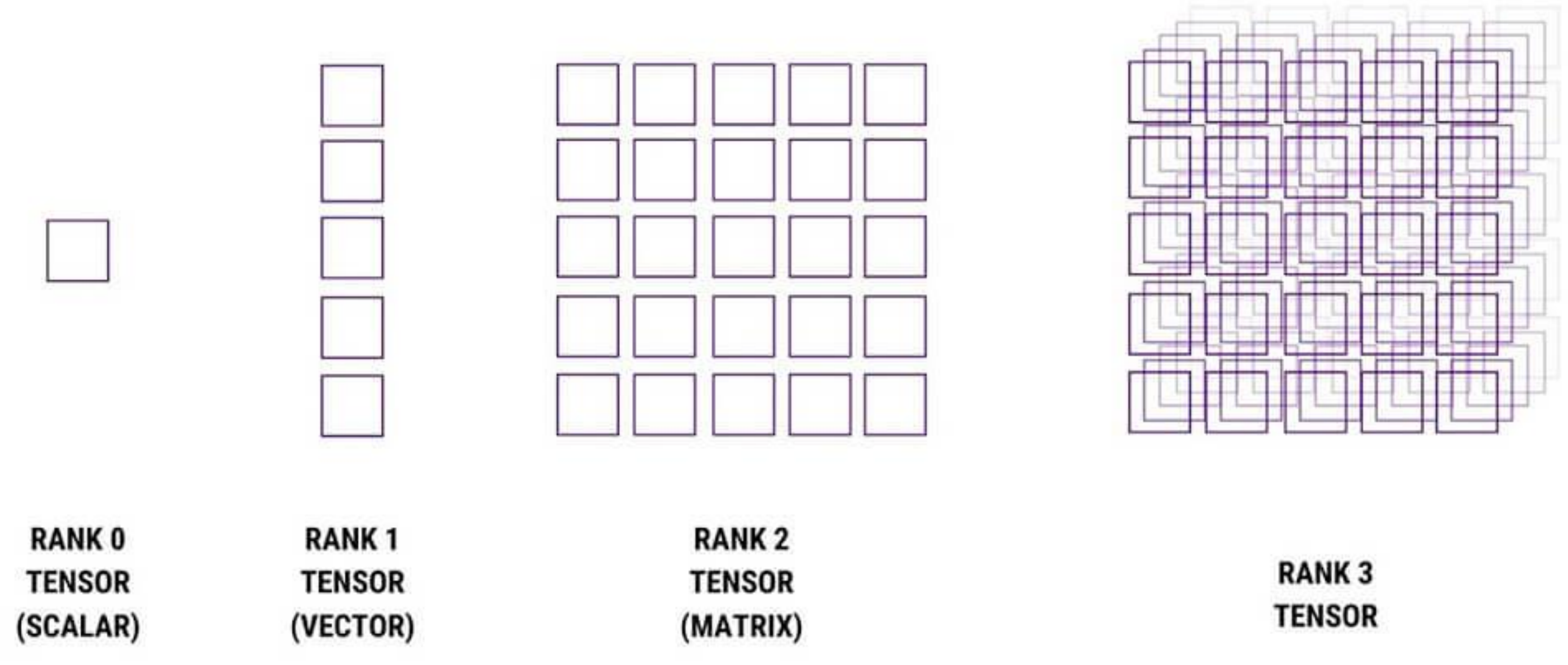
Learning with Graphs

Roger Wattenhofer

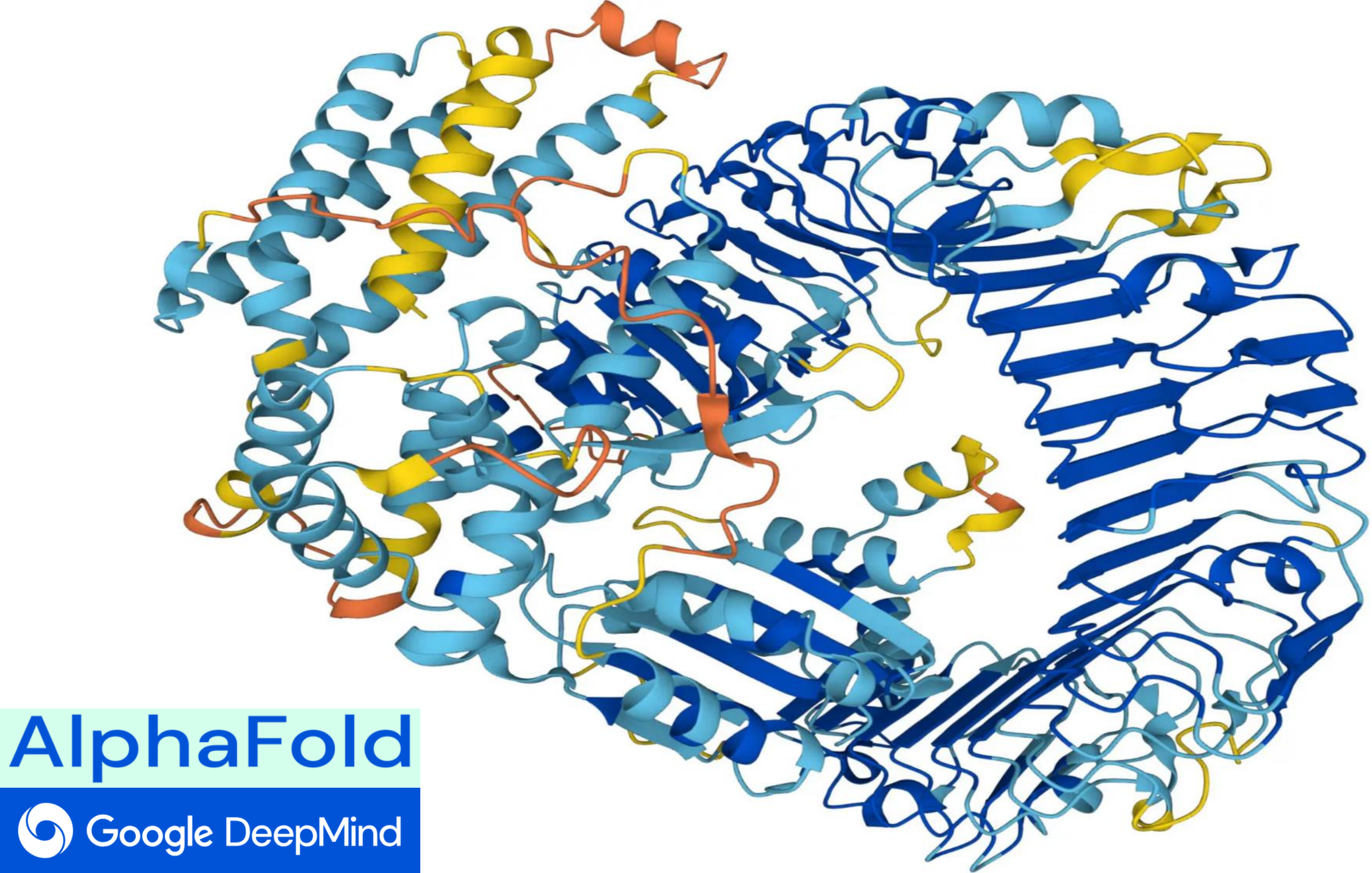

Midjourney

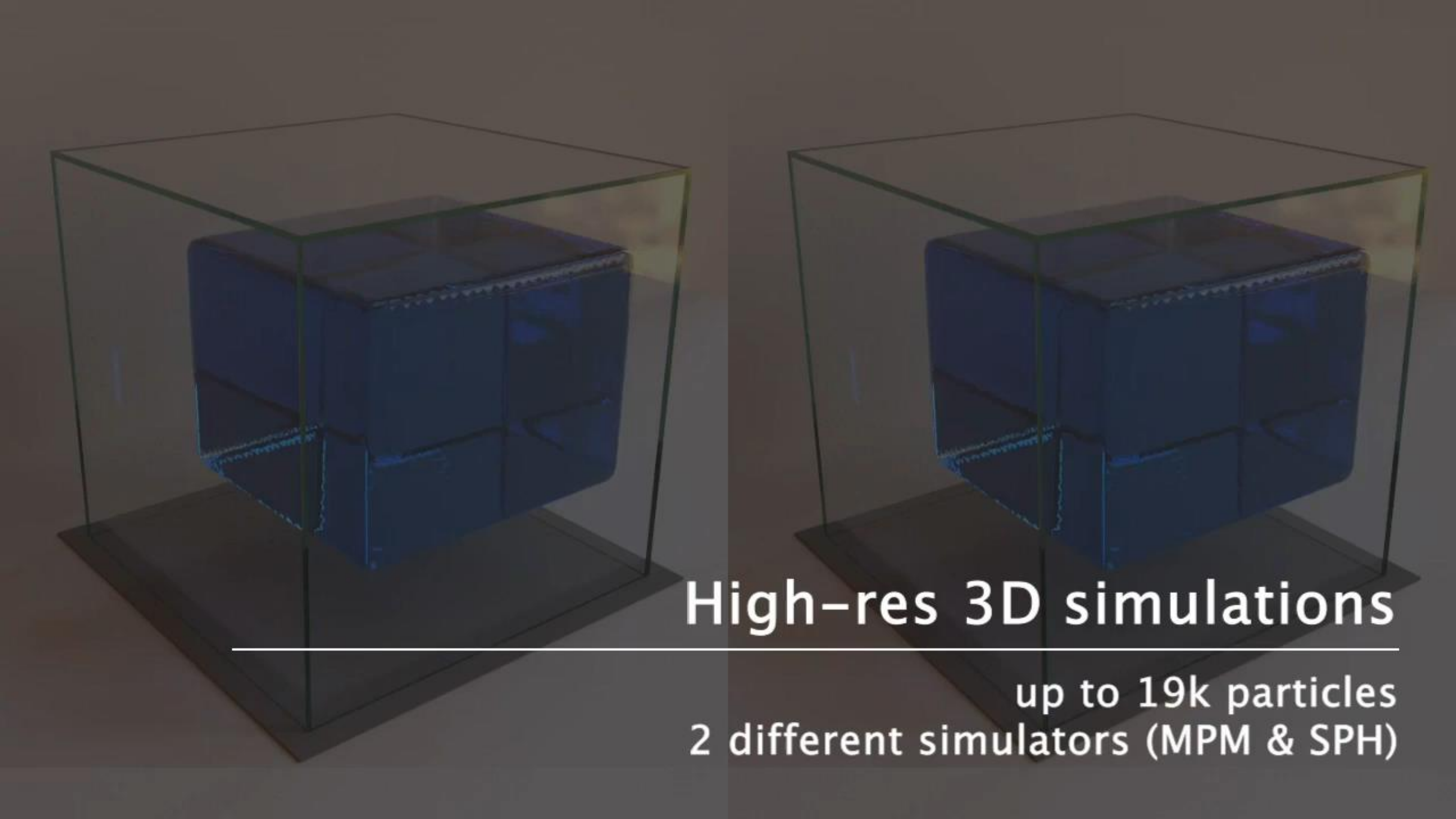

Boris Eldagsen

Machine Learning Deals with …

