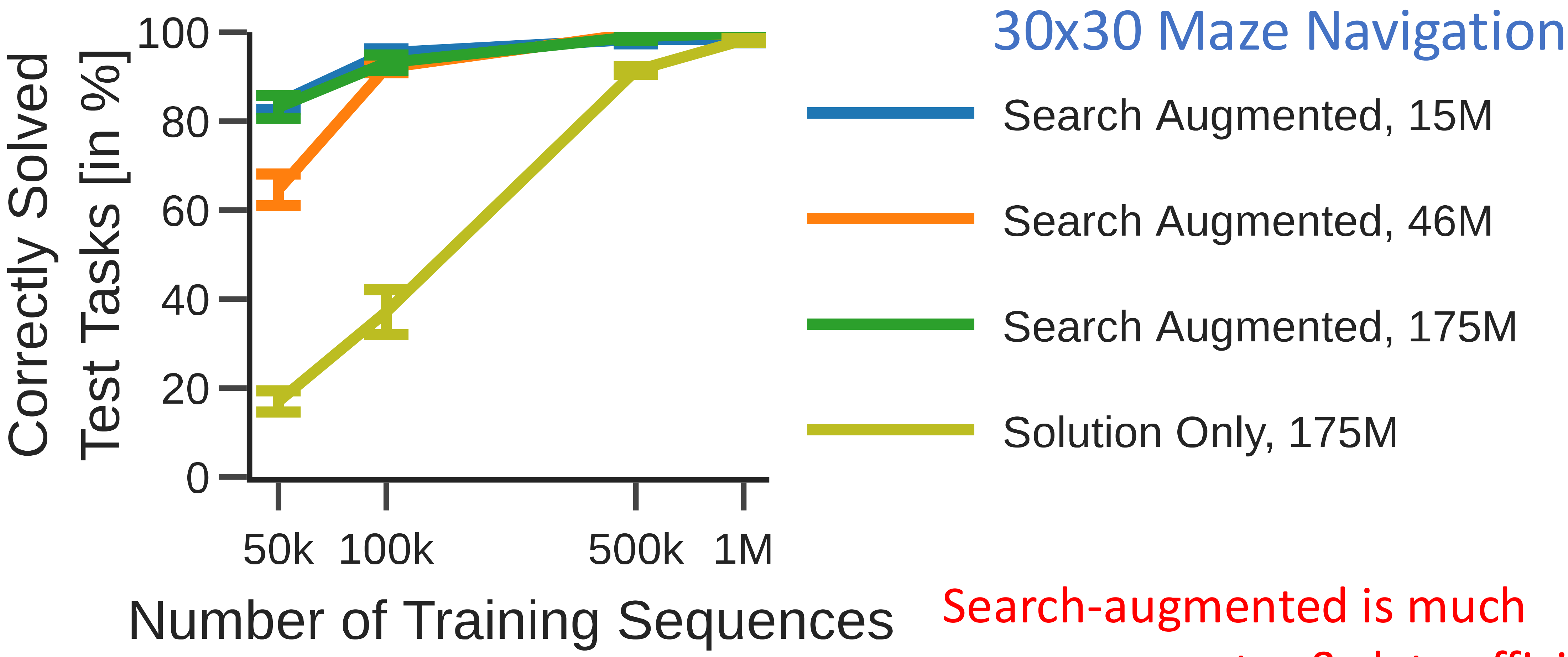
30x30 Maze Navigation



Search-Augmented vs. Solution-Only Models

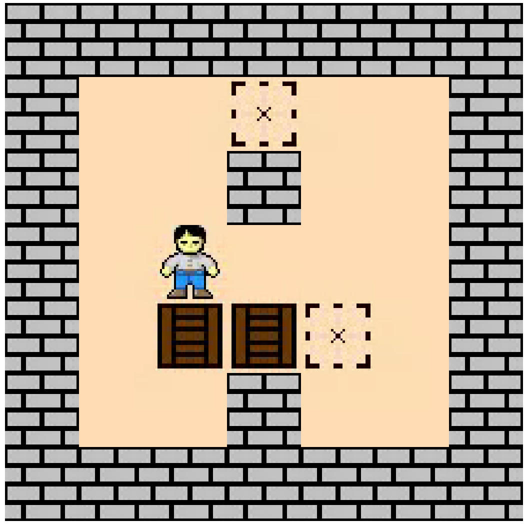


Search-Augmented vs. Solution-Only Models

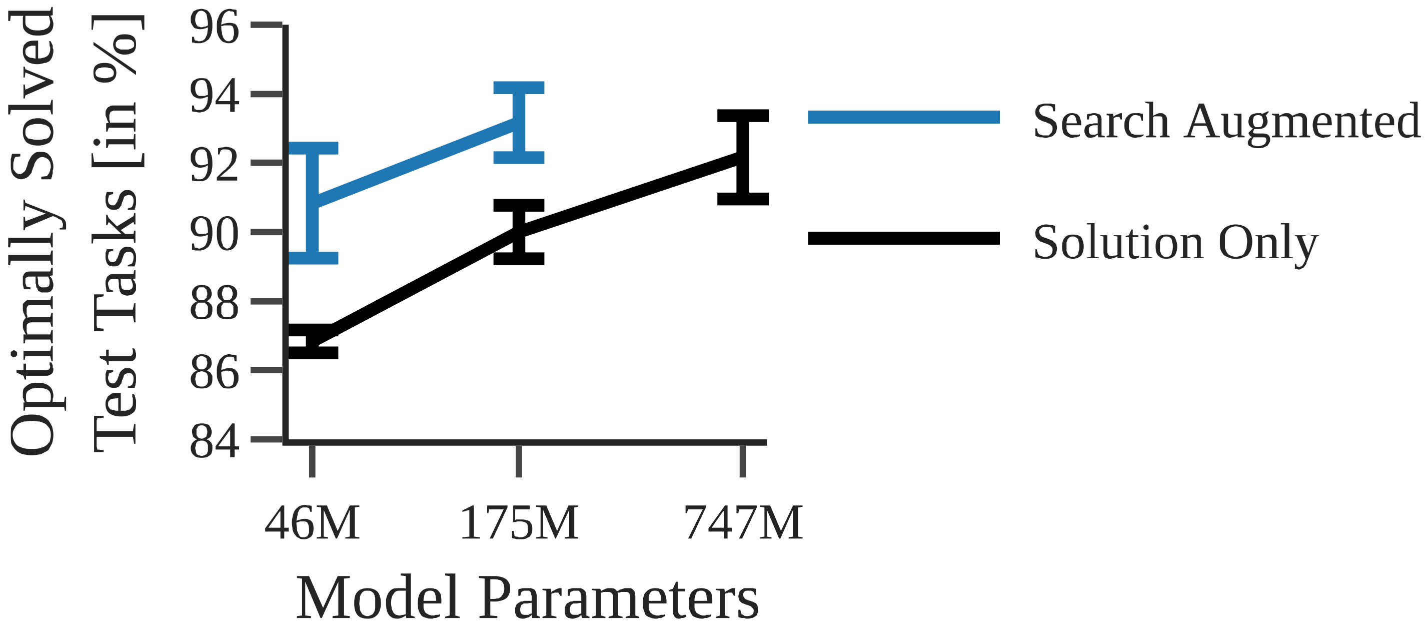


Search-augmented is much more parameter & data efficient!

Search-Augmented vs. Solution-Only Models

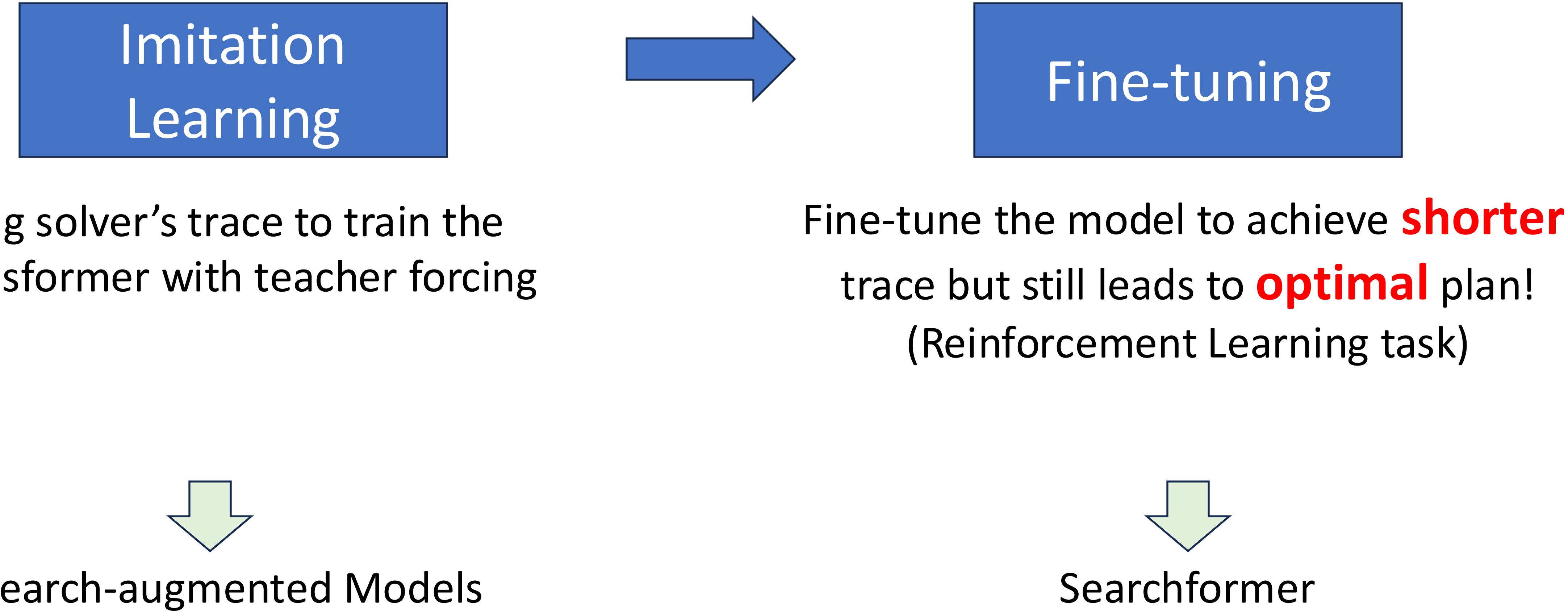


Sokoban



Search-augmented is much more parameter & data efficient!

How to go beyond?



Imitation Learning

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

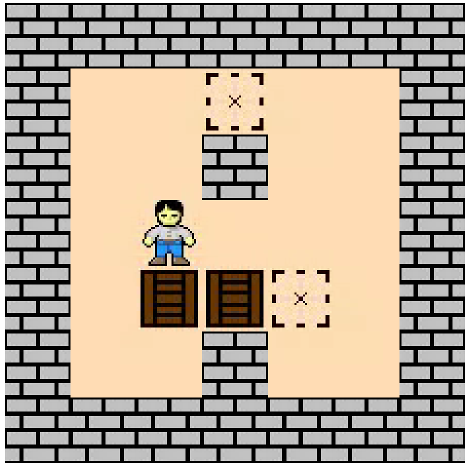
Fine-tuning

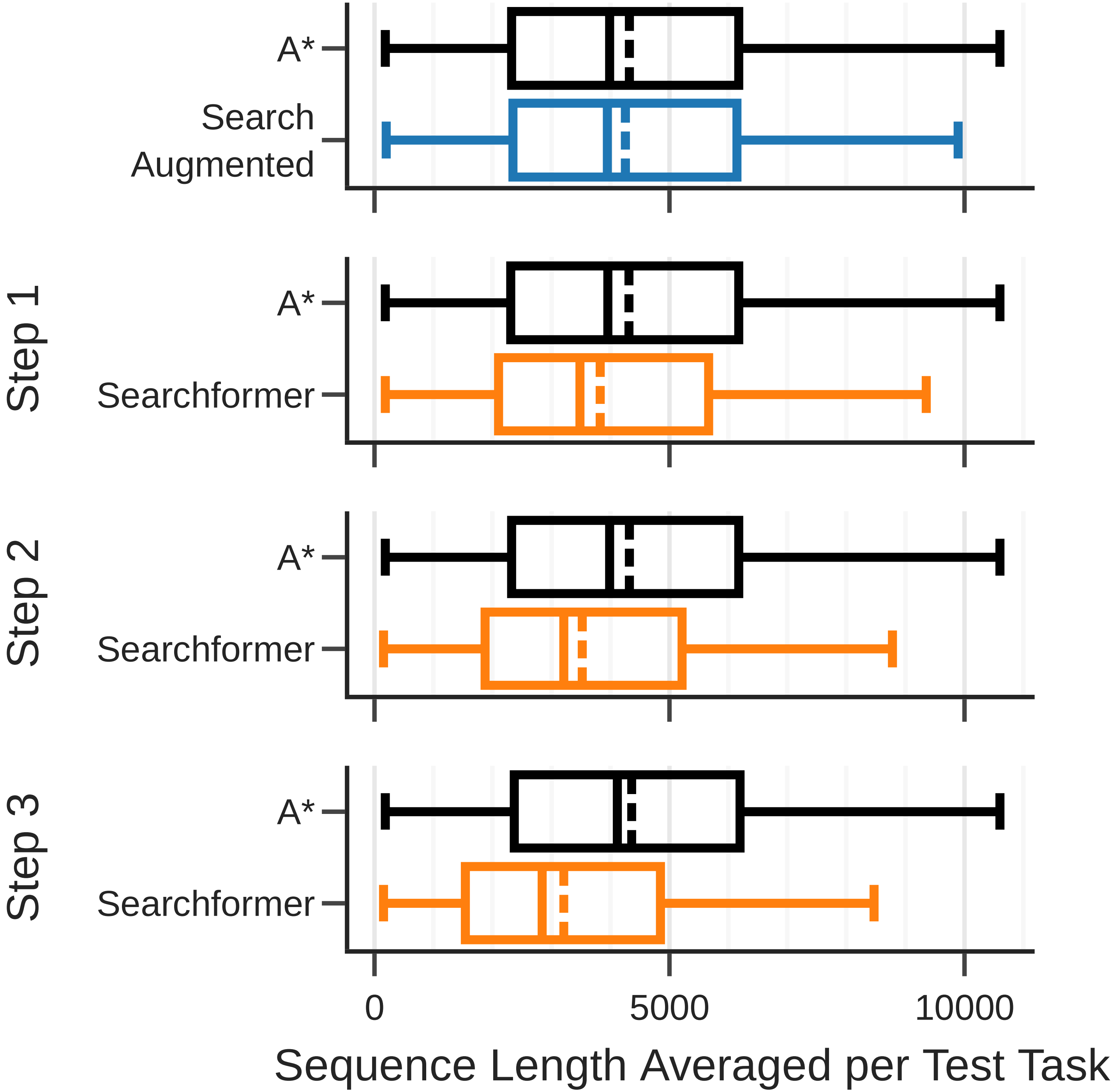
Fine-tune the model to achieve **shorter** trace but still leads to **optimal** plan! (Reinforcement Learning task)

Search-augmented Models

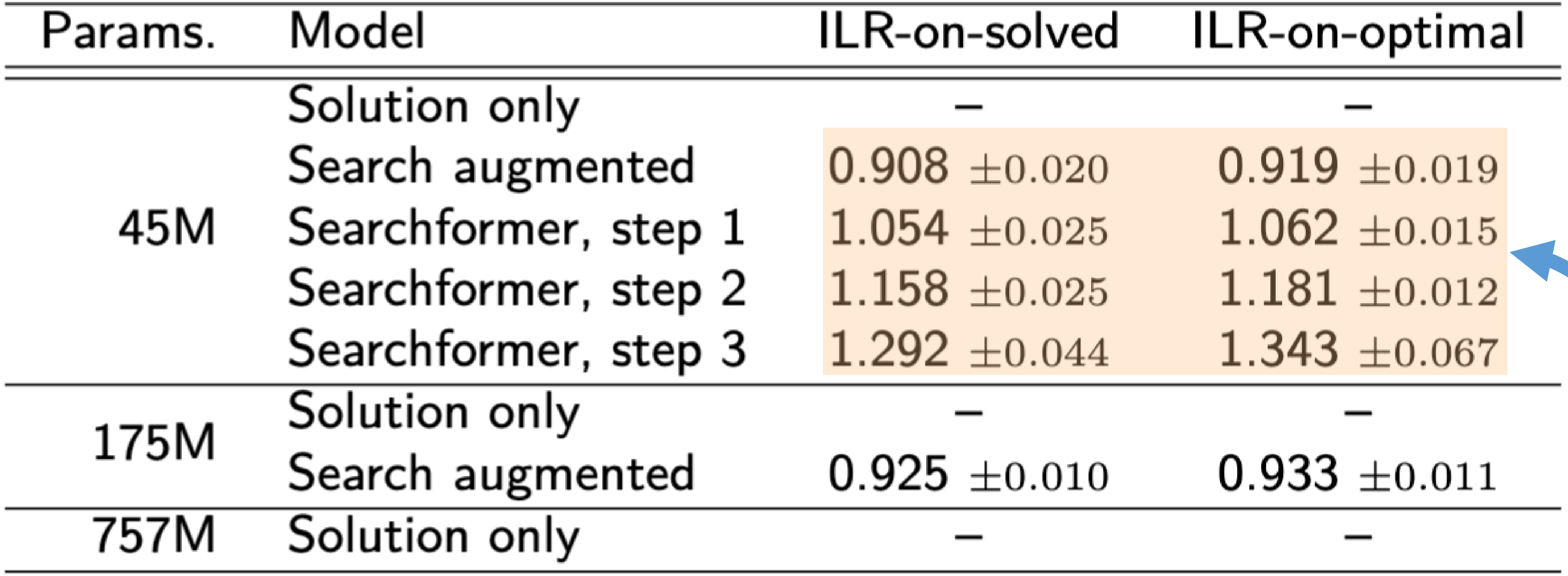
Searchformer

Beyond A*: Improving search dynamics via bootstrapping





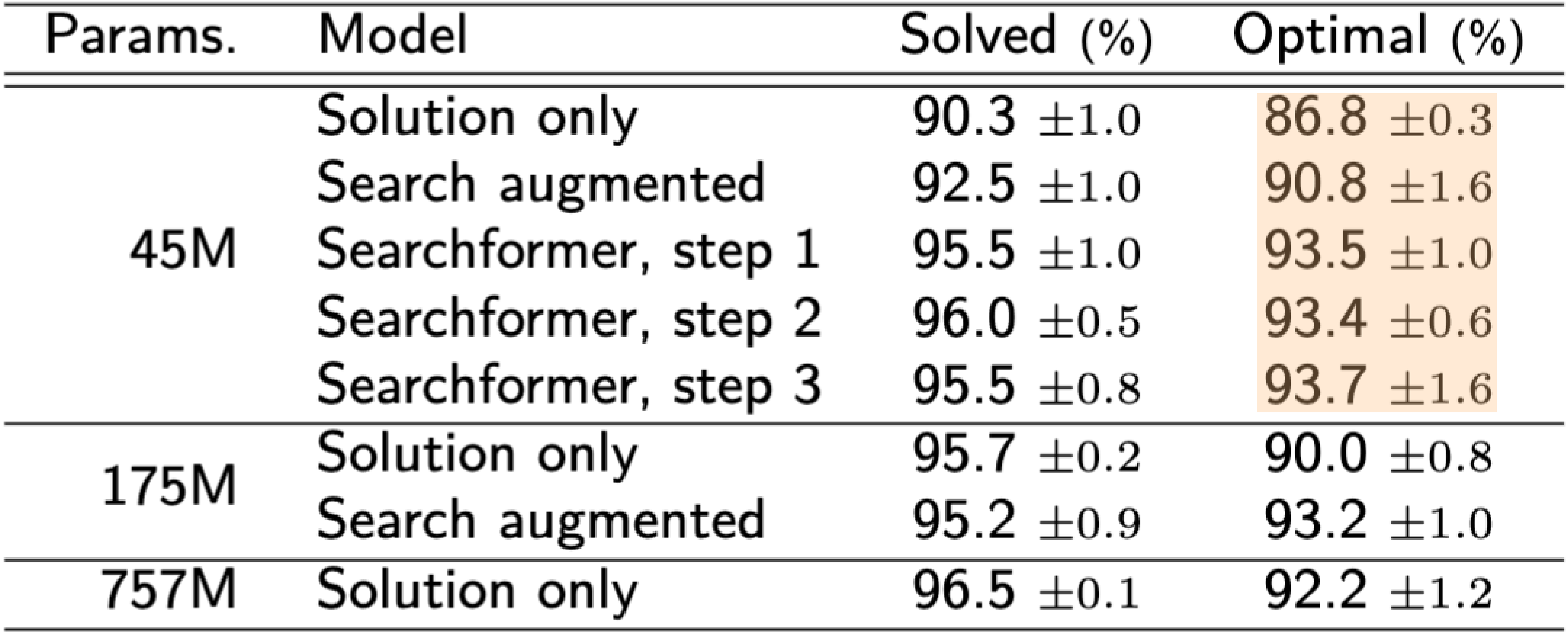
Improving search dynamics via bootstrapping