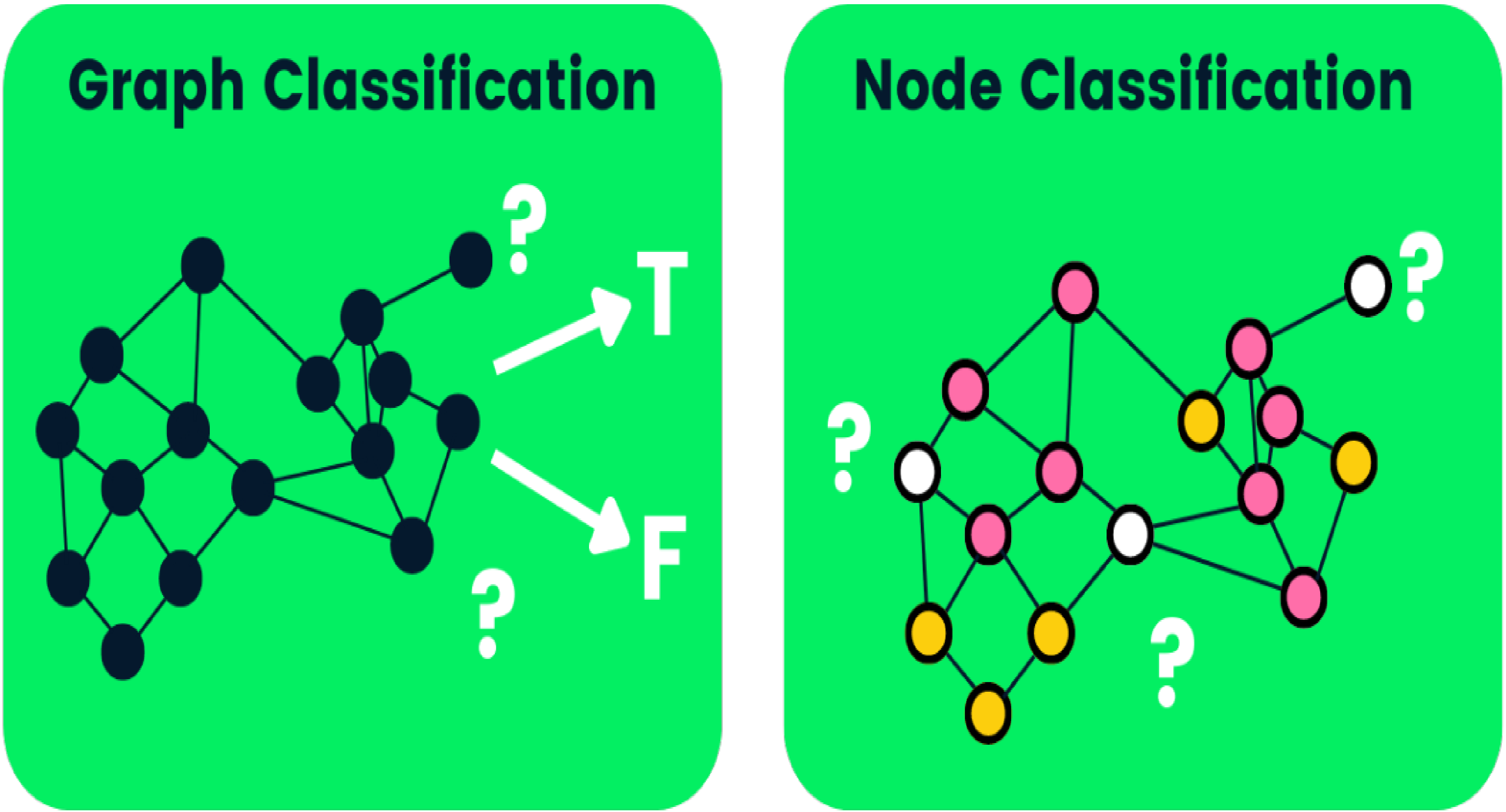


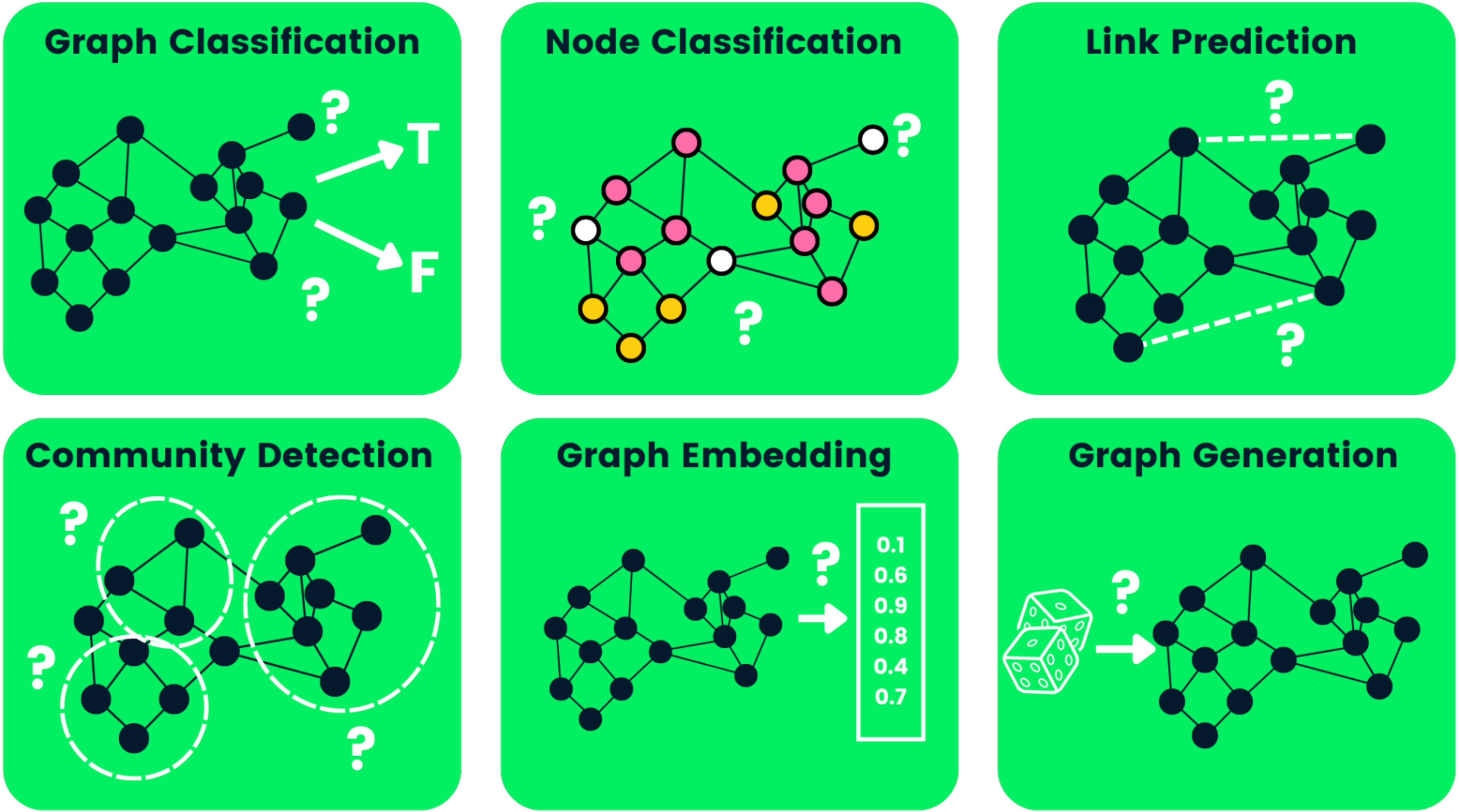
Networks
Social Networks
Neural Networks
Mobile Networks
Wireless Networks
Financial Networks
Economic Networks
Biological Networks
Computer Networks


AlphaFold