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
	Searchformer, step 3	1.292 ±0.044	1.343 ±0.067
175M	Solution only	–	–
	Search augmented	0.925 ±0.010	0.933 ±0.011
757M	Solution only	–	–

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

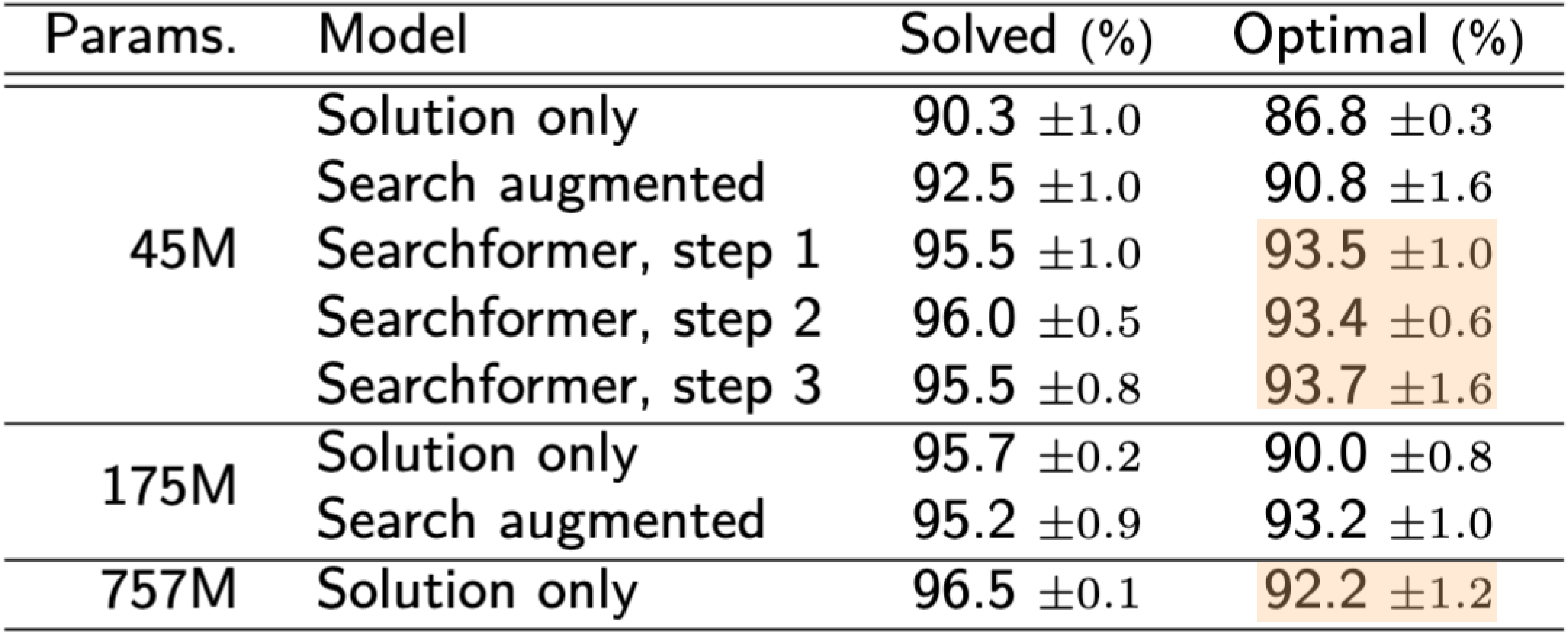
Improving search dynamics via bootstrapping



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.

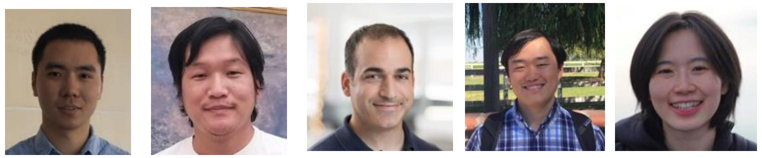
Improving search dynamics via bootstrapping

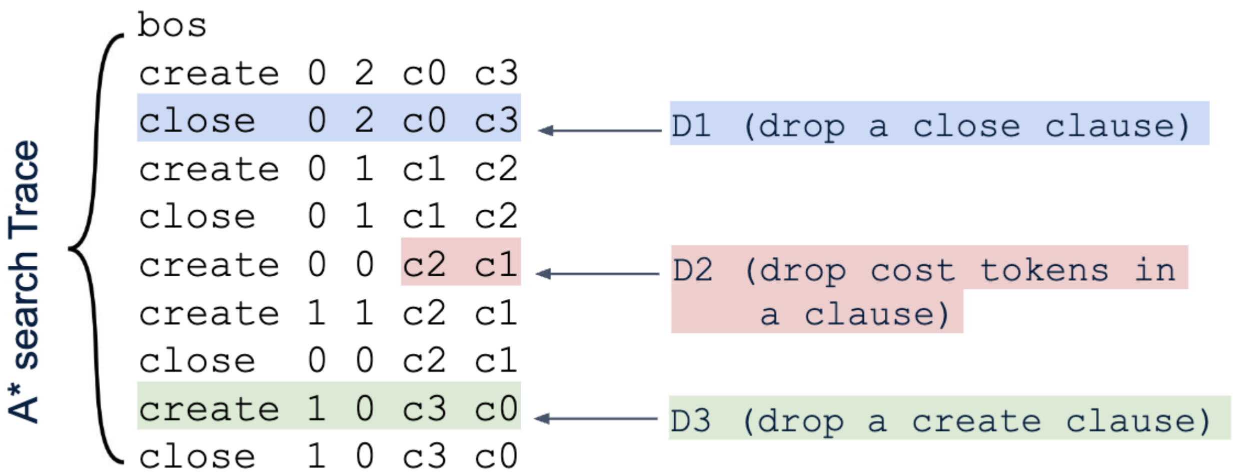


Params.	Model	Solved (%)	Optimal (%)
45M	Solution only	90.3 ±1.0	86.8 ±0.3
	Search augmented	92.5 ±1.0	90.8 ±1.6
	Searchformer, step 1	95.5 ±1.0	93.5 ±1.0
	Searchformer, step 2	96.0 ±0.5	93.4 ±0.6
	Searchformer, step 3	95.5 ±0.8	93.7 ±1.6
175M	Solution only	95.7 ±0.2	90.0 ±0.8
	Search augmented	95.2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

Searchformer outperforms largest solution-only model.

DualFormer (Searchformer v2)





A* search Trace

```
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
```

D1 (drop a close clause)

D2 (drop cost tokens in a clause)

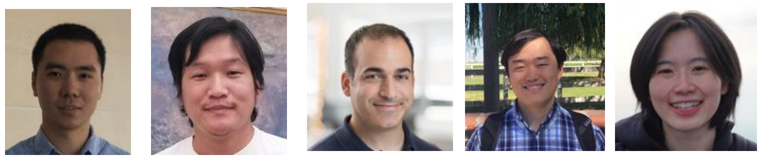
D3 (drop a create clause)

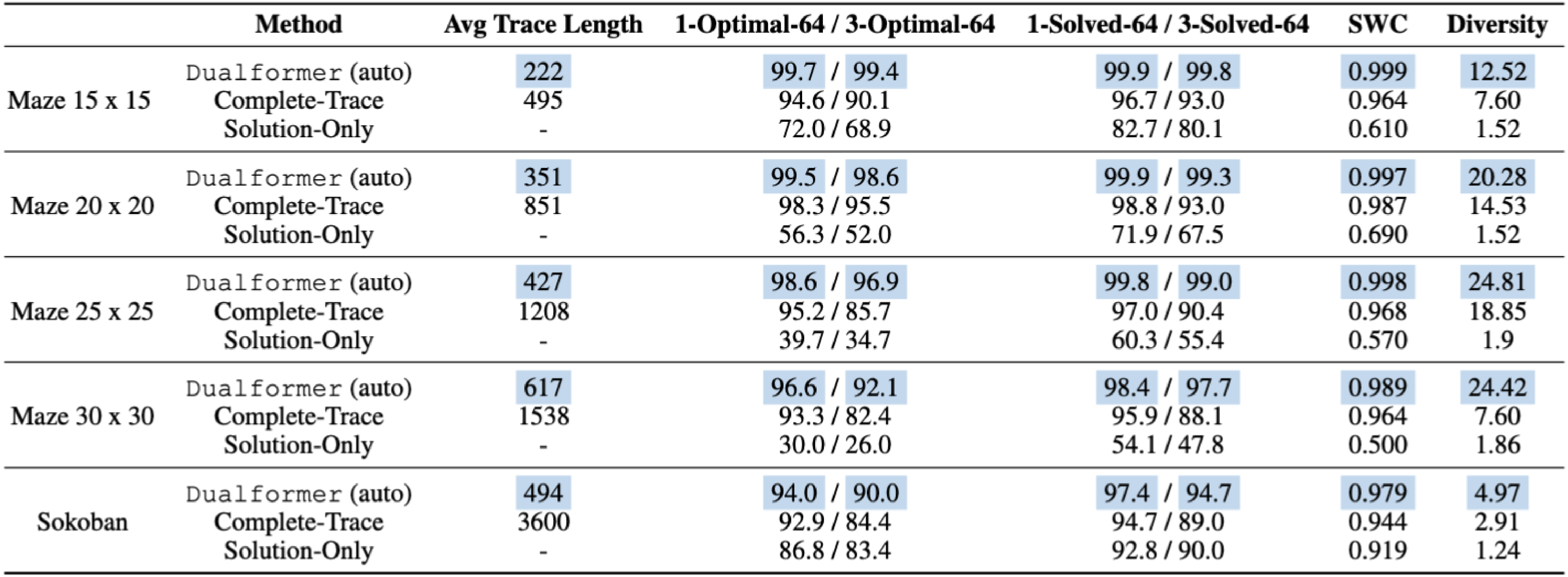
Structured Trace Dropping Strategies

```
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
```

[D. Su et al, *Dualformer: Controllable Fast and Slow Thinking by Learning with Randomized Reasoning Traces*, arXiv'24]

DualFormer (Searchformer v2)

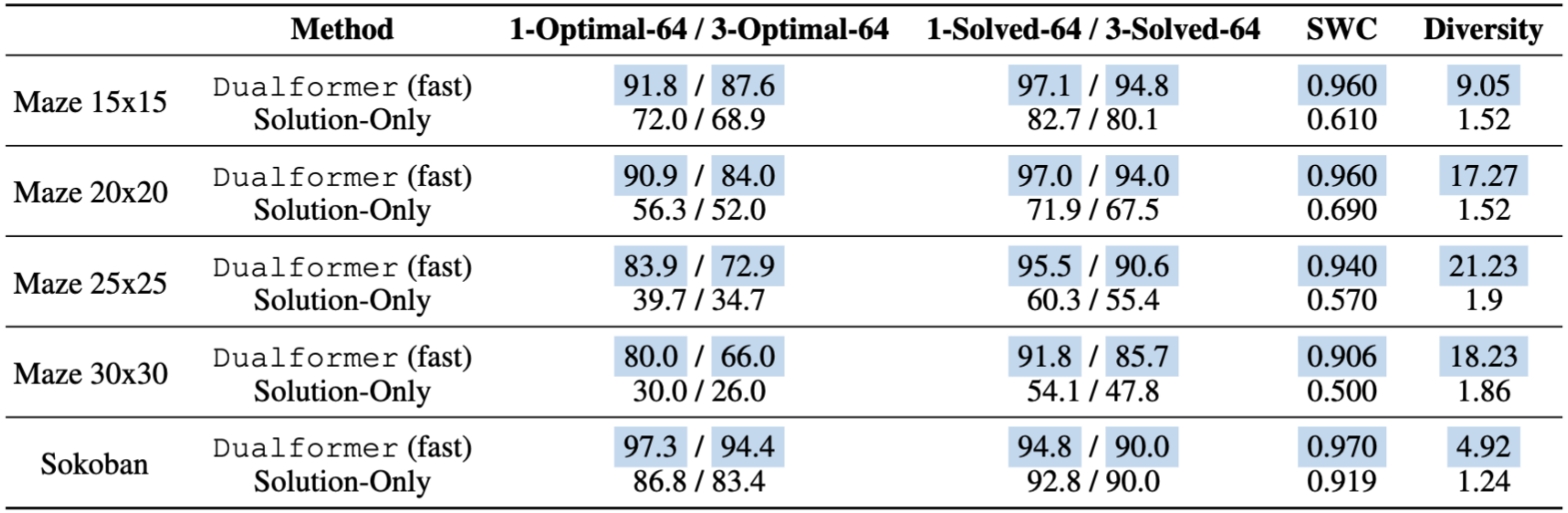




	Method	Avg Trace Length	1-Optimal-64 / 3-Optimal-64	1-Solved-64 / 3-Solved-64	SWC	Diversity
Maze 15 x 15	`Dualformer` (auto)	222	99.7 / 99.4	99.9 / 99.8	0.999	12.52
	Complete-Trace	495	94.6 / 90.1	96.7 / 93.0	0.964	7.60
	Solution-Only	-	72.0 / 68.9	82.7 / 80.1	0.610	1.52
Maze 20 x 20	`Dualformer` (auto)	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

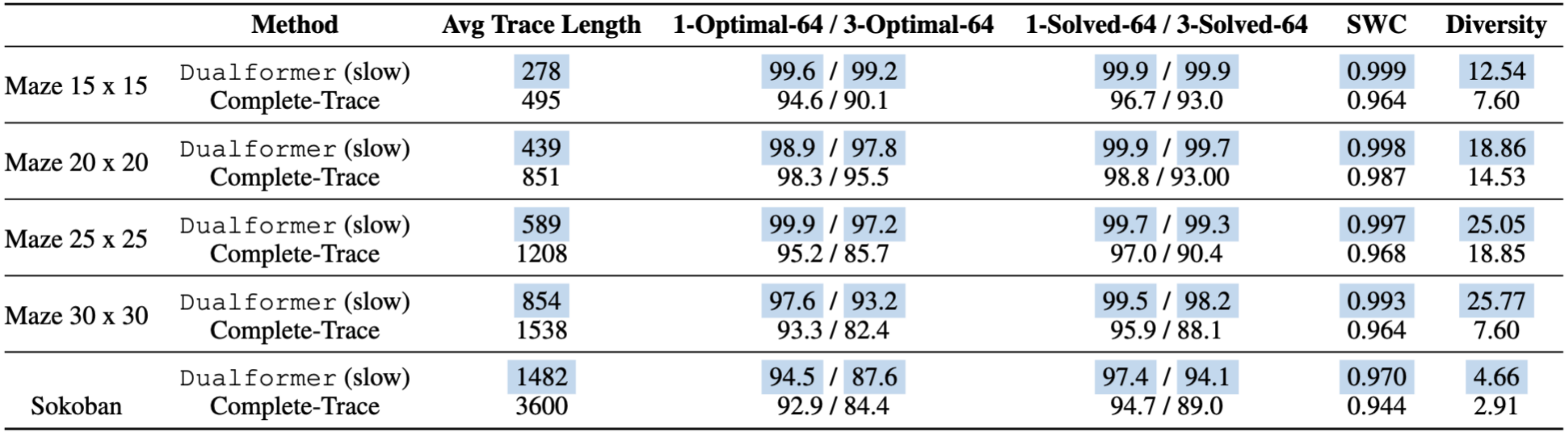
Dualformer **automatically** switches between fast mode (System 1) and slow mode (System 2) and works **better** for **dedicated** models on either modes.

Fast mode performance



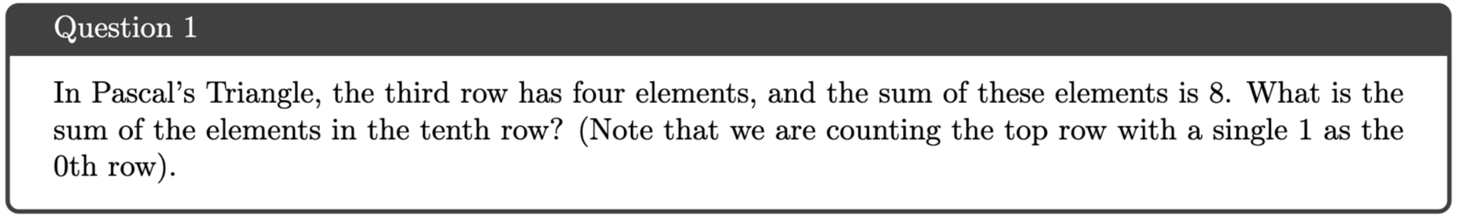
	Method	1-Optimal-64 / 3-Optimal-64	1-Solved-64 / 3-Solved-64	SWC	Diversity
Maze 15x15	Dualformer (fast)	91.8 / 87.6	97.1 / 94.8	0.960	9.05
	Solution-Only	72.0 / 68.9	82.7 / 80.1	0.610	1.52
Maze 20x20	Dualformer (fast)	90.9 / 84.0	97.0 / 94.0	0.960	17.27
	Solution-Only	56.3 / 52.0	71.9 / 67.5	0.690	1.52
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



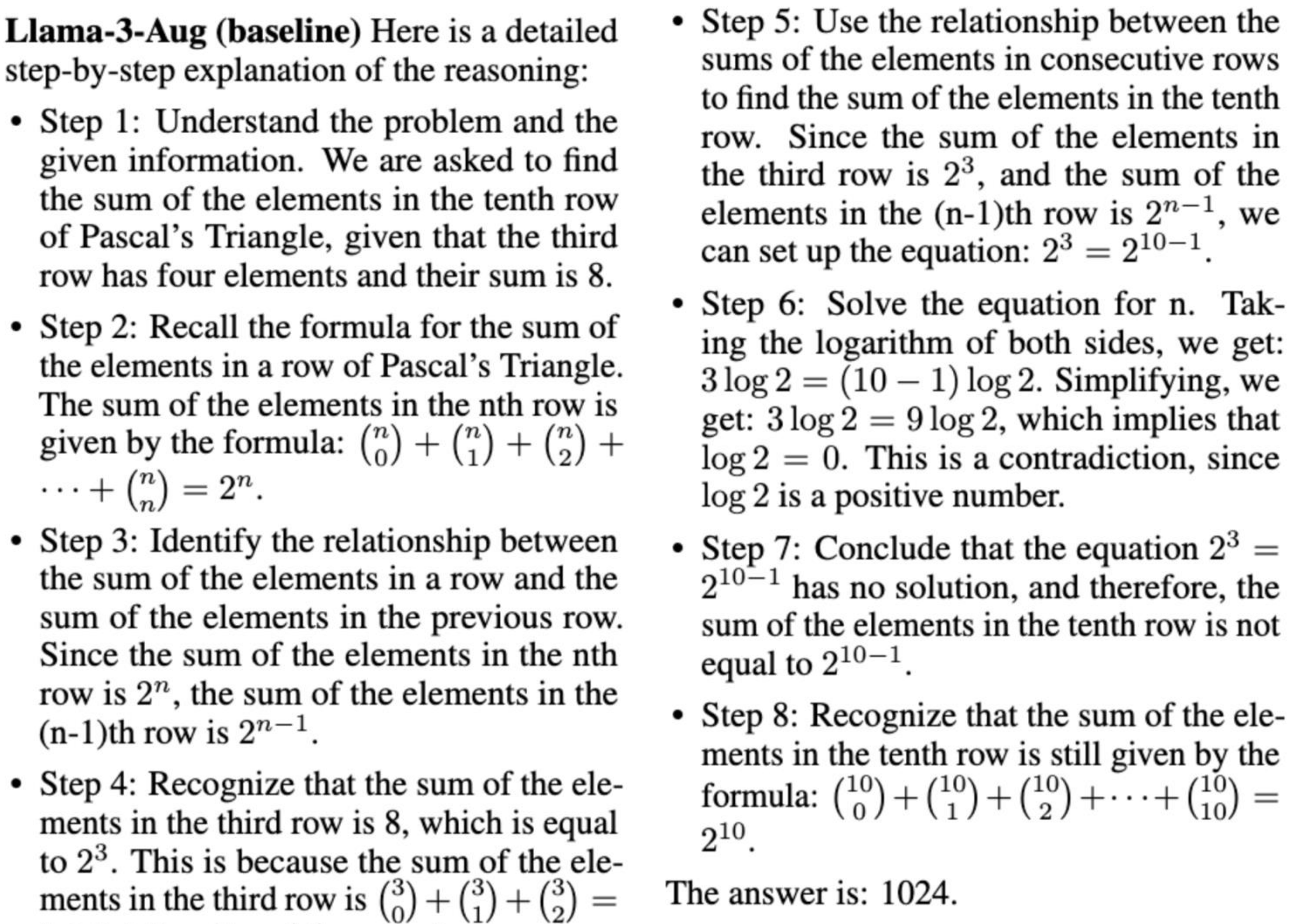
	Method	Avg Trace Length	1-Optimal-64 / 3-Optimal-64	1-Solved-64 / 3-Solved-64	SWC	Diversity
Maze 15 x 15	Dualformer (slow)	278	99.6 / 99.2	99.9 / 99.9	0.999	12.54
	Complete-Trace	495	94.6 / 90.1	96.7 / 93.0	0.964	7.60
Maze 20 x 20	Dualformer (slow)	439	98.9 / 97.8	99.9 / 99.7	0.998	18.86
	Complete-Trace	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



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).

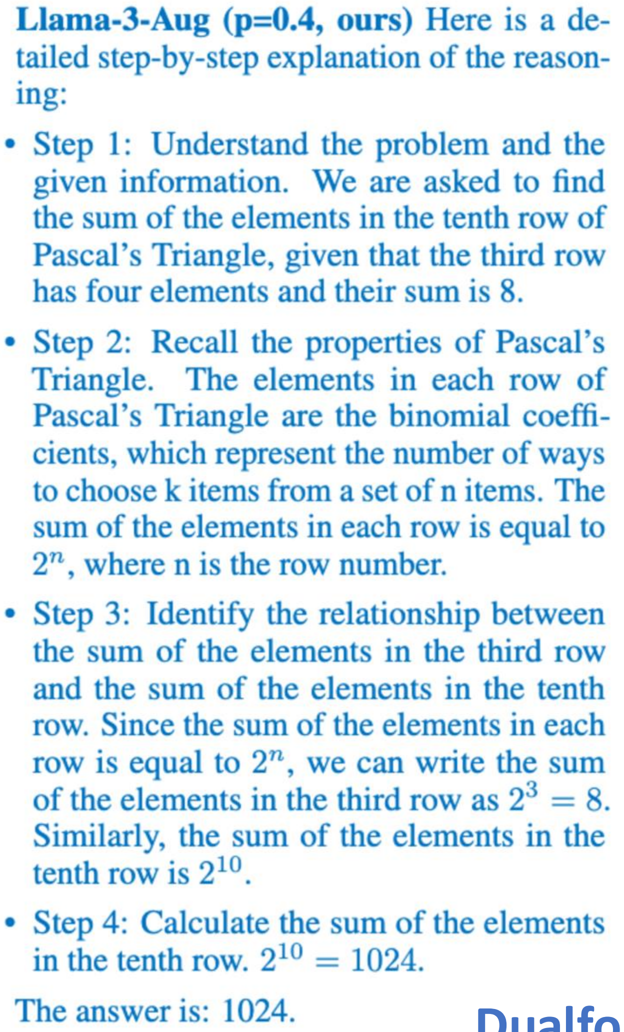


Llama-3-Aug (baseline) Here is a detailed step-by-step explanation of the reasoning:

- 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: $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n$.
- 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.
- 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: $\binom{10}{0} + \binom{10}{1} + \binom{10}{2} + \cdots + \binom{10}{10} = 2^{10}$.

The answer is: 1024.

Baseline



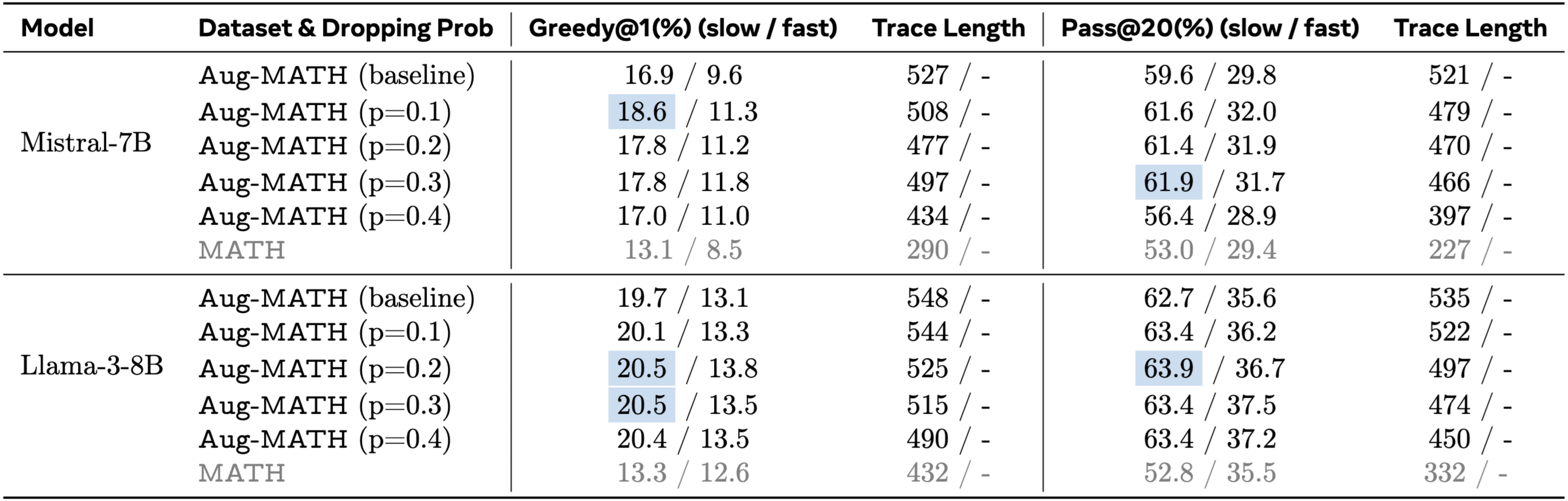
Llama-3-Aug (p=0.4, ours) Here is a detailed step-by-step explanation of the reasoning:

- 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.

Dualformer

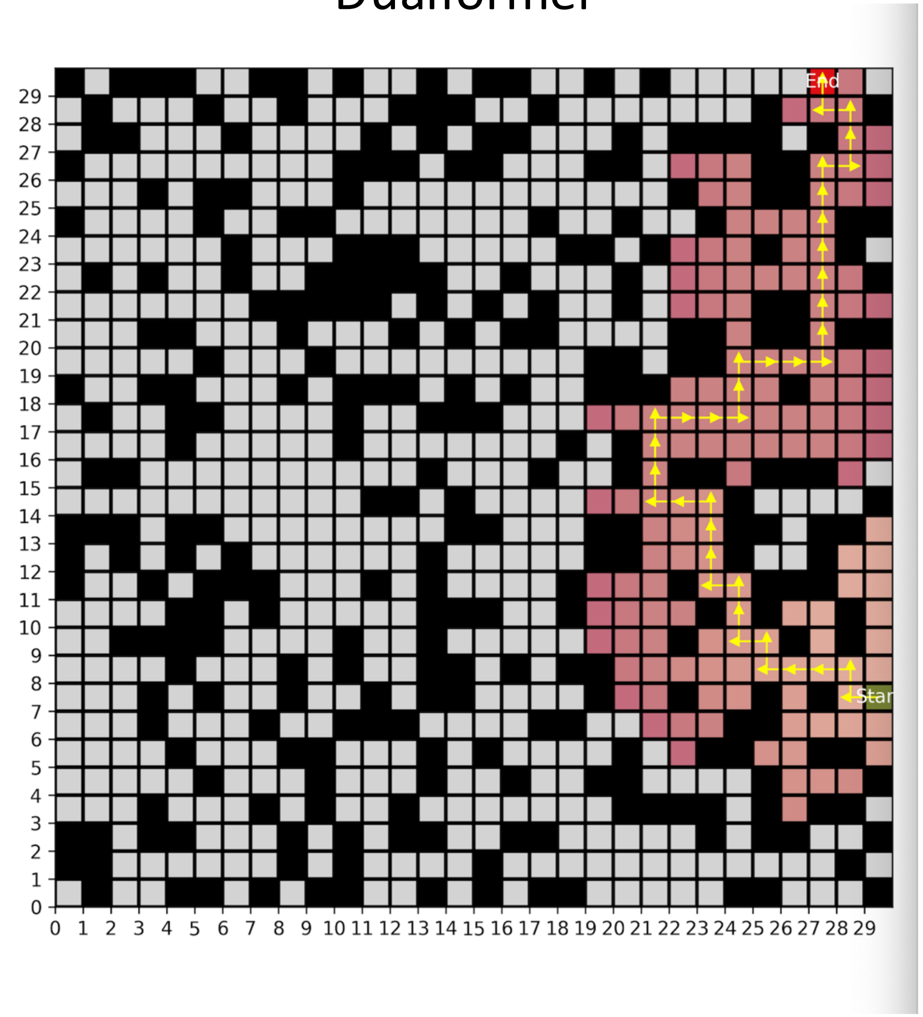
Math Problems



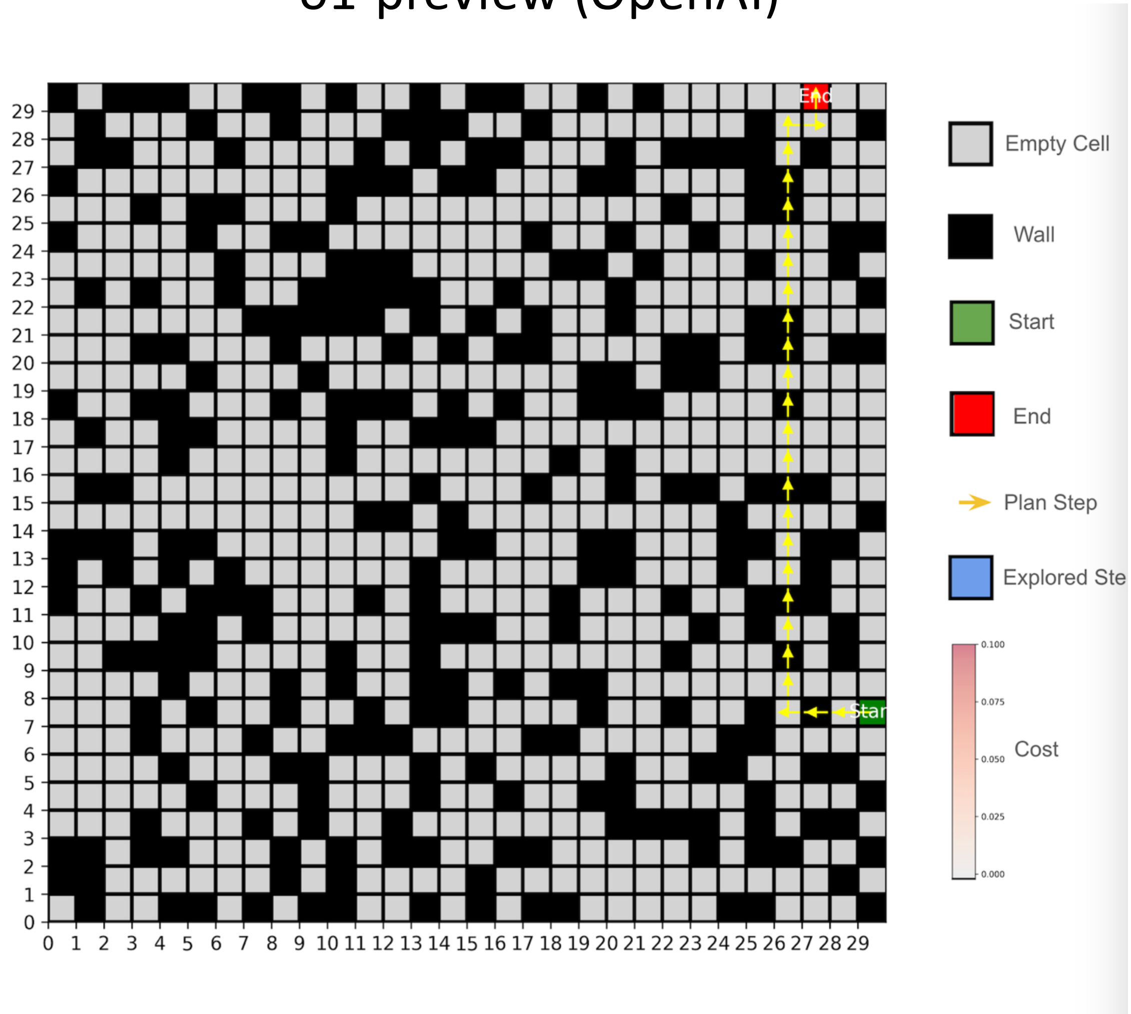
Model	Dataset & Dropping Prob	Greedy@1(%) (slow / fast)	Trace Length	Pass@20(%) (slow / fast)	Trace Length
Mistral-7B	Aug-MATH (baseline)	16.9 / 9.6	527 / -	59.6 / 29.8	521 / -
	Aug-MATH (p=0.1)	18.6 / 11.3	508 / -	61.6 / 32.0	479 / -
	Aug-MATH (p=0.2)	17.8 / 11.2	477 / -	61.4 / 31.9	470 / -
	Aug-MATH (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