Google DeepMind


High-res 3D simulations
up to 19k particles
2 different simulators (MPM & SPH)

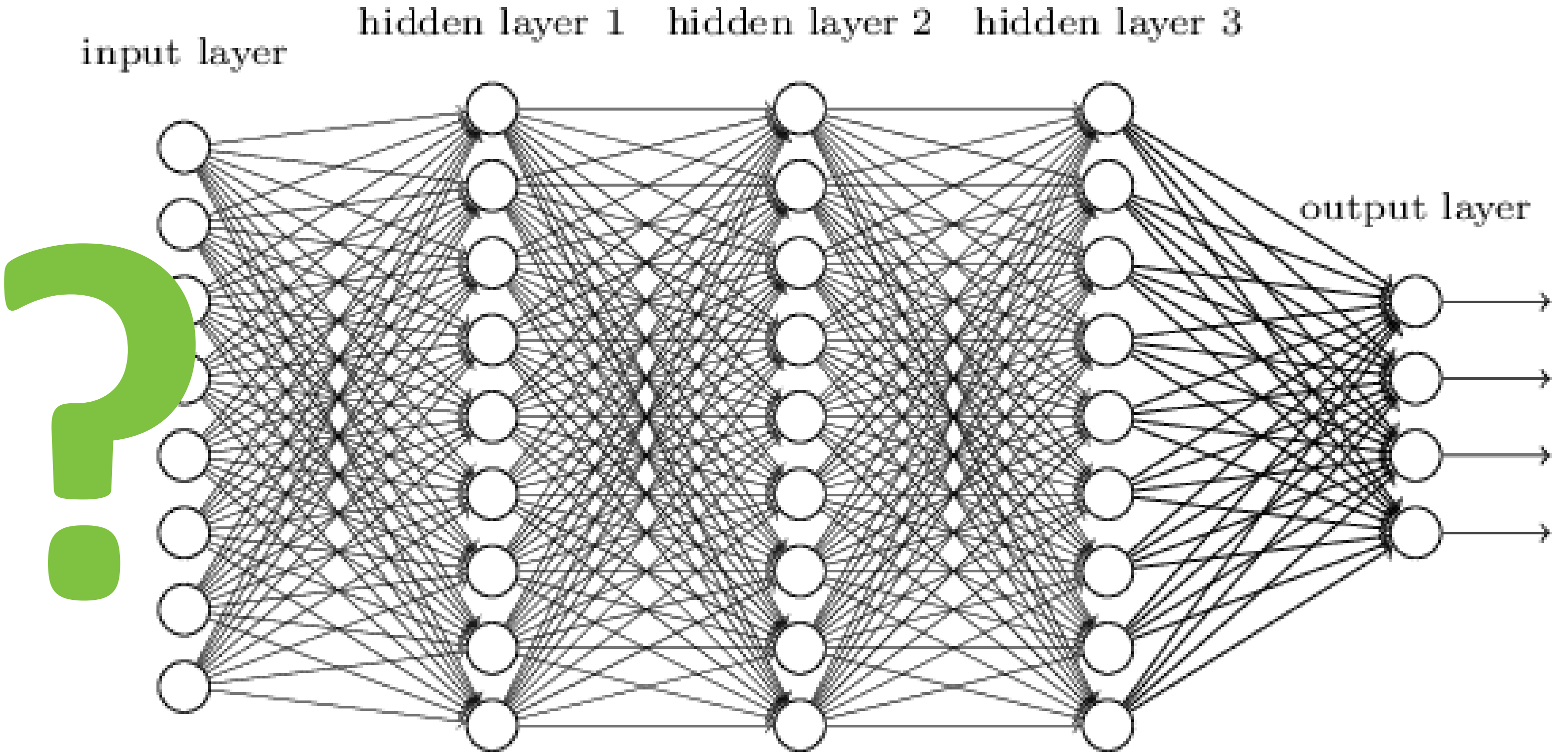
Refik Anadol, Unsupervised, MOMA


Graph Classification
?
T
F
?
Node Classification
?
?
?