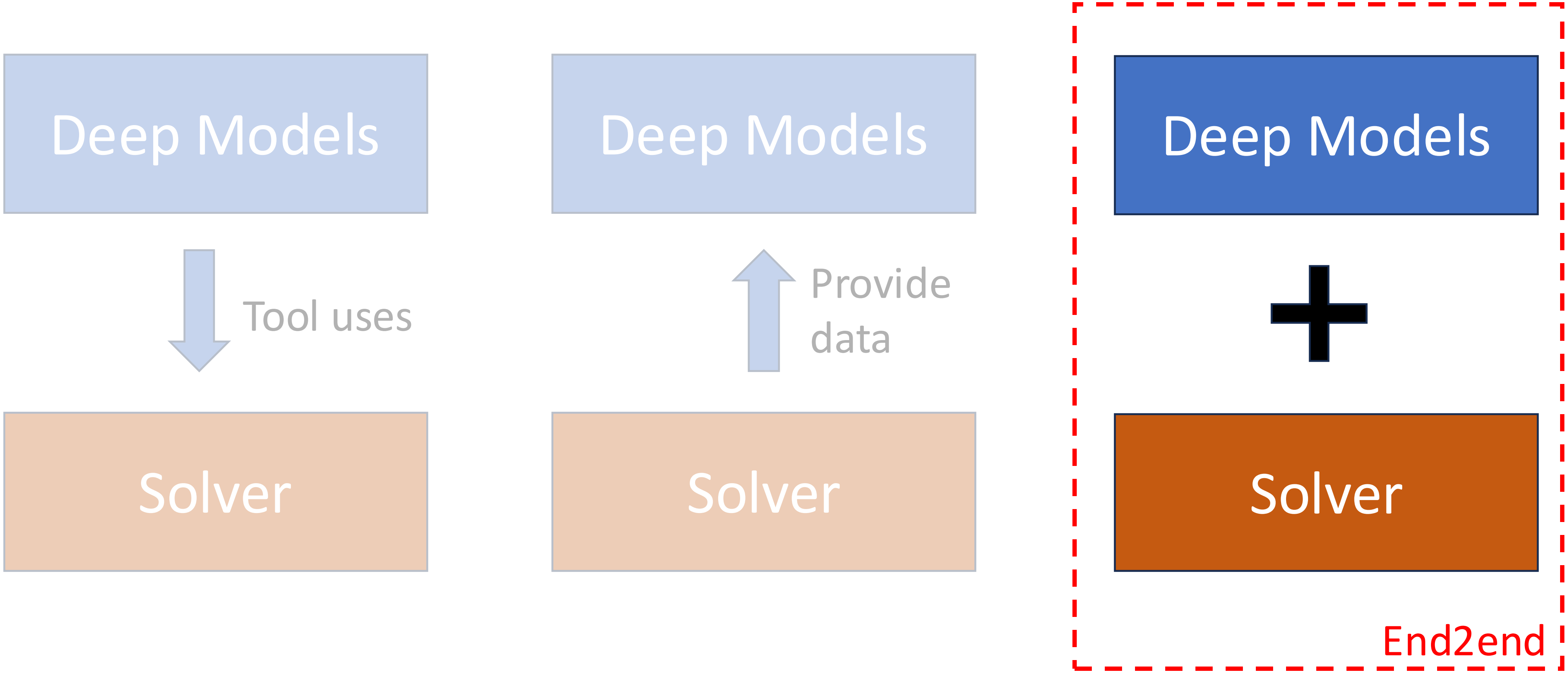
Dualformer



o1-preview (OpenAI)



Option **Two**: Hybrid Systems



Deep Models

Tool uses

Solver

Deep Models

Provide data

Solver

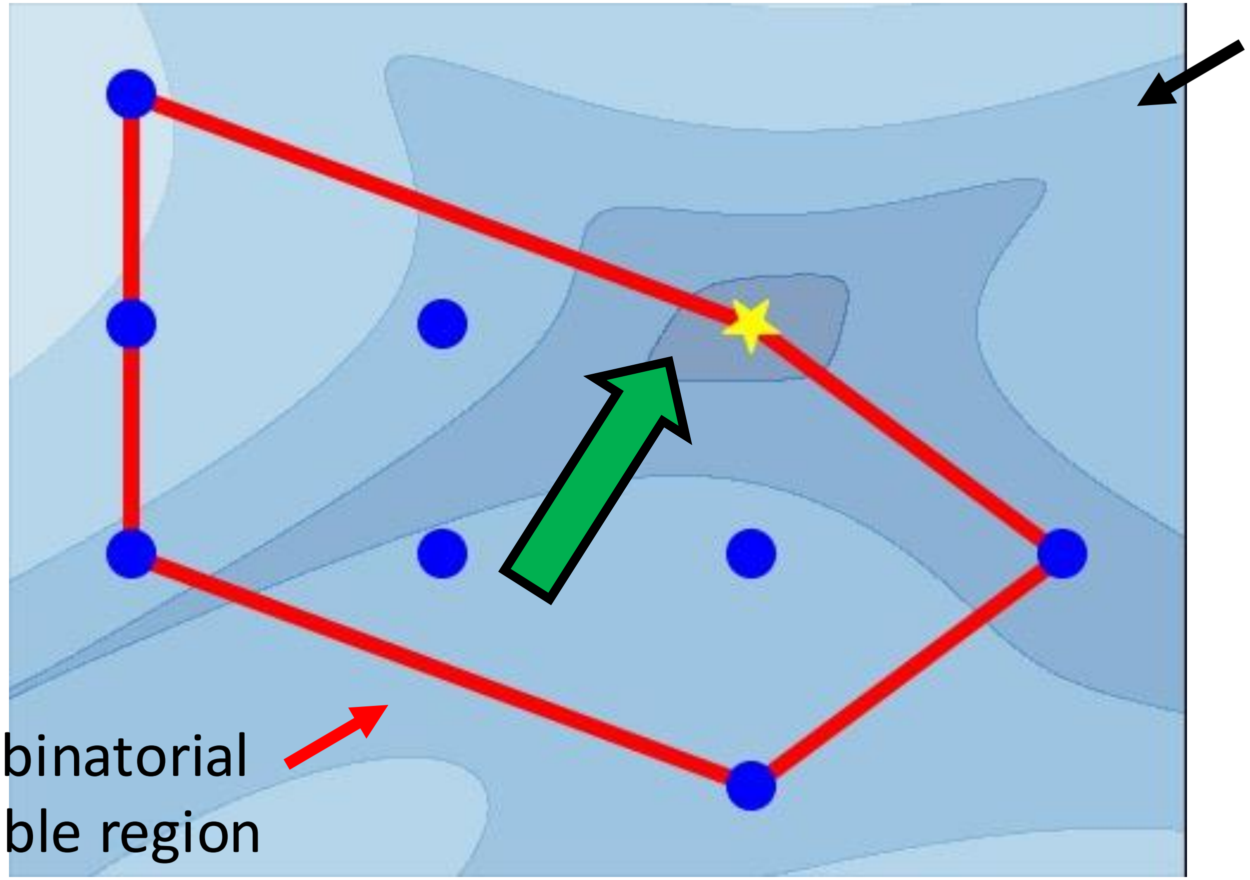
Deep Models

+

Solver

End2end

Nonlinear objective with combinatorial constraints

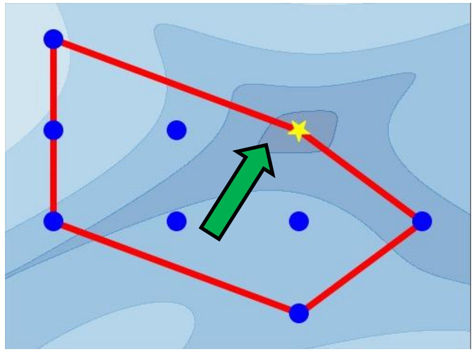


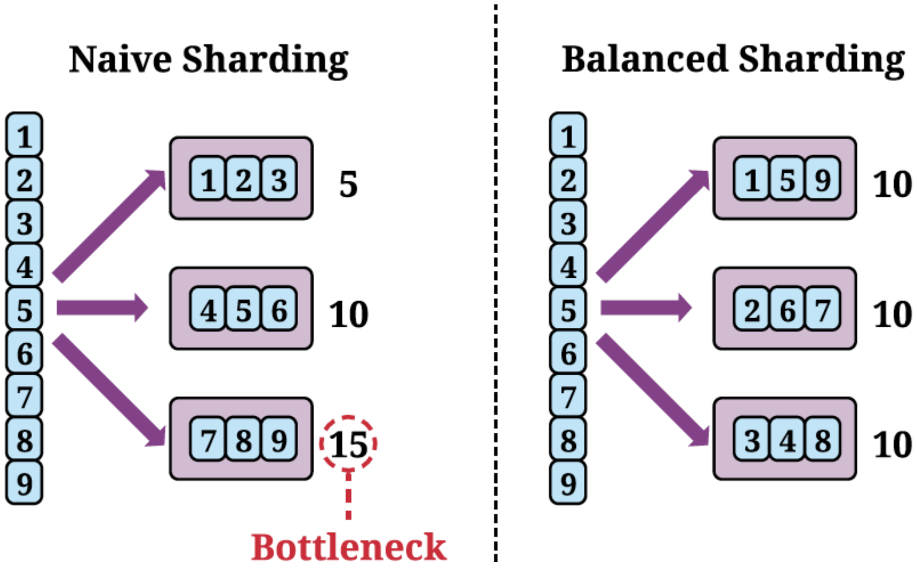
Nonlinear + differentiable objective $f(\boldsymbol{x})$

Combinatorial feasible region

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

Example: Embedding Table Placement





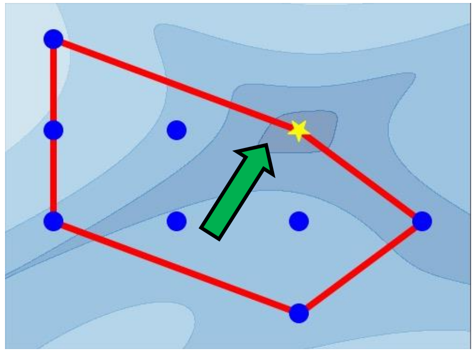
Given:

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

Find

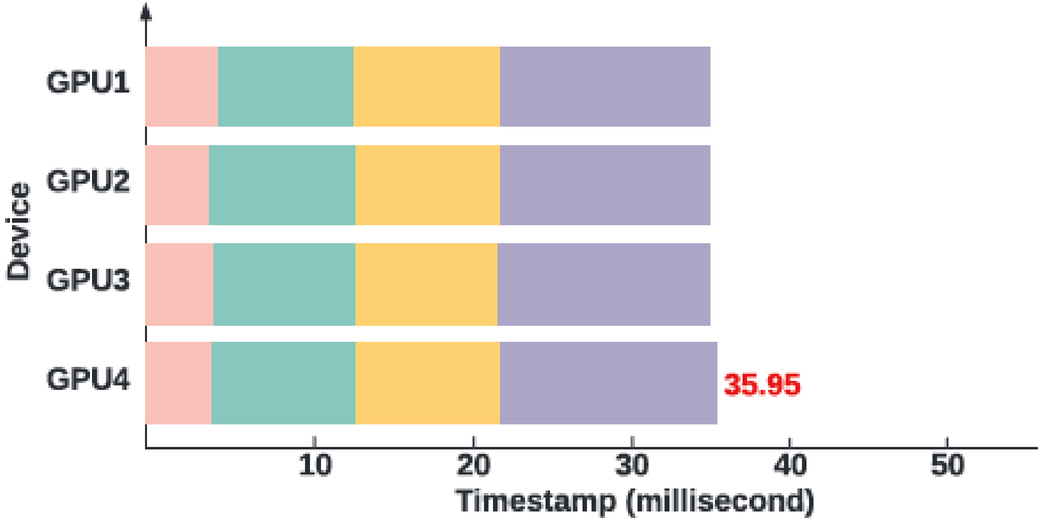
- Allocation of tables to devices observing device memory limits
- Minimize latency which is estimated by a neural network (capturing nonlinear interactions)

Example: Embedding Table Placement



Given:

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

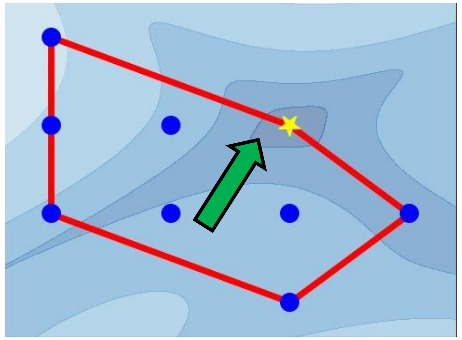


Formulation

$$\mathrm{Min}_x\, \boldsymbol{L}(\{x_{ij}\}) \quad \text{s.t. } \textstyle\sum_i x_{ij} m_i \le M_j, \quad \sum_j x_{ij} = 1\,, \quad x_{ij} \in \{0,1\}$$

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

Solve the Combinatorial Problem in the Latent Space

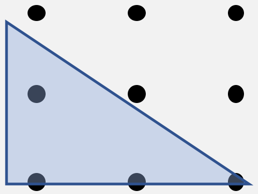


Original Space

Nonlinear optimization with combinatorial constraints

$$\min_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{y})$$

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



combinatorial constraints

Predict surrogate cost $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{y})$

Latent Space

Surrogate optimization

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

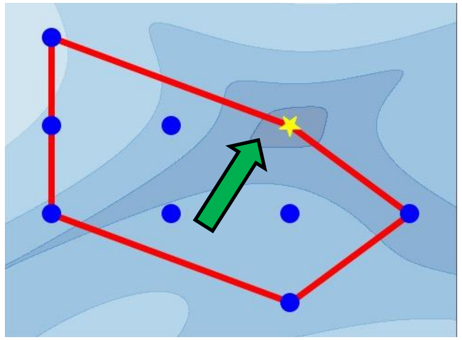
$$\text{s.t }\ \boldsymbol{x} \in \Omega$$

solved by existing combinatorial solvers

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

[A. Ferber et al, *SurCo: Learning Linear Surrogates For Combinatorial Nonlinear Optimization Problems*, ICML'23 and **outstanding paper** in SODS workshop]

Solve the Combinatorial Problem in the Latent Space

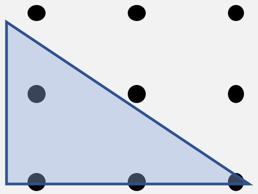


Original Space

Nonlinear optimization with combinatorial constraints

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{y})$$

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



combinatorial constraints

Predict surrogate cost $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{y})$

Latent Space

Surrogate optimization

$$\boldsymbol{x}^*(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \boldsymbol{c}(\boldsymbol{y})^T \boldsymbol{x}$$

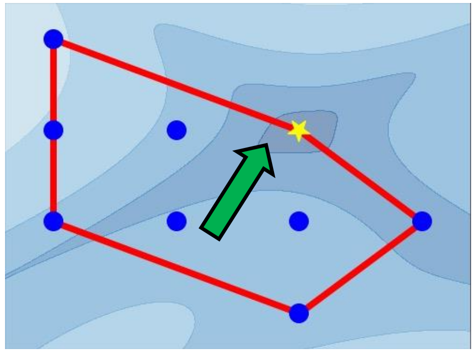
$$\text{s.t } \boldsymbol{x} \in \Omega$$

solved by existing combinatorial solvers

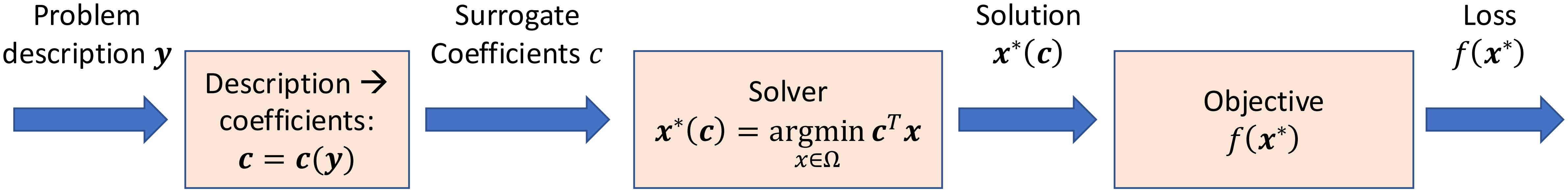
$\boldsymbol{x}^*(\boldsymbol{y})$ optimizes $f(\boldsymbol{x}; \boldsymbol{y})$ as much as possible

Proposal: gradient-based optimization

SurCo: Surrogate combinatorial opt



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



Problem description $\boldsymbol{y}$

Description → coefficients: $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{y})$

Surrogate Coefficients c

Solver $\boldsymbol{x}^*(\boldsymbol{c}) = \underset{x\in\Omega}{\operatorname{argmin}}\, \boldsymbol{c}^T\boldsymbol{x}$

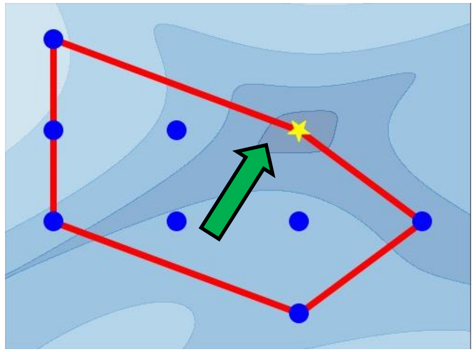
Solution $\boldsymbol{x}^*(\boldsymbol{c})$

Objective $f(\boldsymbol{x}^*)$

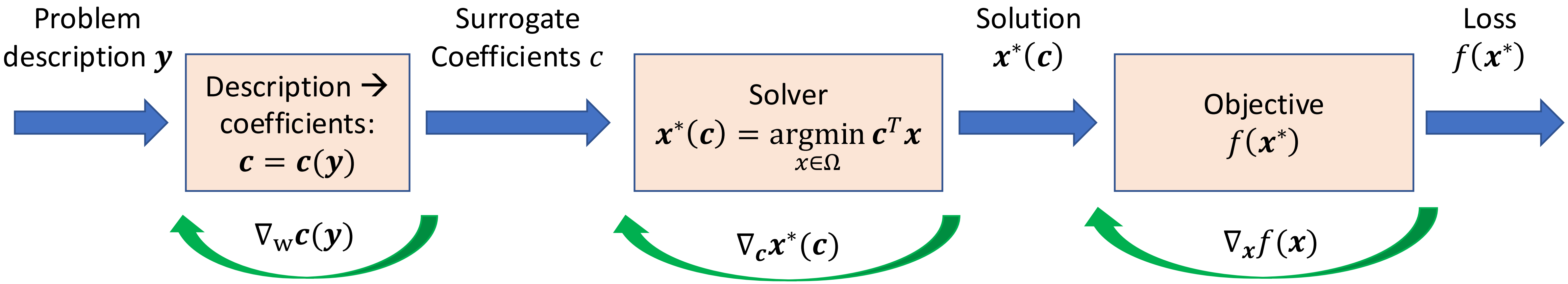
Loss $f(\boldsymbol{x}^*)$

[A. Ferber et al, *SurCo: Learning Linear Surrogates For Combinatorial Nonlinear Optimization Problems*, ICML'23 and **outstanding paper** in SODS workshop]

Gradient-based Optimization



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



Problem description $\boldsymbol{y}$

Description → coefficients: $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{y})$

Surrogate Coefficients c

Solver $\boldsymbol{x}^*(\boldsymbol{c}) = \underset{x\in\Omega}{\operatorname{argmin}}\, \boldsymbol{c}^T \boldsymbol{x}$

Solution $\boldsymbol{x}^*(\boldsymbol{c})$

Objective $f(\boldsymbol{x}^*)$

Loss $f(\boldsymbol{x}^*)$

$\nabla_{\mathrm{W}} \boldsymbol{c}(\boldsymbol{y})$

$\nabla_{\boldsymbol{c}} \boldsymbol{x}^*(\boldsymbol{c})$

$\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$

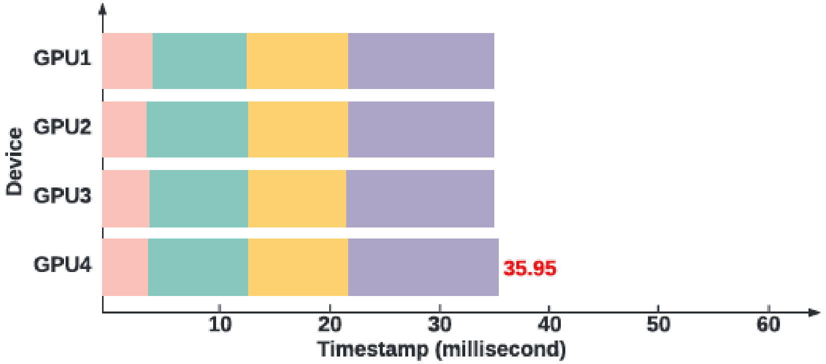
Assumed differentiable

Recent work on differentiable optimization

Differentation of blackbox optimizers
CVXPYLayers
MIPaaL
Etc.

Assumed differentiable

Embedding Table Sharding



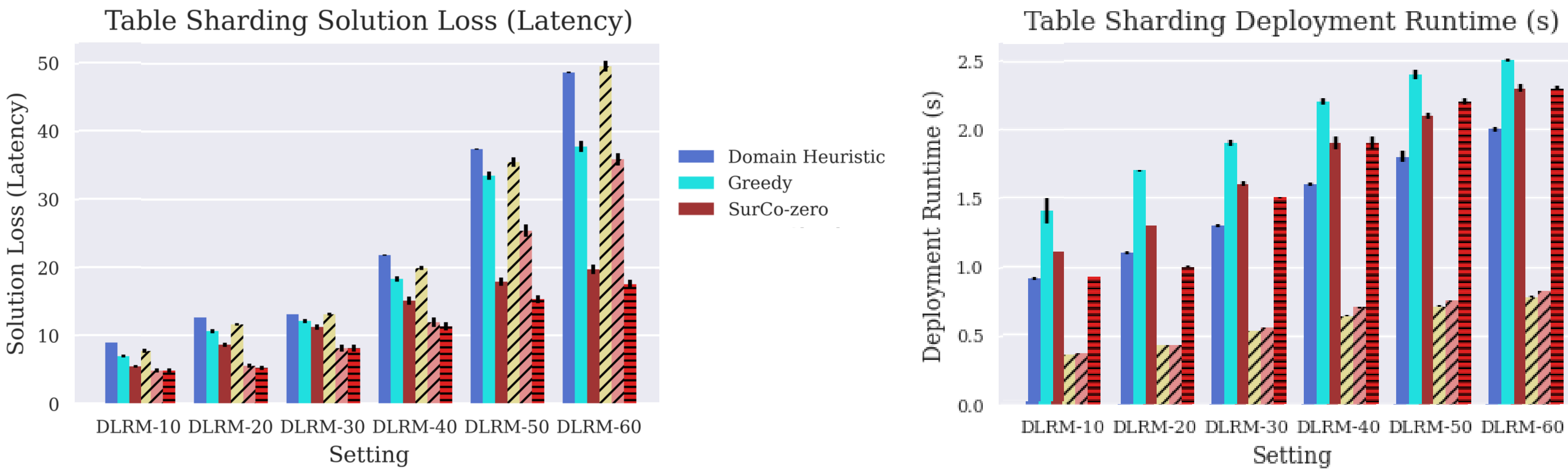
- Public **D**eep **L**earning **R**ecommendation **M**odel (DLRM dataset) placing between 10 to 60 tables on 4 GPUs

- Baseline: Greedy
- SoTA: RL approach Dreamshard[1]
- SurCo: Surrogate NN model learned via CVXPYLayers (differentiable LP Solver)

[1] Zha et al. NeurIPS 2022

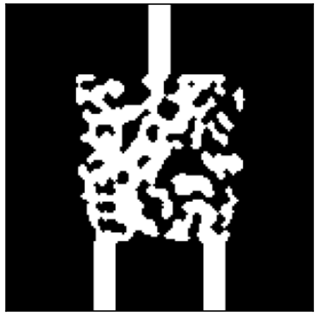
Dataset: https://github.com/facebookresearch/dlrm_datasets

Results – Table Sharding

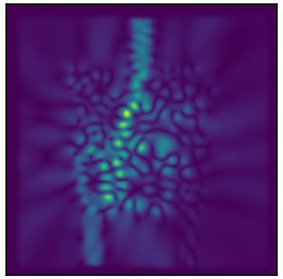


Inverse Photonic Design

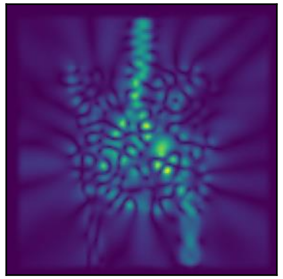
Device Design



E_z magnitude first wavelength



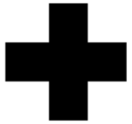
E_z magnitude second wavelength

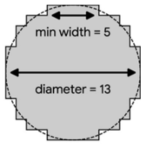


- Design physically-viable devices that take light waves and routes different wavelengths to correct locations

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

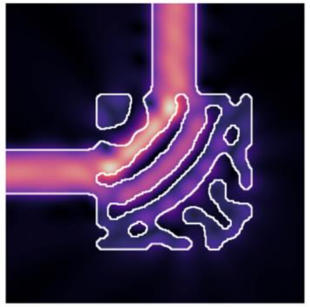
- Device design misspecification loss $f(\boldsymbol{x})$ computed by differentiable electromagnetic simulator
- Feasible solution: the design must be the union of brush pattern
  - x = binary_opening(x, brush)
  - x = ~binary_opening(~x, brush)

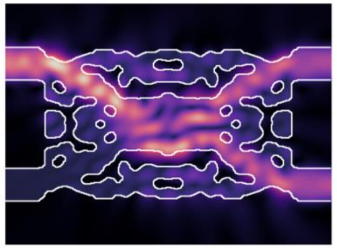


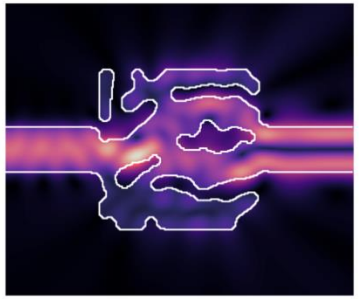


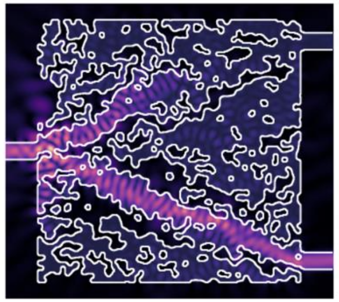
Inverse Photonic Design

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


Waveguide bend

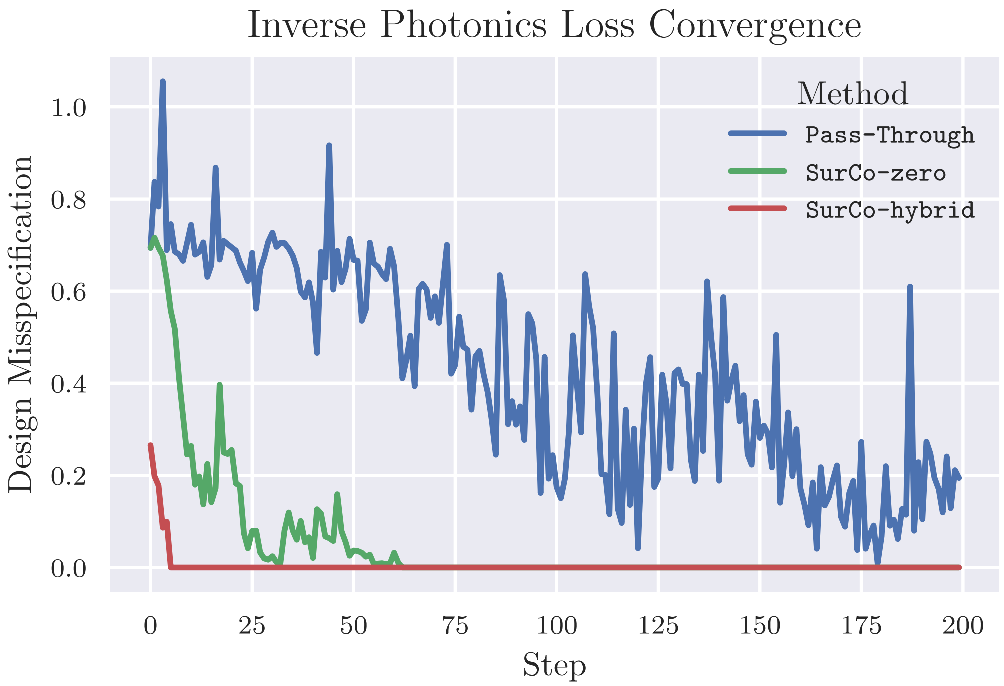

Beam splitter


Mode converter


Wavelength division multiplexer

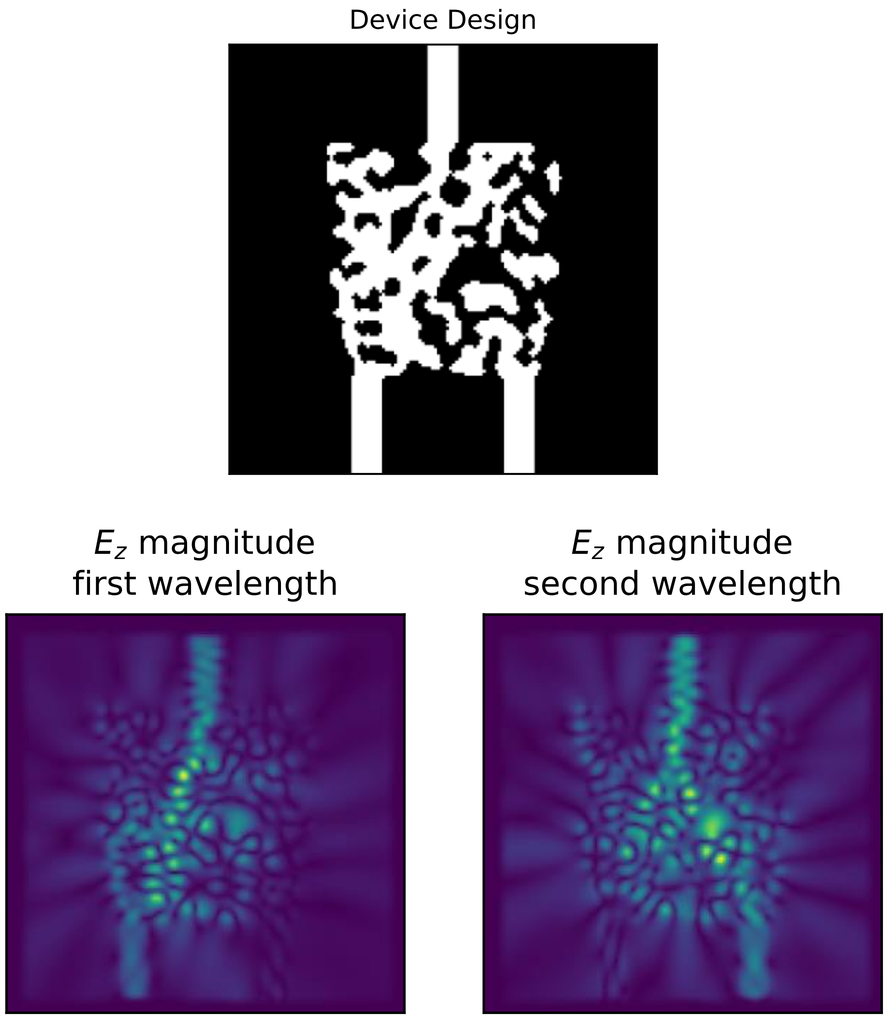
[1] Schubert et al. ACS Photonics 2022
[2] Vlastelica et al. ICLR 2019
Dataset: https://github.com/google/ceviche-challenges

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

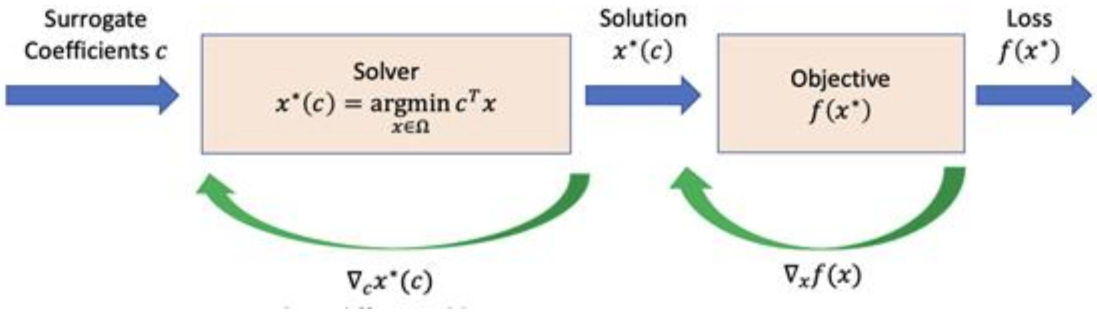
Wavelength division multiplexer

Limitation of SurCo

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

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

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



- Requires $\nabla_x f(x)$ →Does not apply to nondifferentiable functions
- Requires $\nabla_c \boldsymbol{g}_{\boldsymbol{\theta}}(c)$ →Solver is backpropagatable

[A. Zharmagambetov et al, *Landscape Surrogate: Learning Decision Losses for Mathematical Optimization Under Partial Information*, NeurIPS'23]

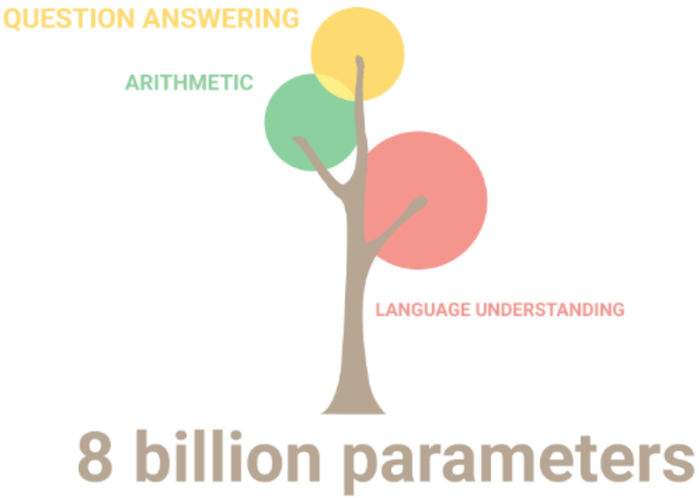
[A. Ferber et al, GenCO: Generating Diverse Solutions to Design Problems with Combinatorial Nature, ICML'24]

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

Emerging Symbolic
Structure

Deep Models

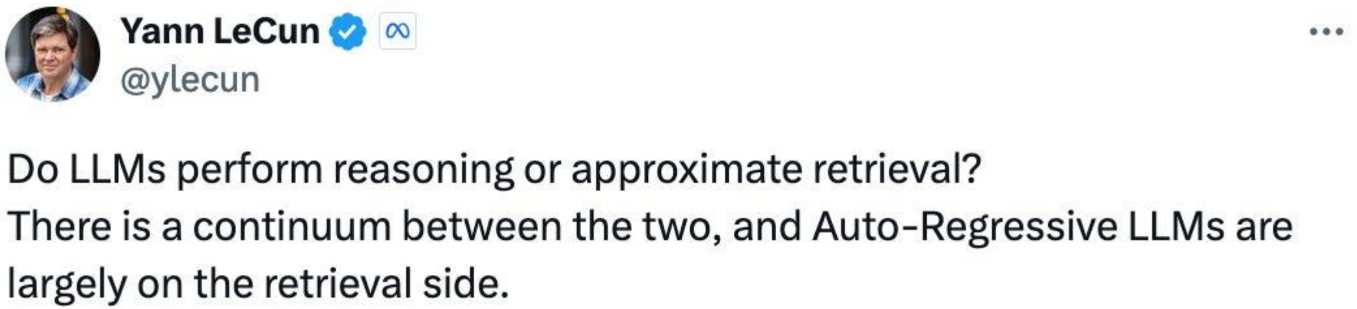
Debate: Is LLM doing retrieval or true reasoning?



LLM shows emergent behaviors!!

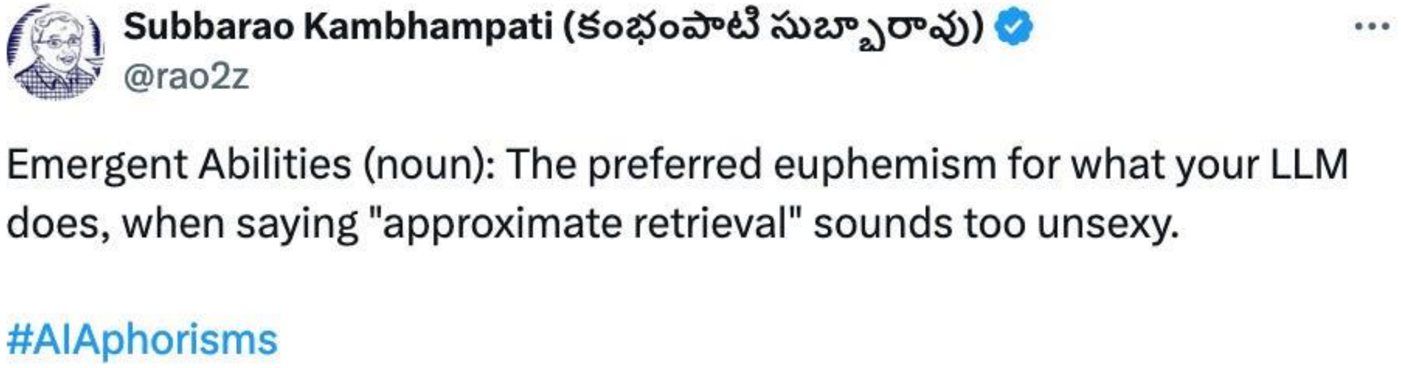
https://medium.com/@fenjiro/large-language-models-llms-emergent-abilities-chatgpt-talks-moroccan-dialect-as-an-example-c945f93aa63a

Debate: Is LLM doing retrieval or true reasoning?