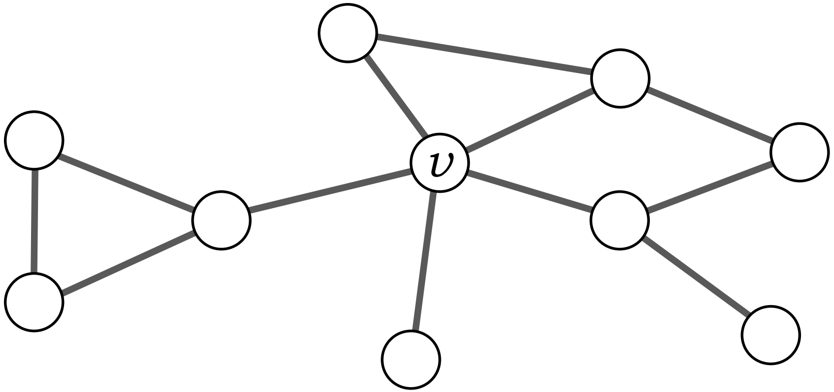

Graph Classification
T
F
Node Classification
Link Prediction
Community Detection
Graph Embedding
0.1
0.6
0.9
0.8
0.4
0.7
Graph Generation

Graph Neural Networks

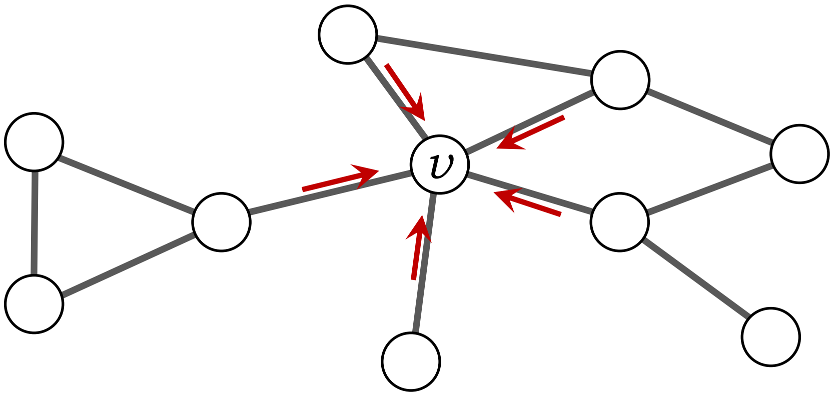
Roger Wattenhofer


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

Graph Neural Networks

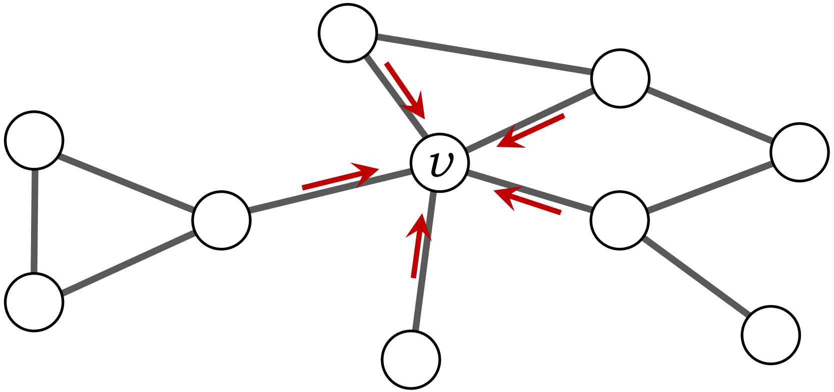


Graph Neural Networks



$a_v = \textsc{Aggregate}\left(\{\{ h_u \mid u \in N(v) \}\} \right)$ (Min, Max, Mean, Sum)

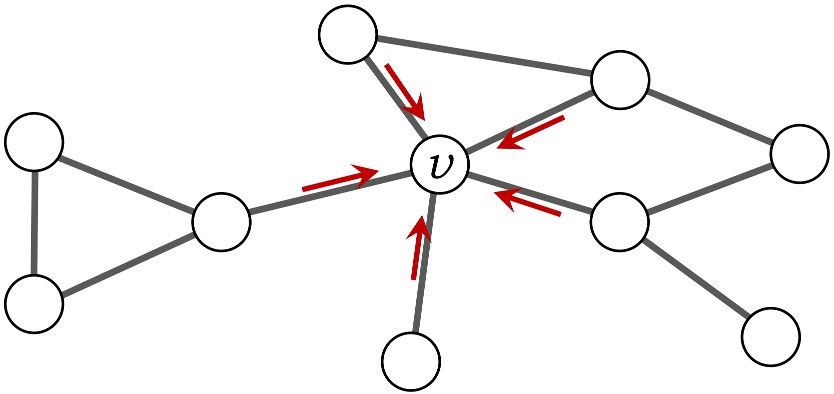
Graph Neural Networks



a_v = AGGREGATE ({{ h_u | $u \in N(v)$ }}) (Min, Max, Mean, Sum)

$h_v^{(t+1)}$ = UPDATE (h_v, a_v)

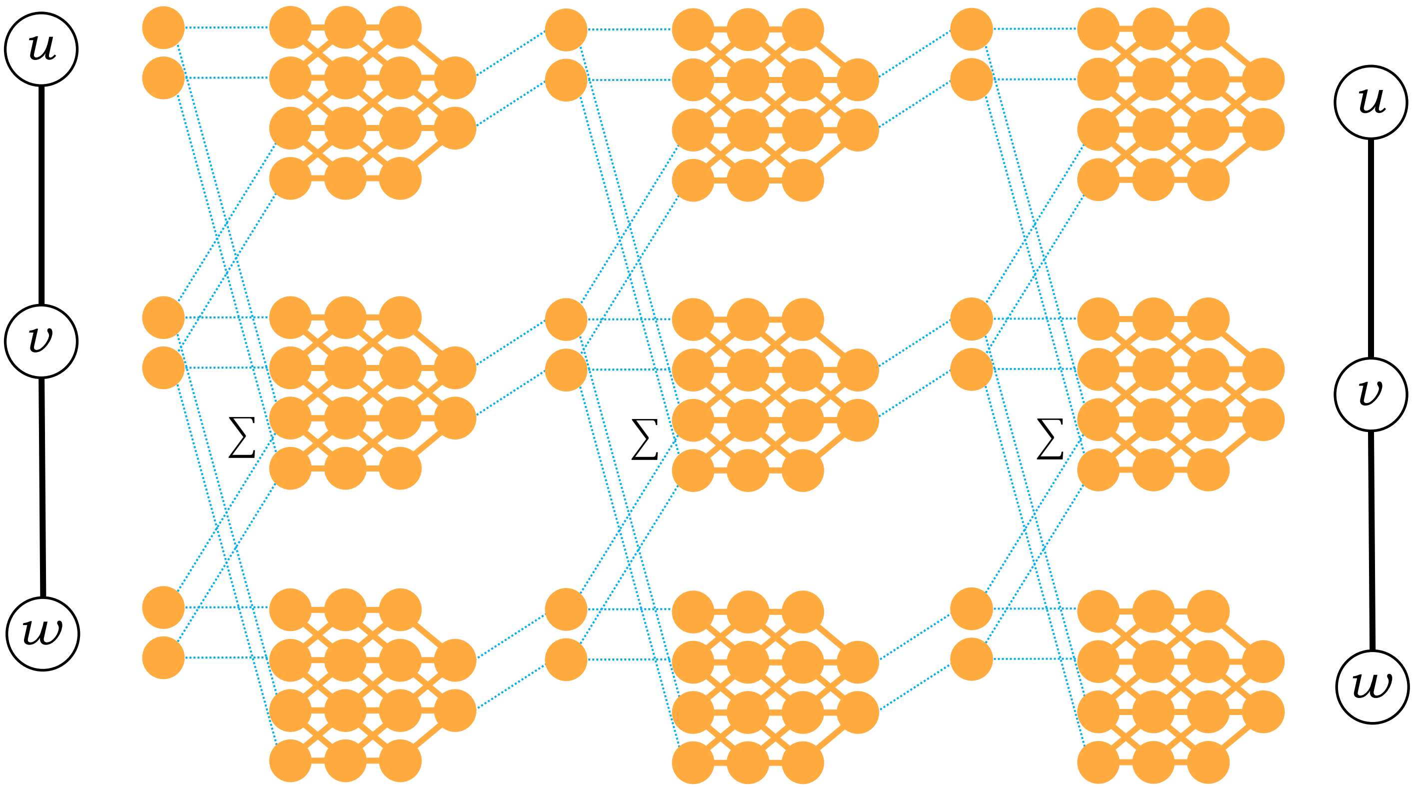
Graph Neural Networks



a_v = AGGREGATE ({{ h_u | $u \in N(v)$ }}) (Min, Max, Mean, Sum)

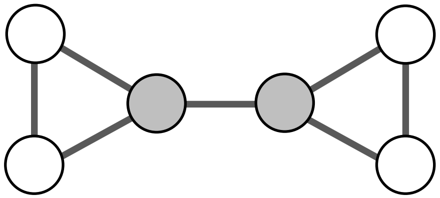
$h_v^{(t+1)}$ = UPDATE (h_v, a_v) (Little Neural Network)

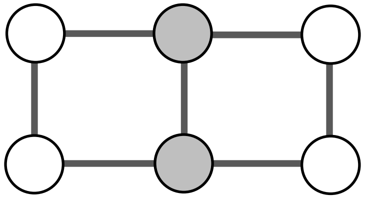
Graph Neural Networks



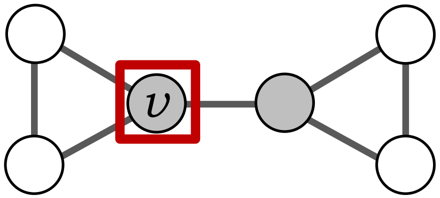
GNN Limitations?

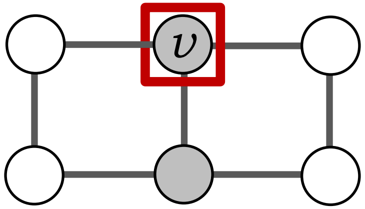
Limits of GNNs





Limits of GNNs





More Expressive GNNs?

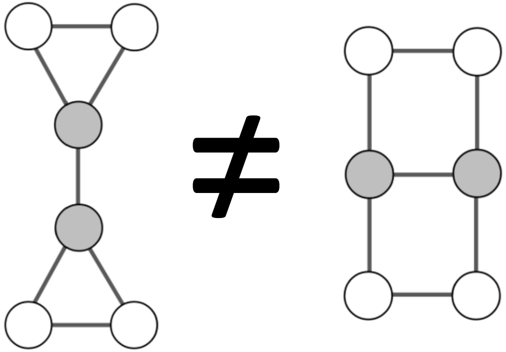
DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks

Pál András Papp
ETH Zurich
apapp@ethz.ch

Karolis Martinkus
ETH Zurich
martinkus@ethz.ch

Lukas Faber
ETH Zurich
lfaber@ethz.ch

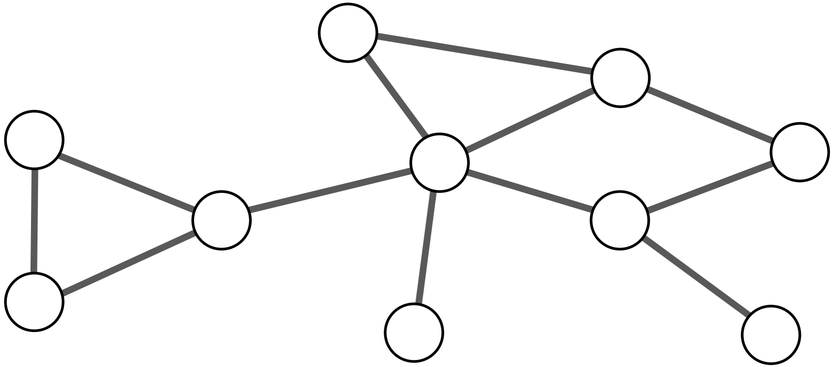
Roger Wattenhofer
ETH Zurich
wattenhofer@ethz.ch



GNNs with Dropouts

Multiple runs of the GNN

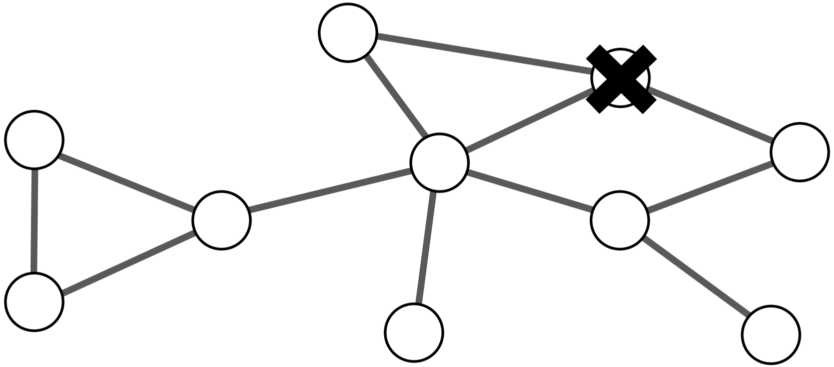
Each node removed with probability *p* independently