Yann LeCun
@ylecun

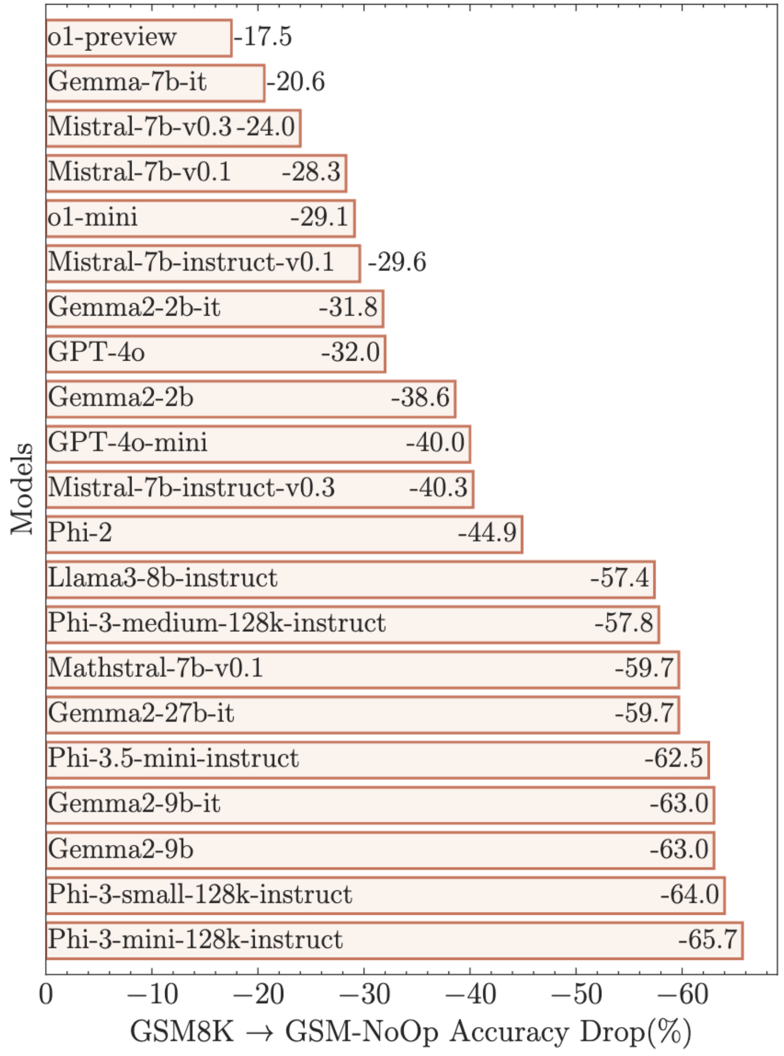
Do LLMs perform reasoning or approximate retrieval?
There is a continuum between the two, and Auto-Regressive LLMs are largely on the retrieval side.



Subbarao Kambhampati (కంభంపాటి సుబ్బారావు)
@rao2z

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

#AIAphorisms



LLM is just doing retrievals!!

Concrete Example: **Modular Addition**

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

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

[T. Zhou et al, Pre-trained Large Language Models Use Fourier Features to Compute Addition]

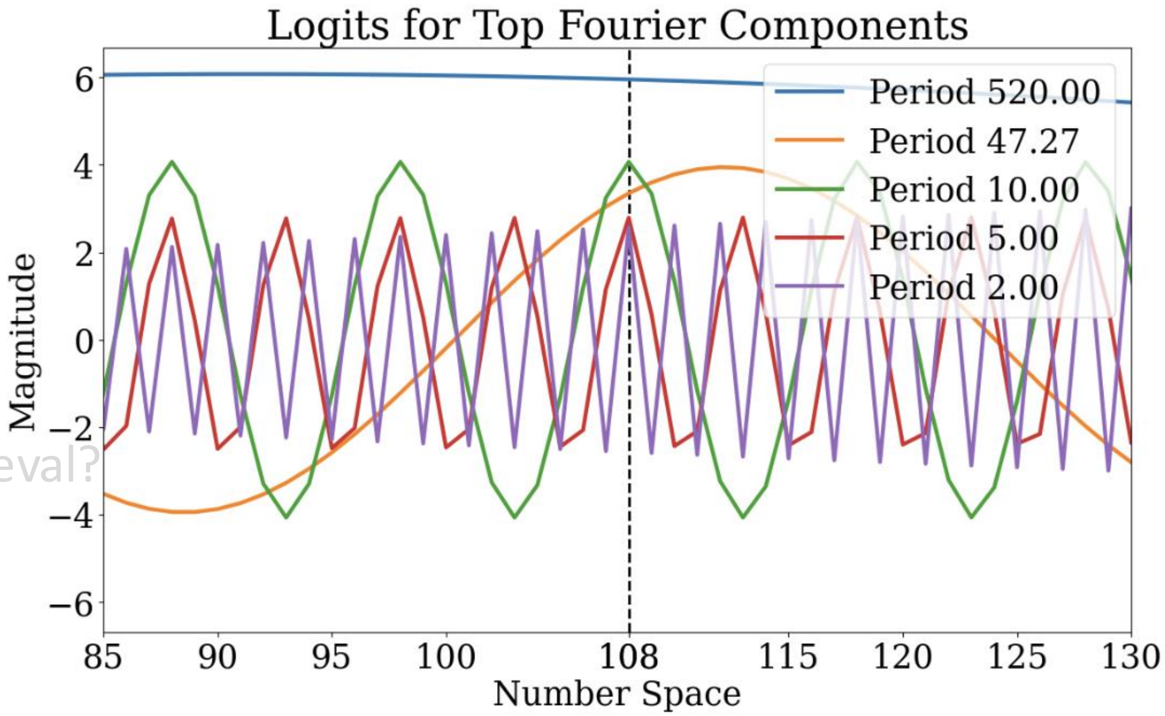
Concrete Example: **Modular Addition**

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

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

Learned representation = Fourier basis 🤯

Why? 🤔

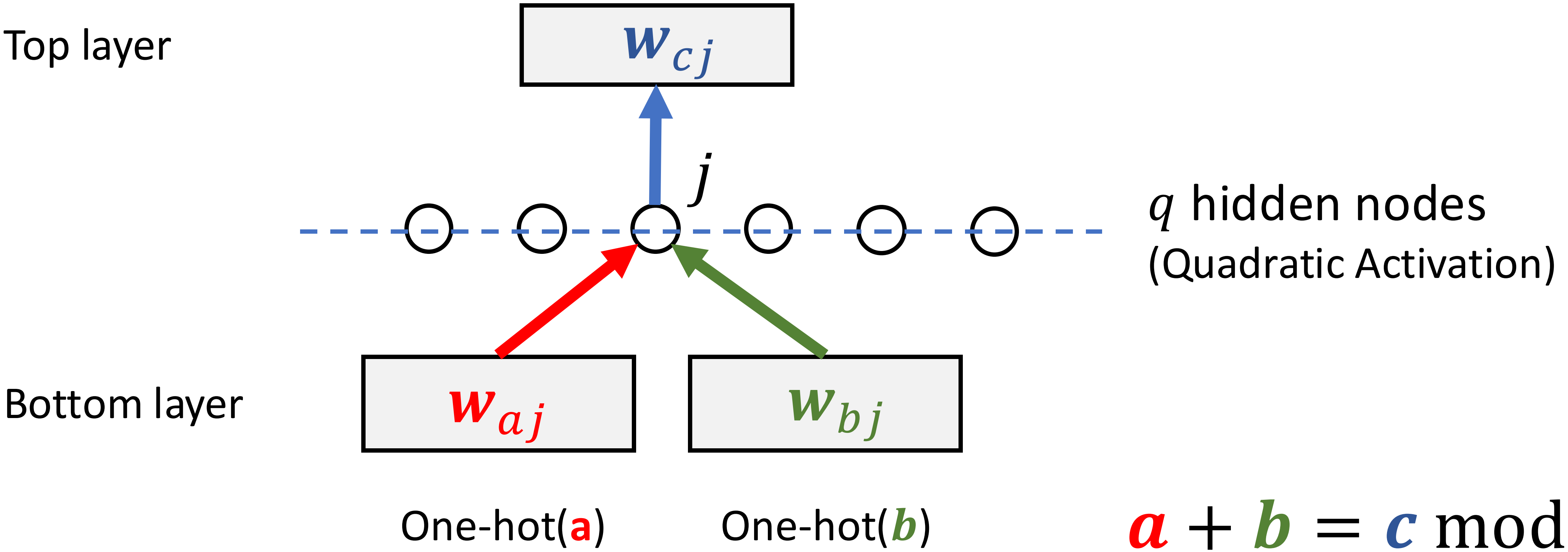


(a) Final logits for top Fourier components

[T. Zhou et al, Pre-trained Large Language Models Use Fourier Features to Compute Addition]

Problem Setup

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



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

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

(Scaled) Fourier Transform

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

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

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

k: frequency

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

↓

Hermitian condition holds

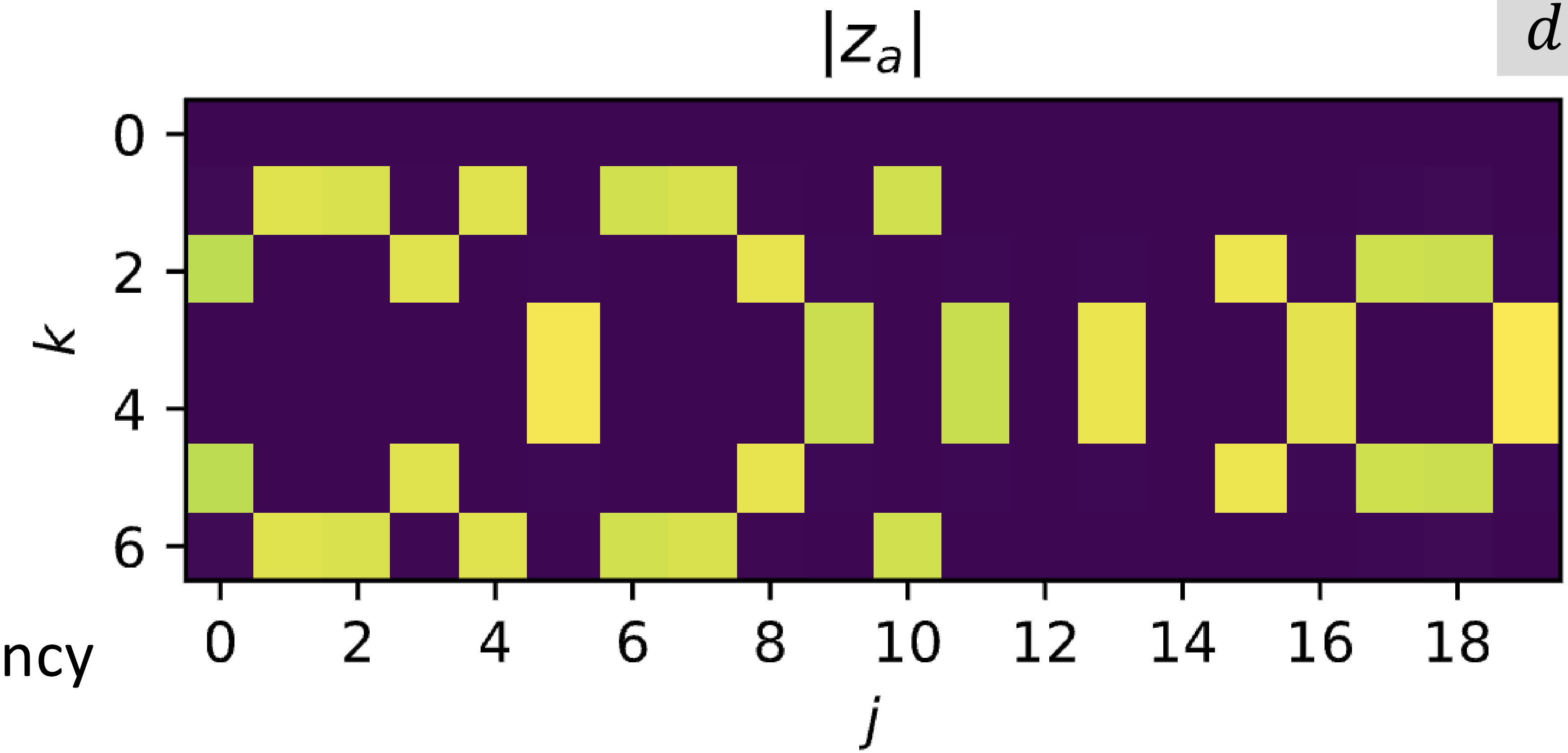
$$z_{akj} = \overline{z_{a,-k,j}}$$

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

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

What a Gradient Descent Solution look like?

$d = 7, q = 20$

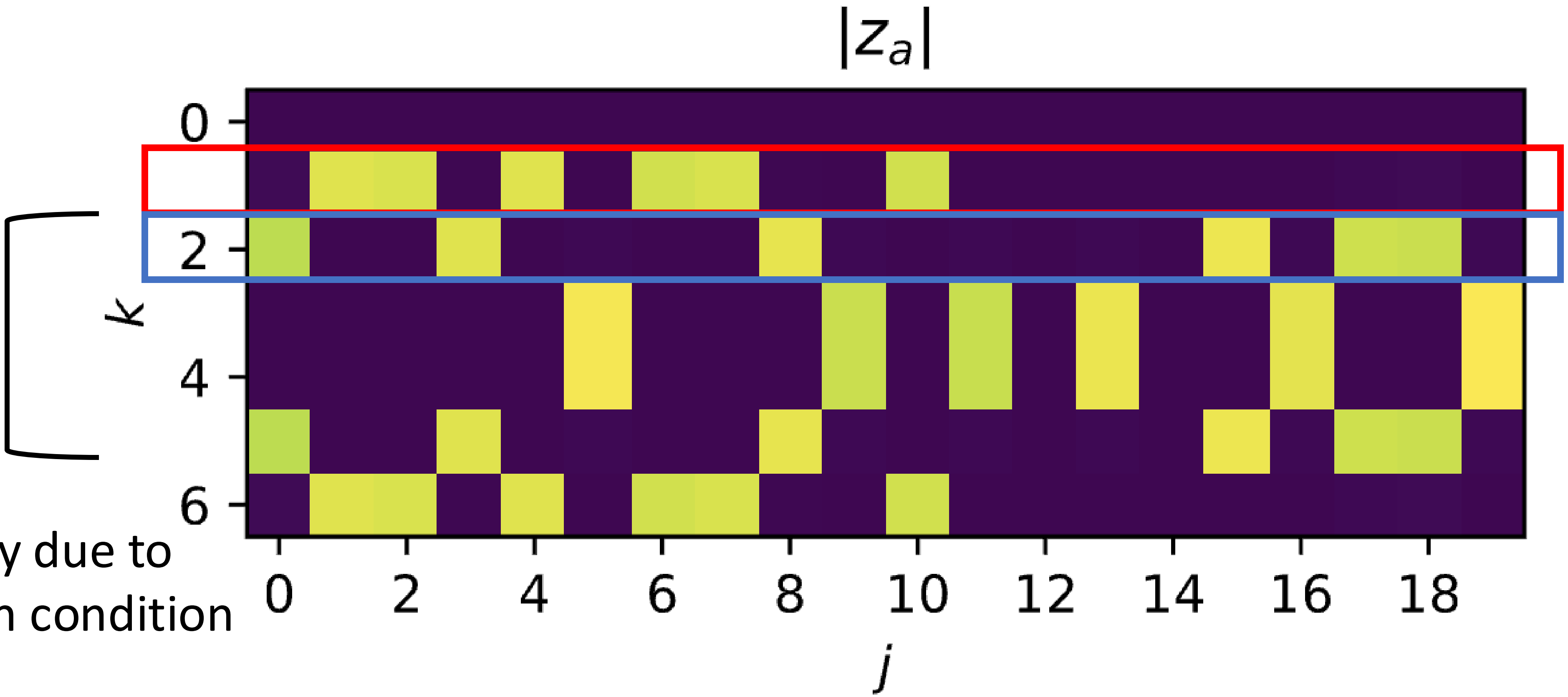


Frequency

Hidden node index

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

What a Gradient Descent Solution look like?