GNNs with Dropouts

Multiple runs of the GNN

Each node removed with probability p independently

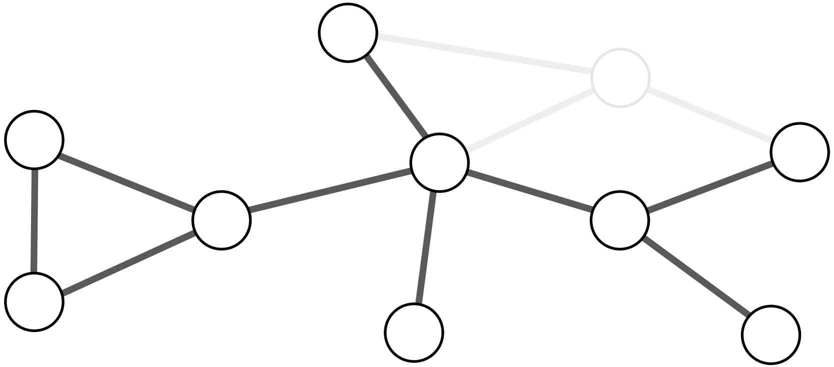


Run #1

GNNs with Dropouts

Multiple runs of the GNN

Each node removed with probability *p* independently

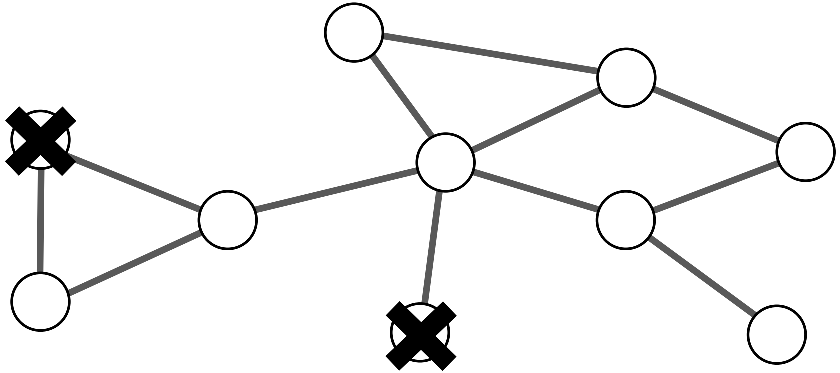


Run #1

GNNs with Dropouts

Multiple runs of the GNN

Each node removed with probability *p* independently

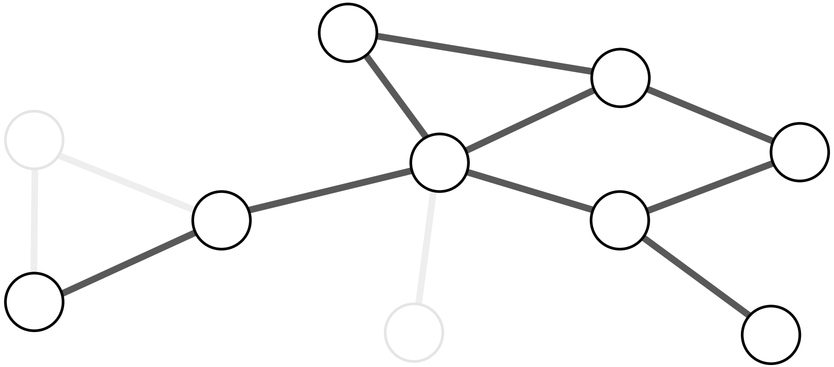


Run #2

GNNs with Dropouts

Multiple runs of the GNN

Each node removed with probability *p* independently

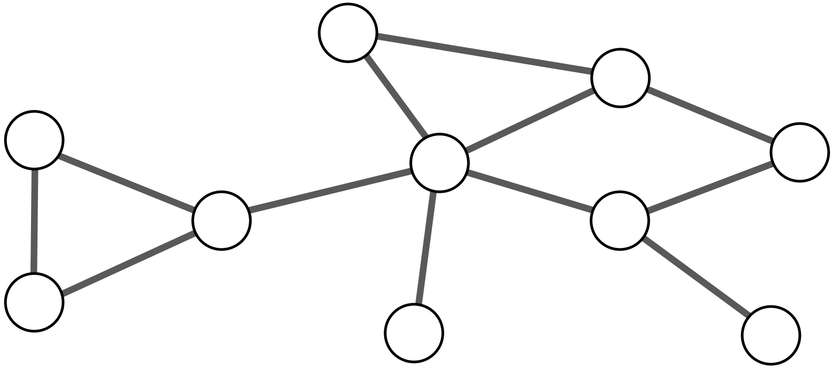


Run #2

GNNs with Dropouts

Multiple runs of the GNN

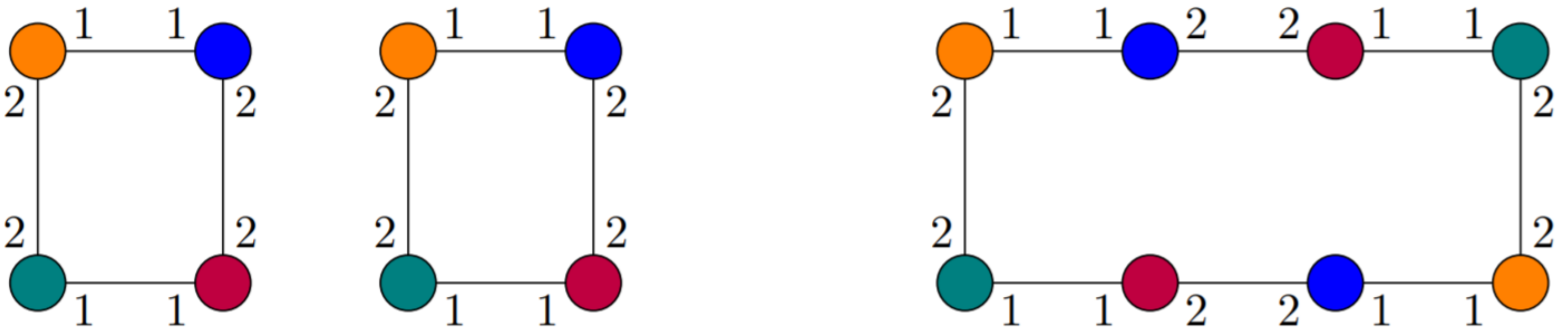
Each node removed with probability *p* independently



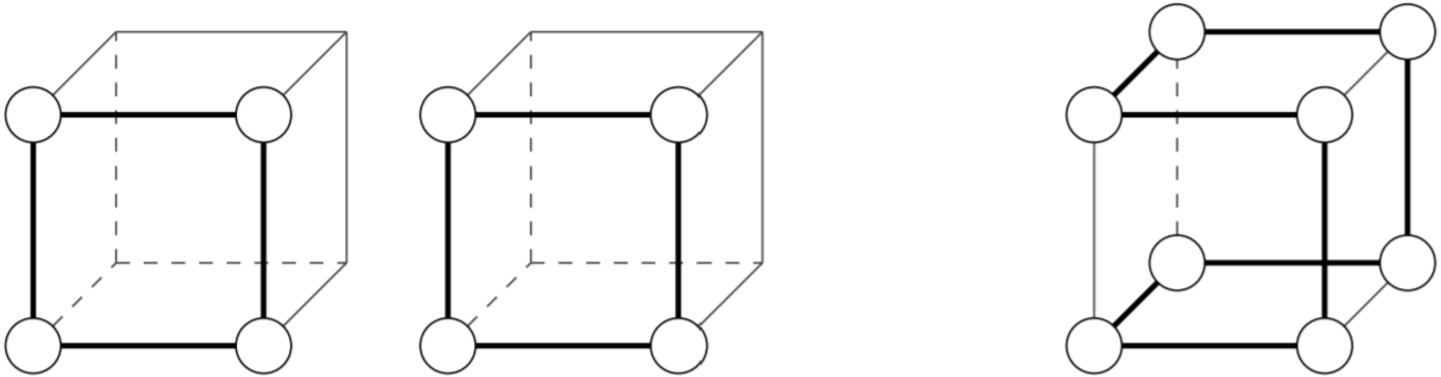
Run #3

GNNs with Dropouts

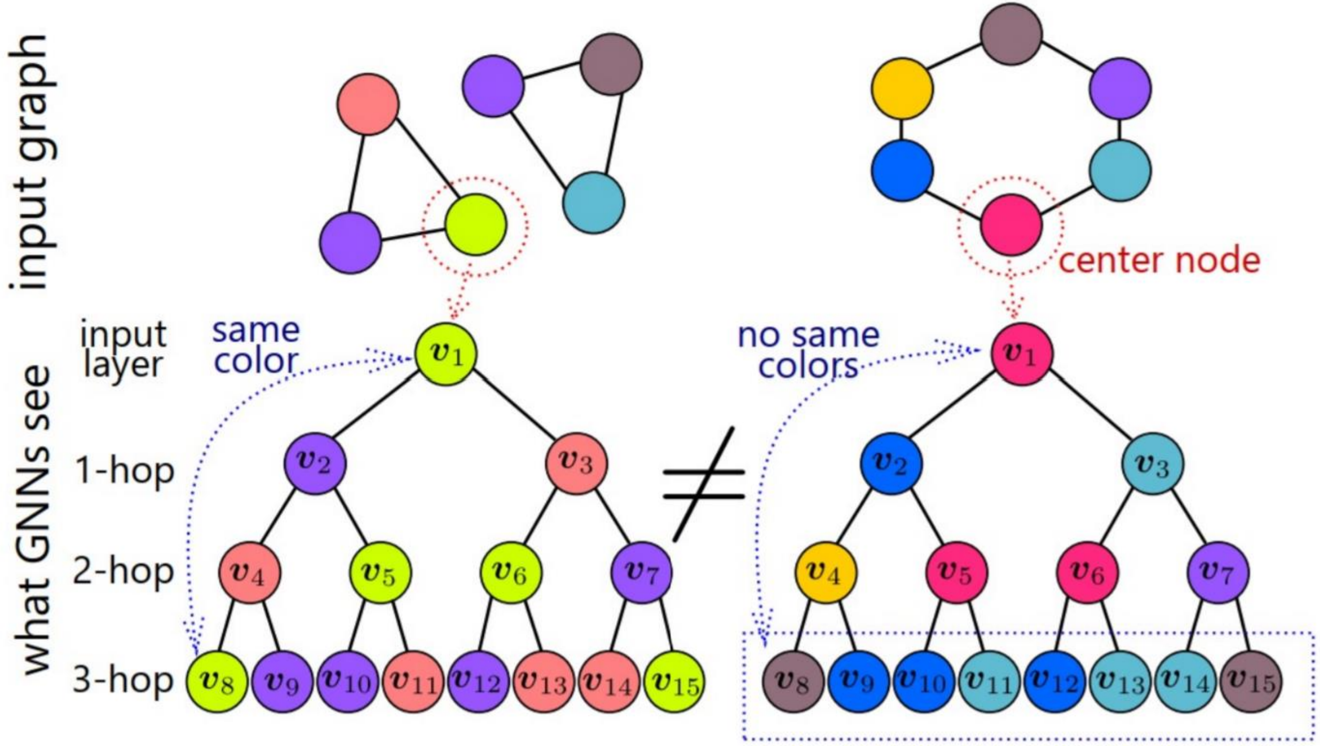
Port Numbers



Angle Features

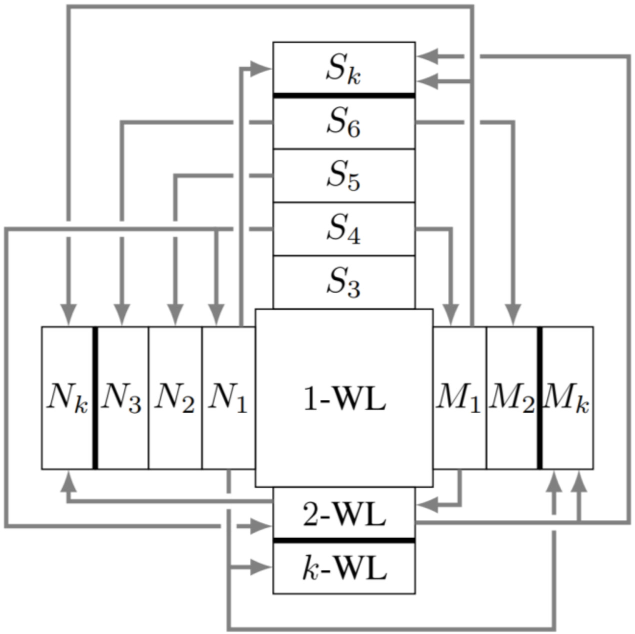


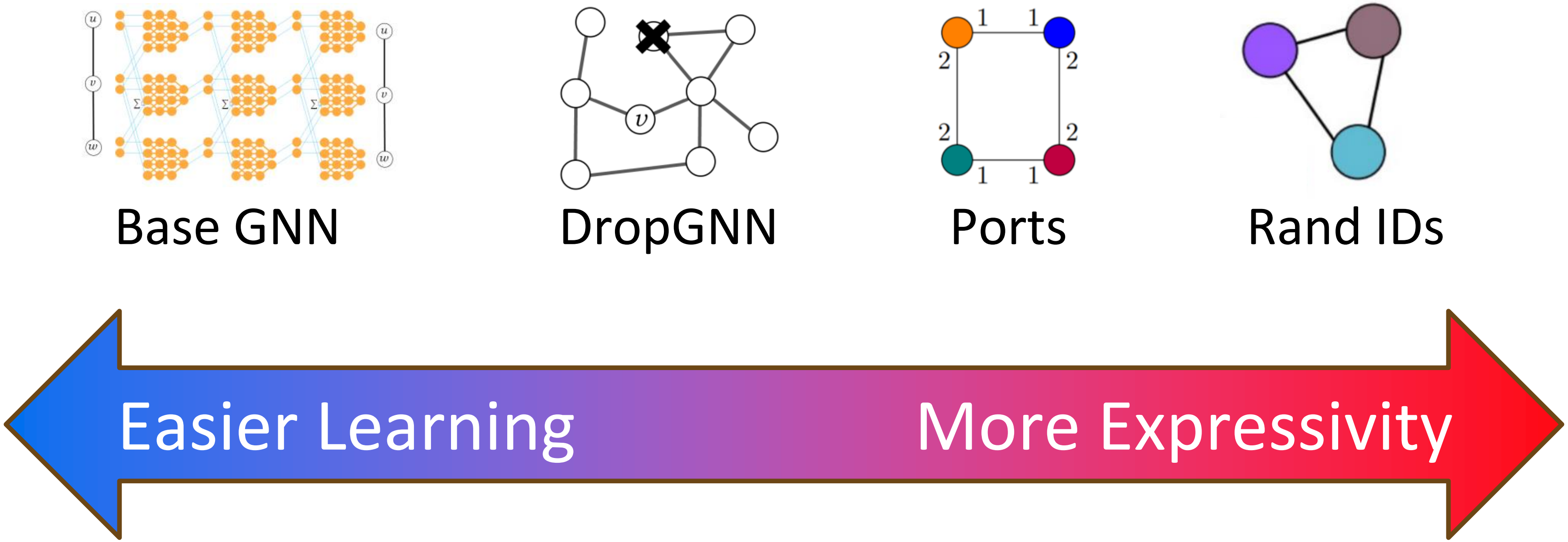
Random Features



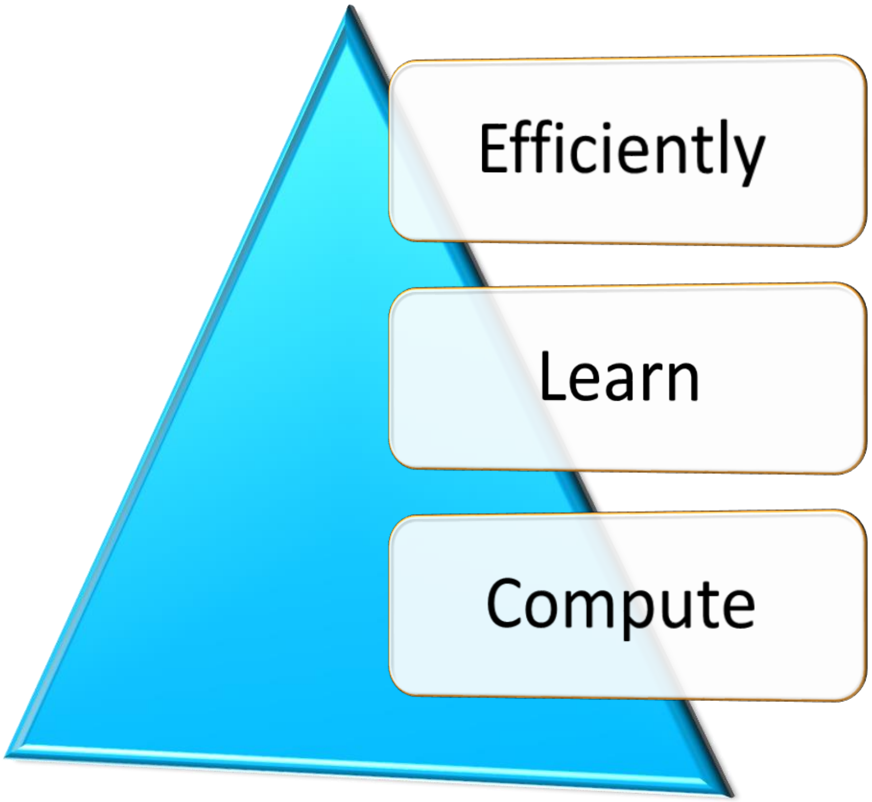
A Theoretical Comparison of Graph Neural Network Extensions

Pál András Papp [1] Roger Wattenhofer [1]