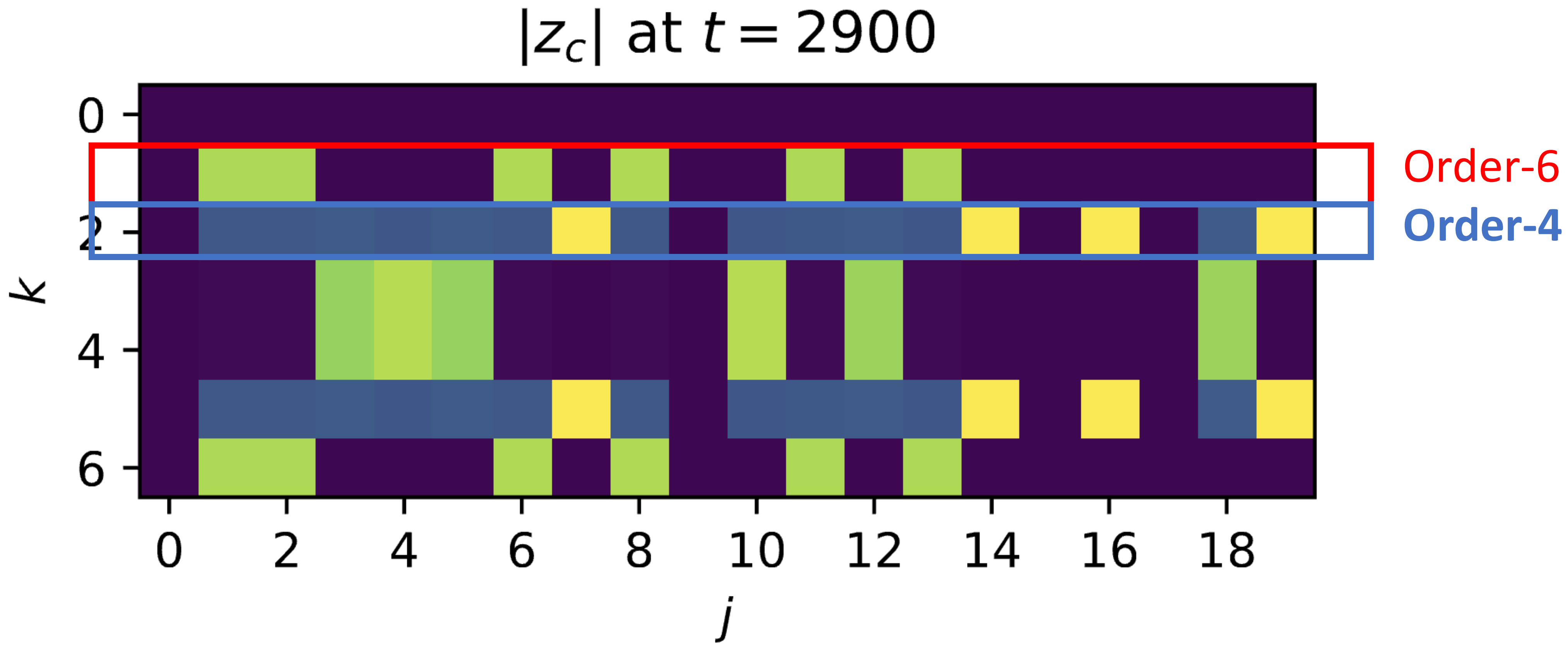


Order-6 solutions

Symmetry due to Hermitian condition

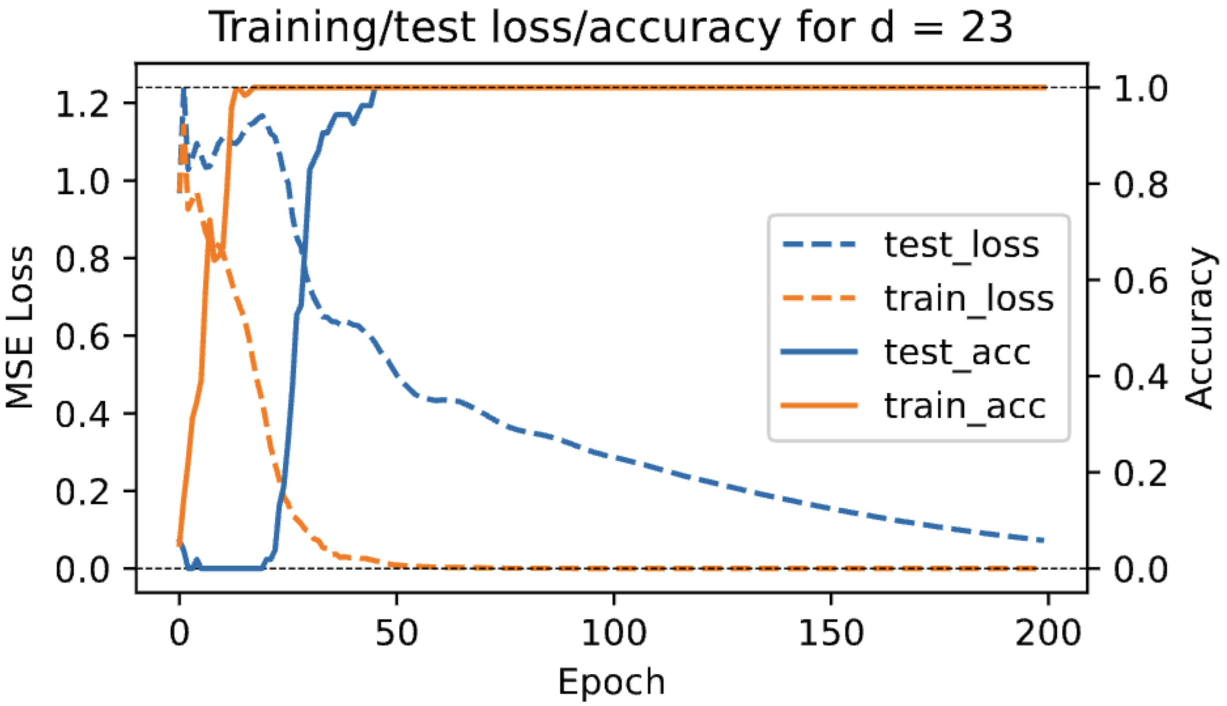
[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

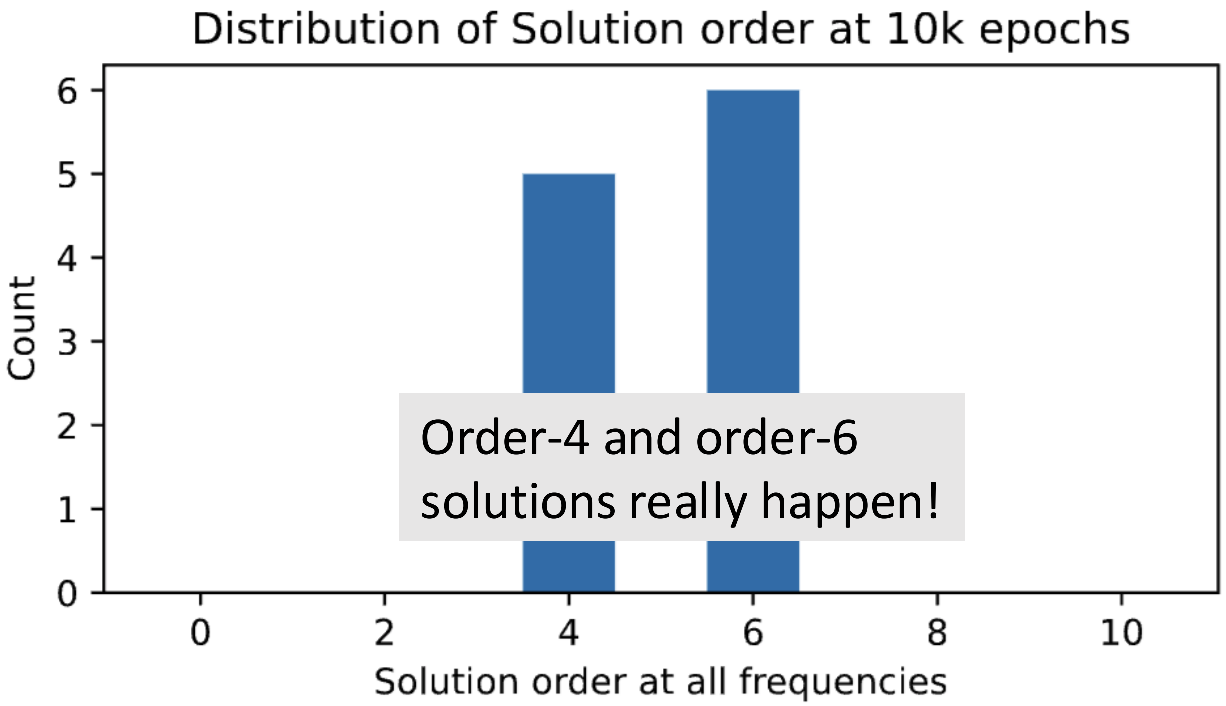
What a Gradient Descent Solution look like?



[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv’24]

More Statistics on Gradient Descent Solutions

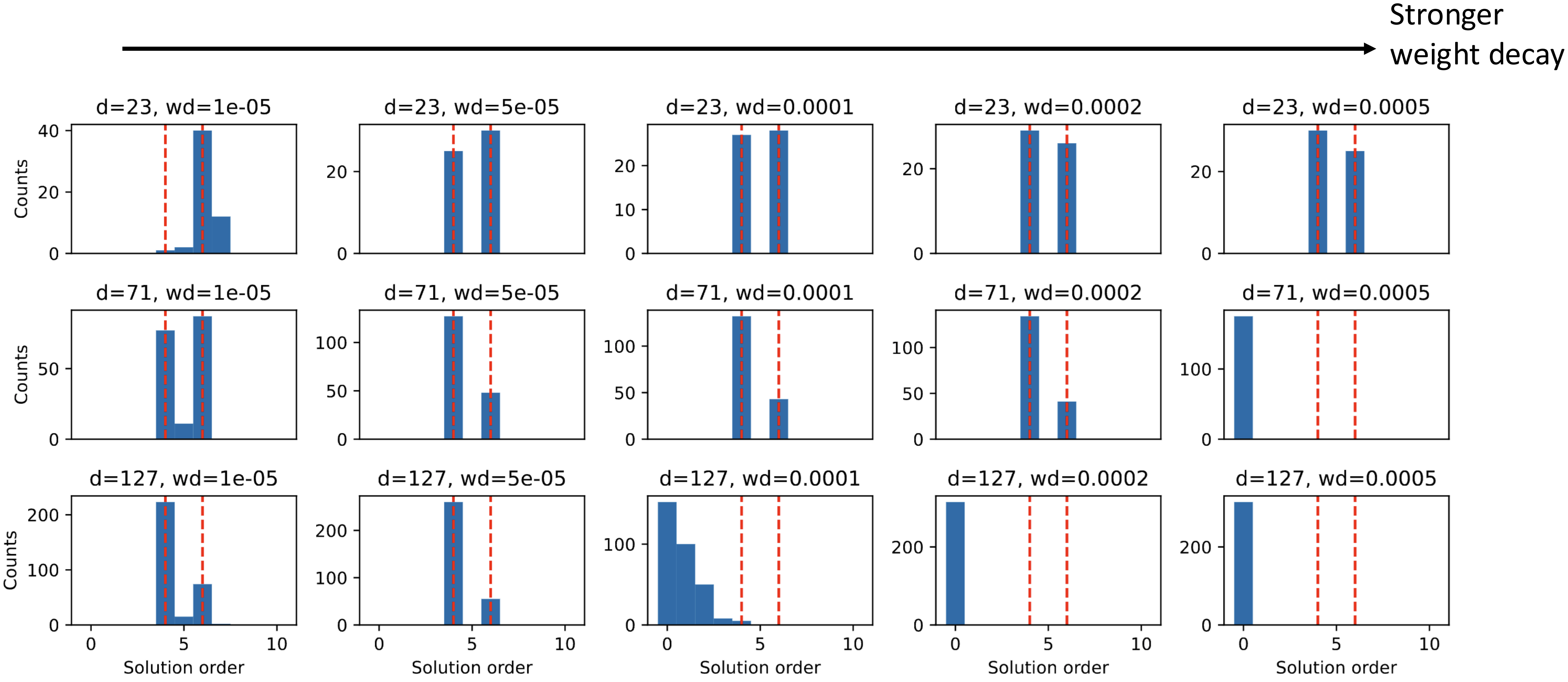

Training/test loss/accuracy for d = 23


Distribution of Solution order at 10k epochs

Order-4 and order-6 solutions really happen!

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv’24]

Effect of Weight Decay



facebook Artificial Intelligence

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv’24]

Why? 🤔

Structure of Loss Functions

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

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

Term $r_{k_1 k_2 k}(\boldsymbol{z}) \coloneqq \sum_j z_{ak_1 j} z_{bk_2 j} z_{ckj}$ and $r_{pk_1 k_2 k}(\boldsymbol{z}) \coloneqq \sum_j z_{pk_1 j} z_{pk_2 j} z_{ckj}$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Structure of Loss Functions

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

$$\ell_k(\boldsymbol{z}) = -2r_{kkk} + \sum_{k_1k_2}\left|r_{k_1k_2k}\right|^2 + \frac{1}{4}\left|\sum_{p\in\{a,b\}}\sum_{k'} r_{p,k',-k',k}\right|^2 + \frac{1}{4}\sum_{m\neq 0}\sum_{p\in\{a,b\}}\left|\sum_{k'} r_{p,k',m-k',k}\right|^2$$

Term $r_{k_1k_2k}(\boldsymbol{z}) := \sum_j z_{ak_1j}z_{bk_2j}z_{ckj}$ and $r_{pk_1k_2k}(\boldsymbol{z}) := \sum_j z_{pk_1j}z_{pk_2j}z_{ckj}$

Sufficient conditions of Global Optimizers:

R_{g}	R_{c}	R_{n}	R_*
$r_{kkk} = 1$	$r_{k_1k_2k} = 0$	$r_{pk',-k',k} = 0$	$r_{pk',m-k',k} = 0$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

How to Optimize?

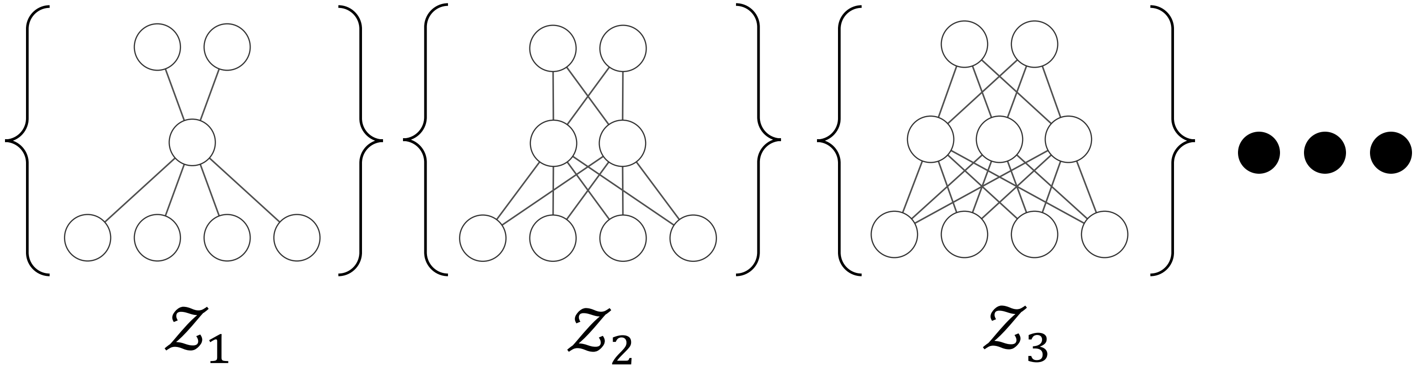
The objective is highly nonlinear !!

However, nice *algebraic structures* exist!

How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!

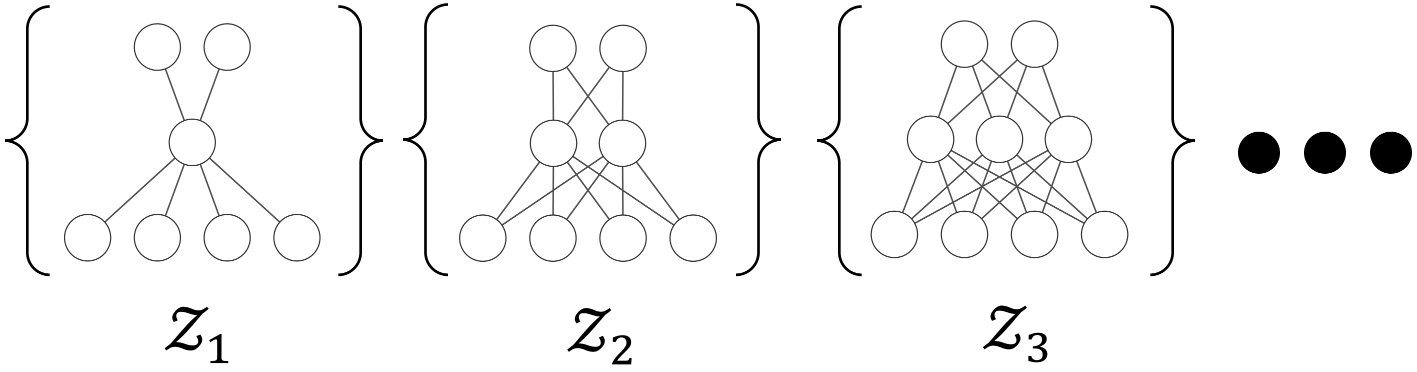


$\mathcal{Z} = \bigcup_{q\geq 0} \mathcal{Z}_q$: All 2-layer networks with different number of hidden nodes

How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!



$\mathcal{Z} = \cup_{q\geq 0} \mathcal{Z}_q$: 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 $r(\boldsymbol{z}): \mathcal{Z} \mapsto \mathbb{C}$ is a *ring homomorphism*, if

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

Ring Homomorphism

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

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

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

MSE Loss

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

Ring Homomorphism

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

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

🎯 $r_{k_1k_2k}(\mathbf{z})$ and $r_{pk_1k_2k}(\mathbf{z})$ are ***ring homomorphisms***!

MSE Loss

$$\ell_k(\mathbf{z}) = -2r_{kkk} + \sum_{k_1k_2} \left|r_{k_1k_2k}\right|^2 + \frac{1}{4}\left|\sum_{p\in\{a,b\}}\sum_{k'} r_{p,k',-k',k}\right|^2 + \frac{1}{4}\sum_{m\neq 0}\sum_{p\in\{a,b\}}\left|\sum_{k'} r_{p,k',m-k',k}\right|^2$$

Partial solution $\mathbf{z}_1$ satisfies $r_{k_1k_2k}(\mathbf{z}_1) = 0$

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

Ring Homomorphism

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

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

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

MSE Loss

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

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

Composing Global Optimizers from Partial Ones

Partial solution #1

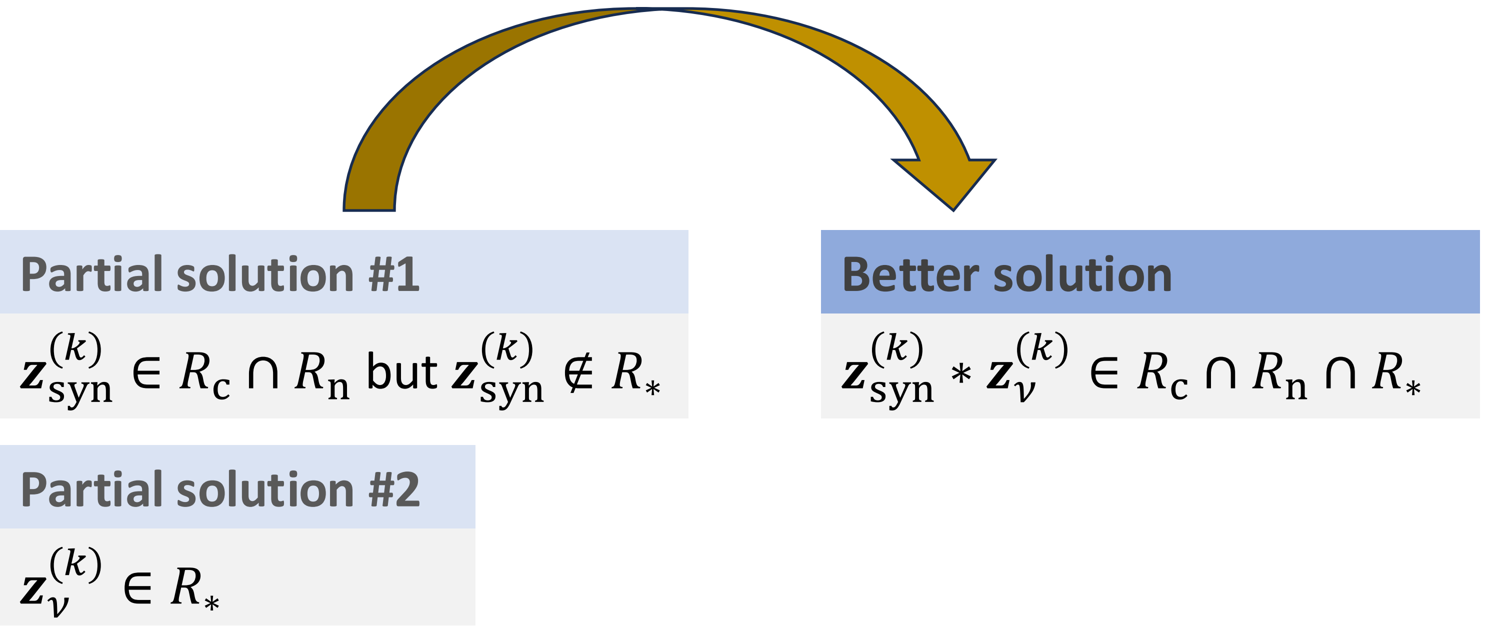
$\boldsymbol{z}_{\mathrm{syn}}^{(k)} \in R_{\mathrm{c}} \cap R_{\mathrm{n}}$ but $\boldsymbol{z}_{\mathrm{syn}}^{(k)} \notin R_*$

Partial solution #2

$\boldsymbol{z}_{\nu}^{(k)} \in R_*$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Composing Global Optimizers from Partial Ones



Compositing
solutions using
ring multiplication $*$

Partial solution #1

$\boldsymbol{z}_{\text{syn}}^{(k)} \in R_{\text{c}} \cap R_{\text{n}}$ but $\boldsymbol{z}_{\text{syn}}^{(k)} \notin R_*$

Partial solution #2

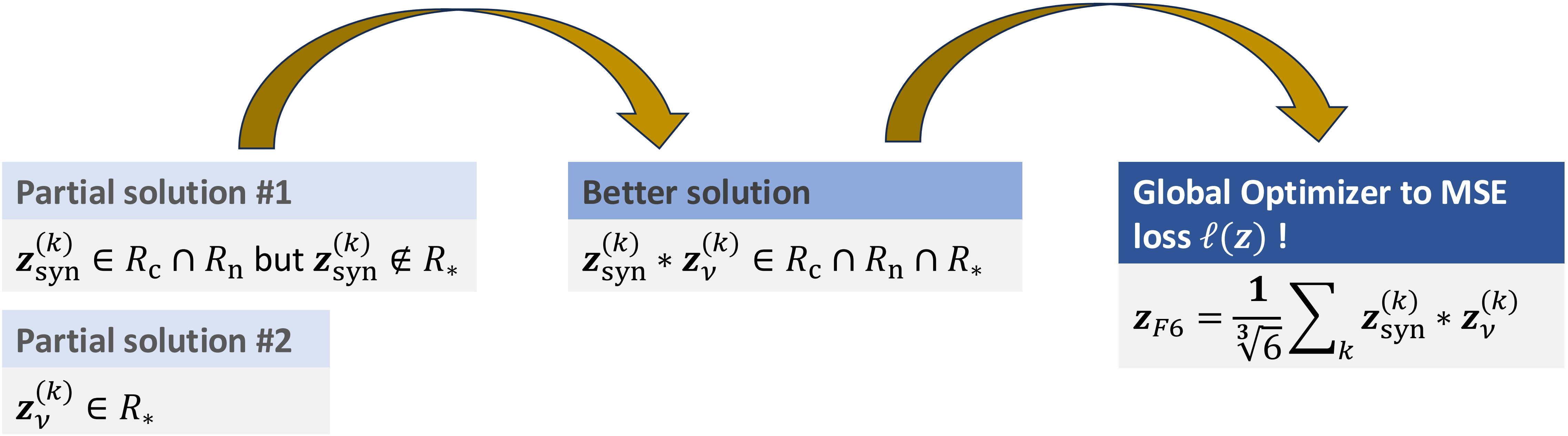
$\boldsymbol{z}_{\nu}^{(k)} \in R_*$

Better solution

$\boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} \in R_{\text{c}} \cap R_{\text{n}} \cap R_*$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Composing Global Optimizers from Partial Ones



Compositing solutions using ***ring multiplication*** $*$

Compositing solutions using ***ring addition*** $+$

Partial solution #1

$\boldsymbol{z}_{\mathrm{syn}}^{(k)} \in R_{\mathrm{c}} \cap R_{\mathrm{n}}$ but $\boldsymbol{z}_{\mathrm{syn}}^{(k)} \notin R_*$

Partial solution #2

$\boldsymbol{z}_{\nu}^{(k)} \in R_*$

Better solution

$\boldsymbol{z}_{\mathrm{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} \in R_{\mathrm{c}} \cap R_{\mathrm{n}} \cap R_*$

Global Optimizer to MSE loss $\ell(\boldsymbol{z})$!

$$\boldsymbol{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_k \boldsymbol{z}_{\mathrm{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)}$$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Exemplar constructed global optimizers

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

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

Exemplar constructed global optimizers

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

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

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

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

Exemplar constructed global optimizers

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

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

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

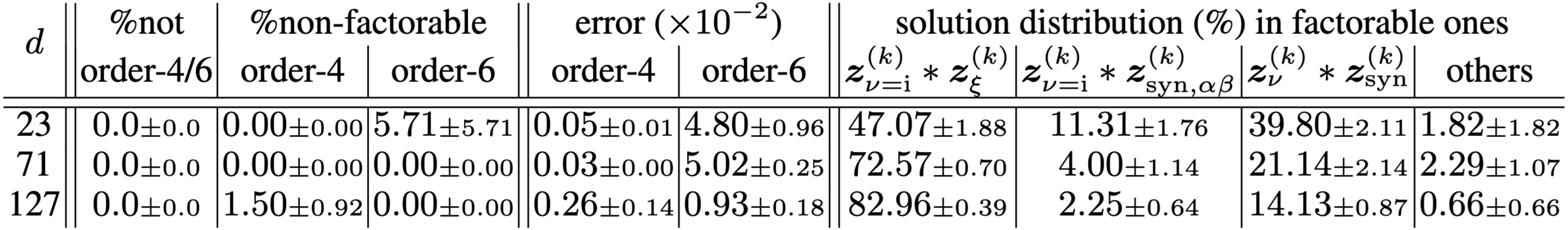
$$\boldsymbol{z}_{F4/6} = \frac{1}{\sqrt[3]{6}} \hat{\boldsymbol{z}}_{F6}^{(k_0)} + \frac{1}{\sqrt[3]{4}} \sum_{k=1, k\neq k_0}^{(d-1)/2} \boldsymbol{z}_{F4}^{(k)}$$

Perfect memorization
(order-d per frequency)

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

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

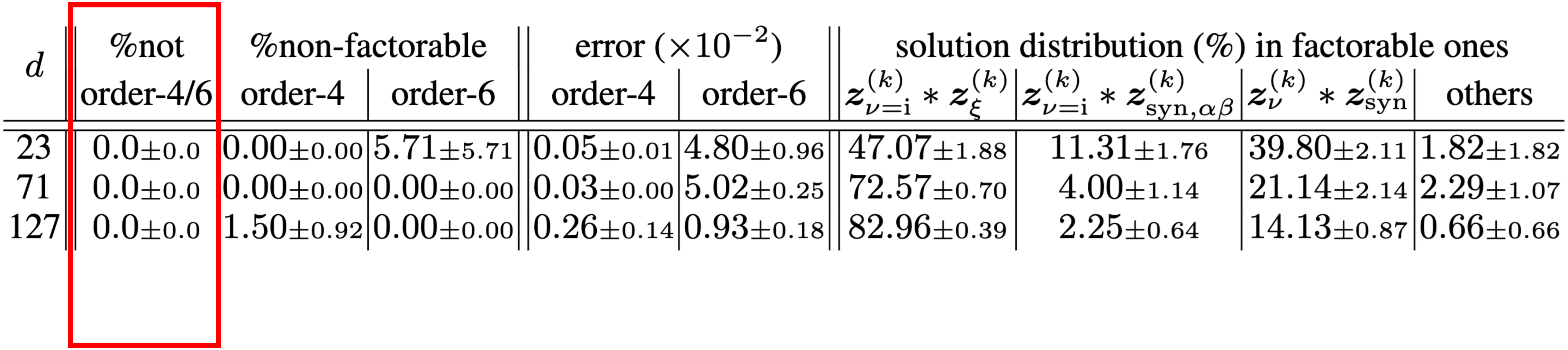
Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\xi}$	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\mathrm{syn},\alpha\beta}$	$\boldsymbol{z}^{(k)}_{\nu} * \boldsymbol{z}^{(k)}_{\mathrm{syn}}$	others
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	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

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

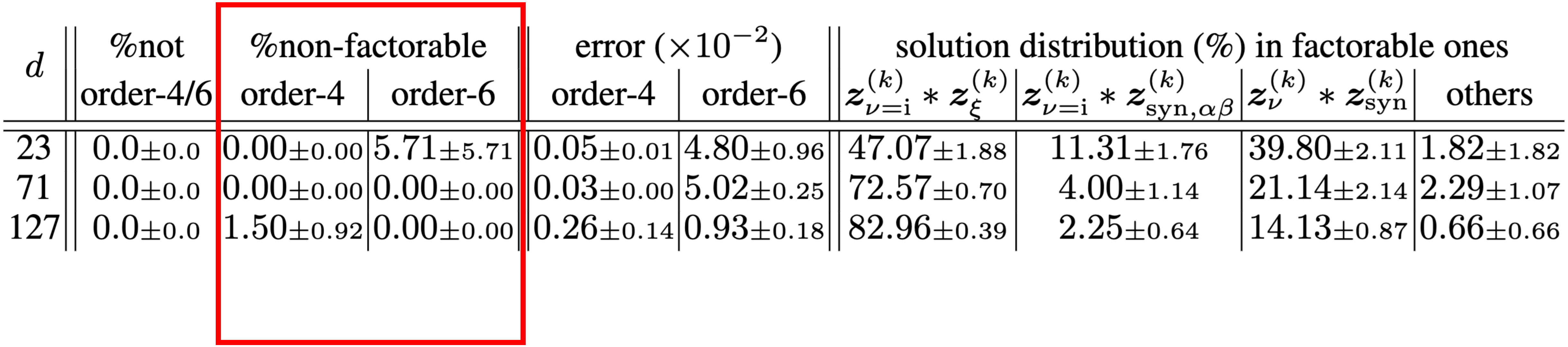
Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\xi}$	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\mathrm{syn},\alpha\beta}$	$\boldsymbol{z}^{(k)}_{\nu} * \boldsymbol{z}^{(k)}_{\mathrm{syn}}$	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

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

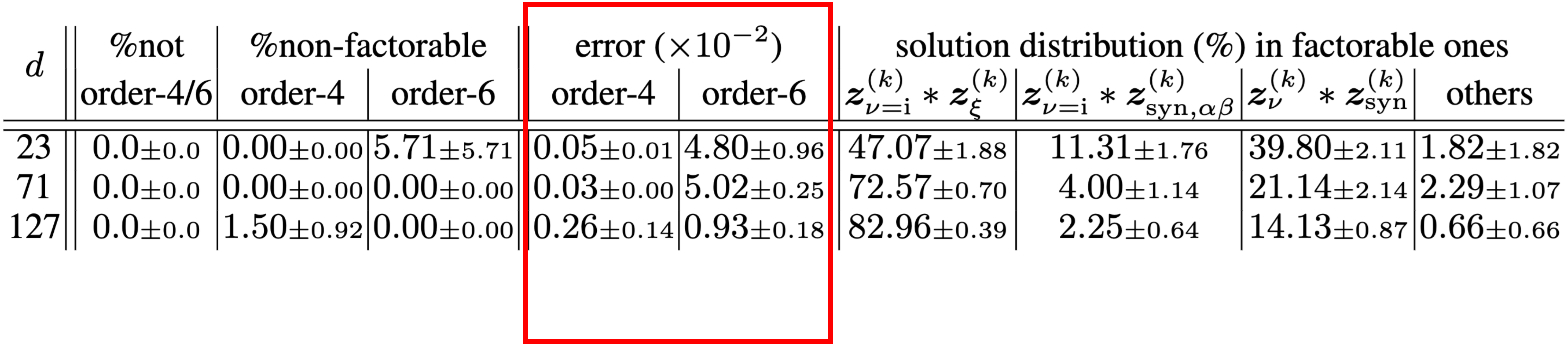
Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$\boldsymbol{z}_{\nu=\mathrm{i}}^{(k)} * \boldsymbol{z}_{\xi}^{(k)}$	$\boldsymbol{z}_{\nu=\mathrm{i}}^{(k)} * \boldsymbol{z}_{\mathrm{syn},\alpha\beta}^{(k)}$	$\boldsymbol{z}_{\nu}^{(k)} * \boldsymbol{z}_{\mathrm{syn}}^{(k)}$	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”

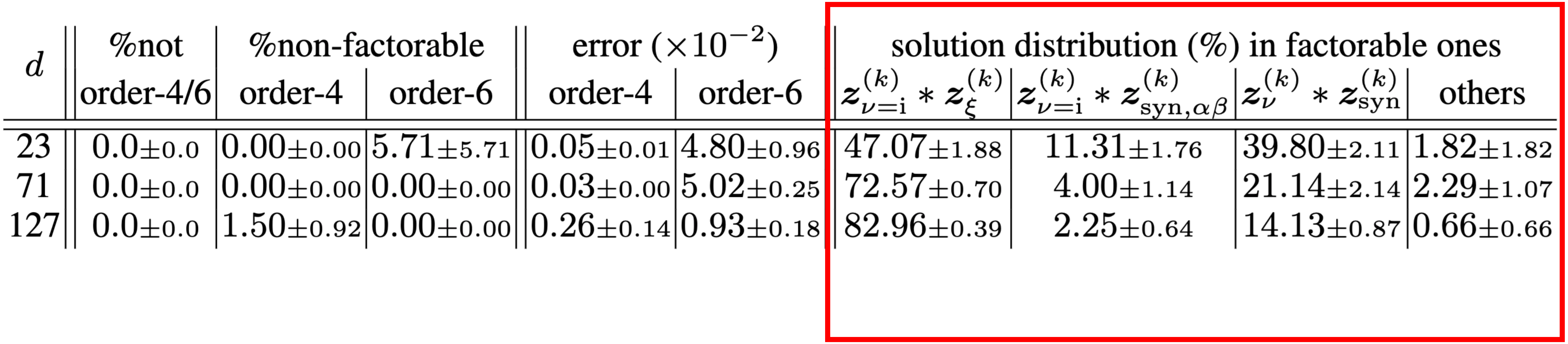
Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\xi}$	$\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\mathrm{syn},\alpha\beta}$	$\boldsymbol{z}^{(k)}_{\nu} * \boldsymbol{z}^{(k)}_{\mathrm{syn}}$	others
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	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

Factorization error is very small

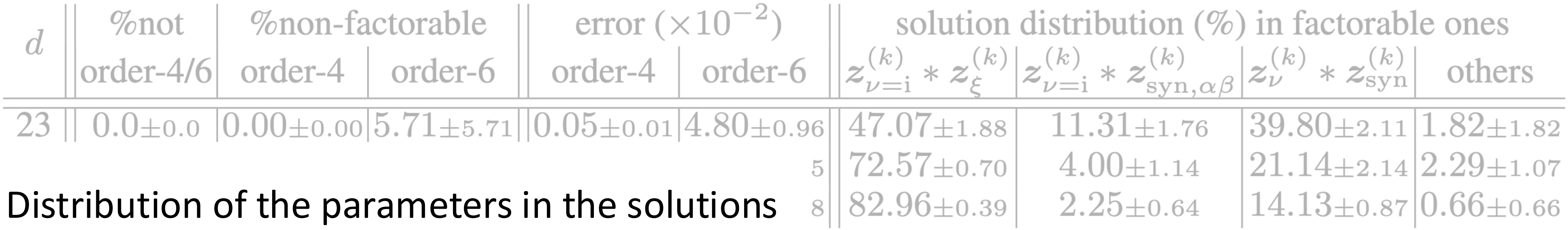
Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$\boldsymbol{z}_{\nu=\mathrm{i}}^{(k)} * \boldsymbol{z}_{\xi}^{(k)}$	$\boldsymbol{z}_{\nu=\mathrm{i}}^{(k)} * \boldsymbol{z}_{\mathrm{syn},\alpha\beta}^{(k)}$	$\boldsymbol{z}_{\nu}^{(k)} * \boldsymbol{z}_{\mathrm{syn}}^{(k)}$	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

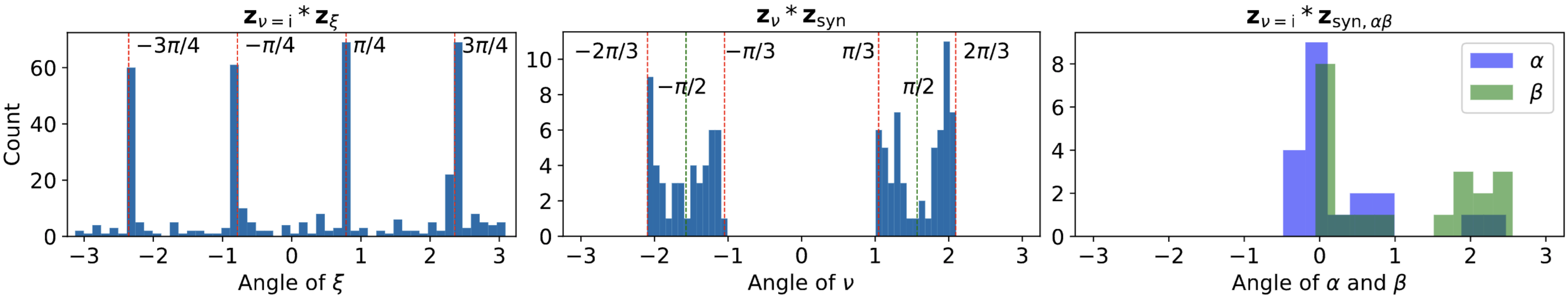
98% of the solutions can be factorizable into the constructed forms

Gradient Descent solutions matches with construction



d	%not	%non-factorable		error ($\times 10^{-2}$)		solution distribution (%) in factorable ones			
	order-4/6	order-4	order-6	order-4	order-6	$z_{\nu=\mathrm{i}}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=\mathrm{i}}^{(k)} * z_{\mathrm{syn},\alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\mathrm{syn}}^{(k)}$	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!