Base GNN
DropGNN
Ports
Rand IDs
Easier Learning
More Expressivity



Efficiently
Learn
Compute

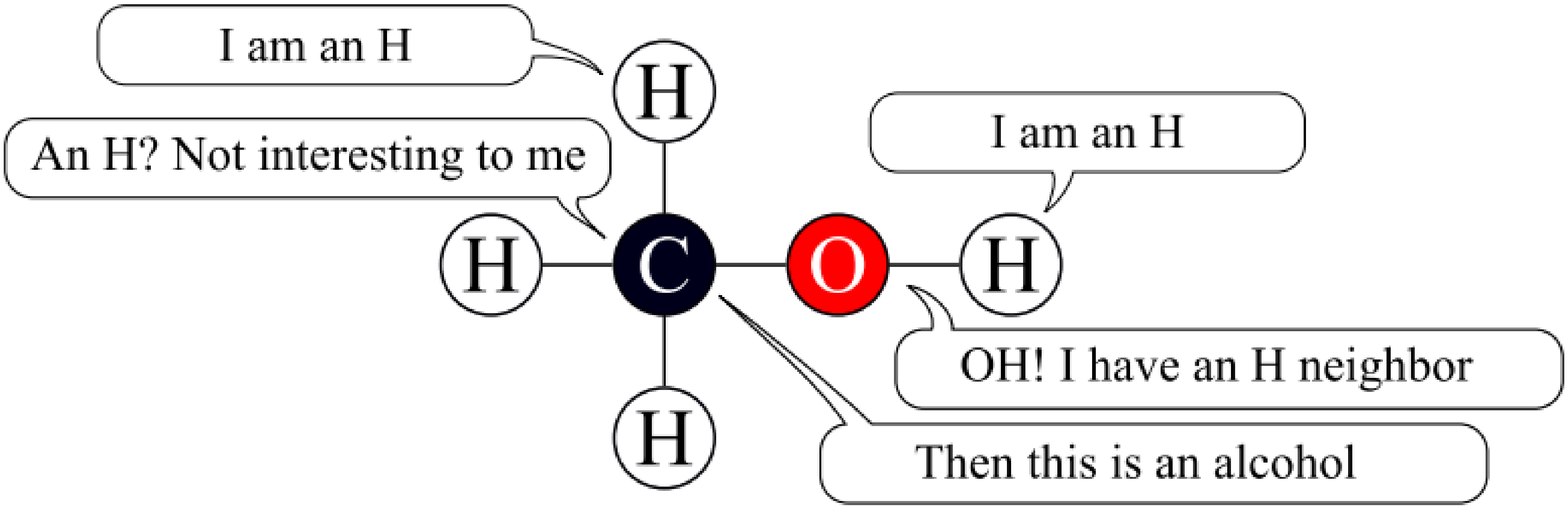
Extrapolation

Without Aggregation?

GwAC: GNNs with Asynchronous Communication

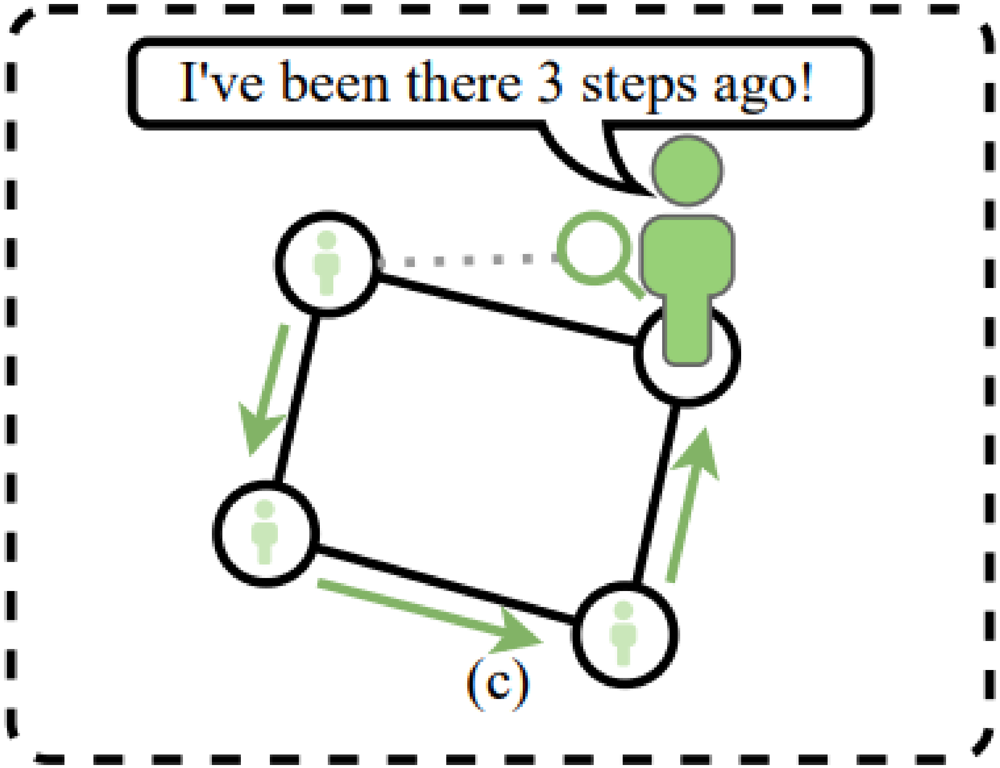
Lukas Faber
ETH Zurich, Switzerland
lfaber@ethz.ch

Roger Wattenhofer
ETH Zurich, Switzerland
wattenhofer@ethz.ch



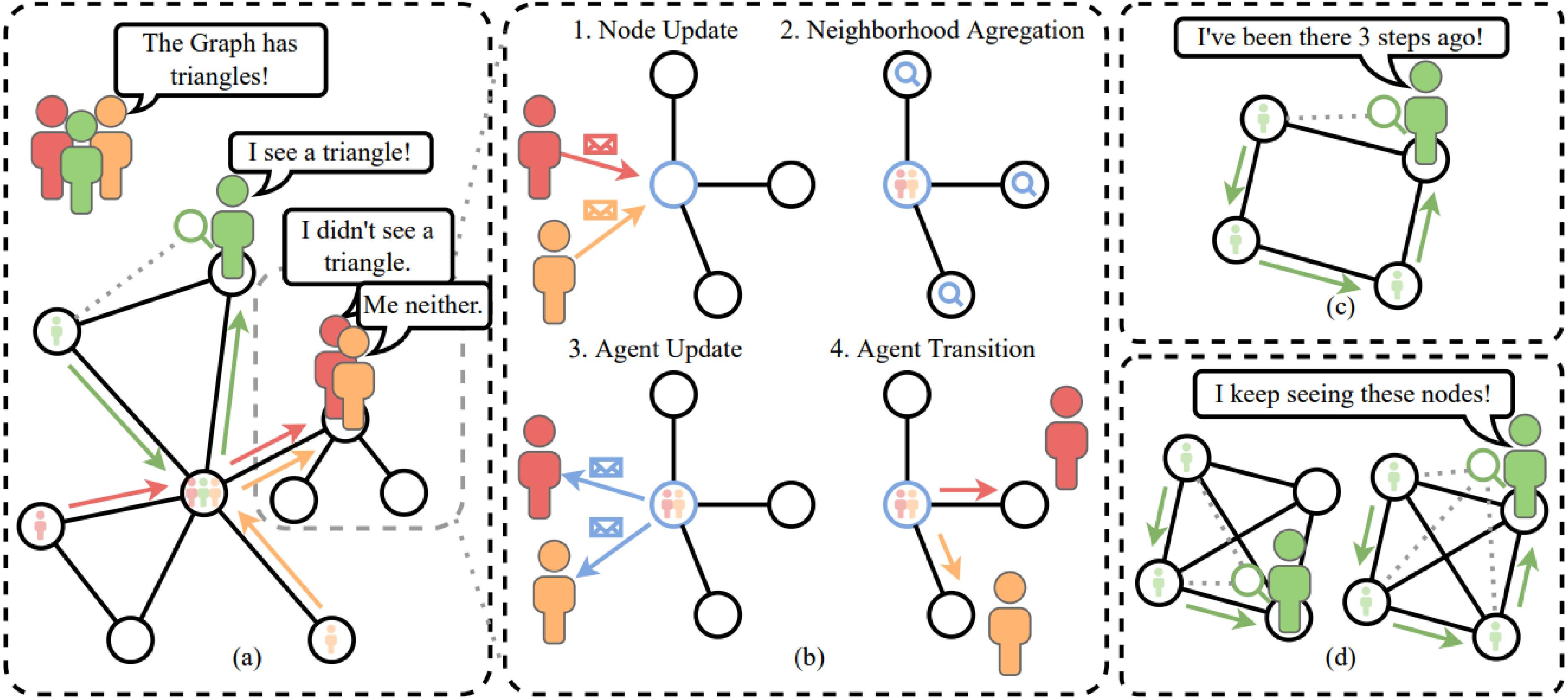
AGENT-BASED GRAPH NEURAL NETWORKS

Karolis Martinkus[1], **Pál András Papp**[2], **Benedikt Schesch**[1], **Roger Wattenhofer**[1]
[1]ETH Zurich [2]Computing Systems Lab, Huawei Zurich Research Center

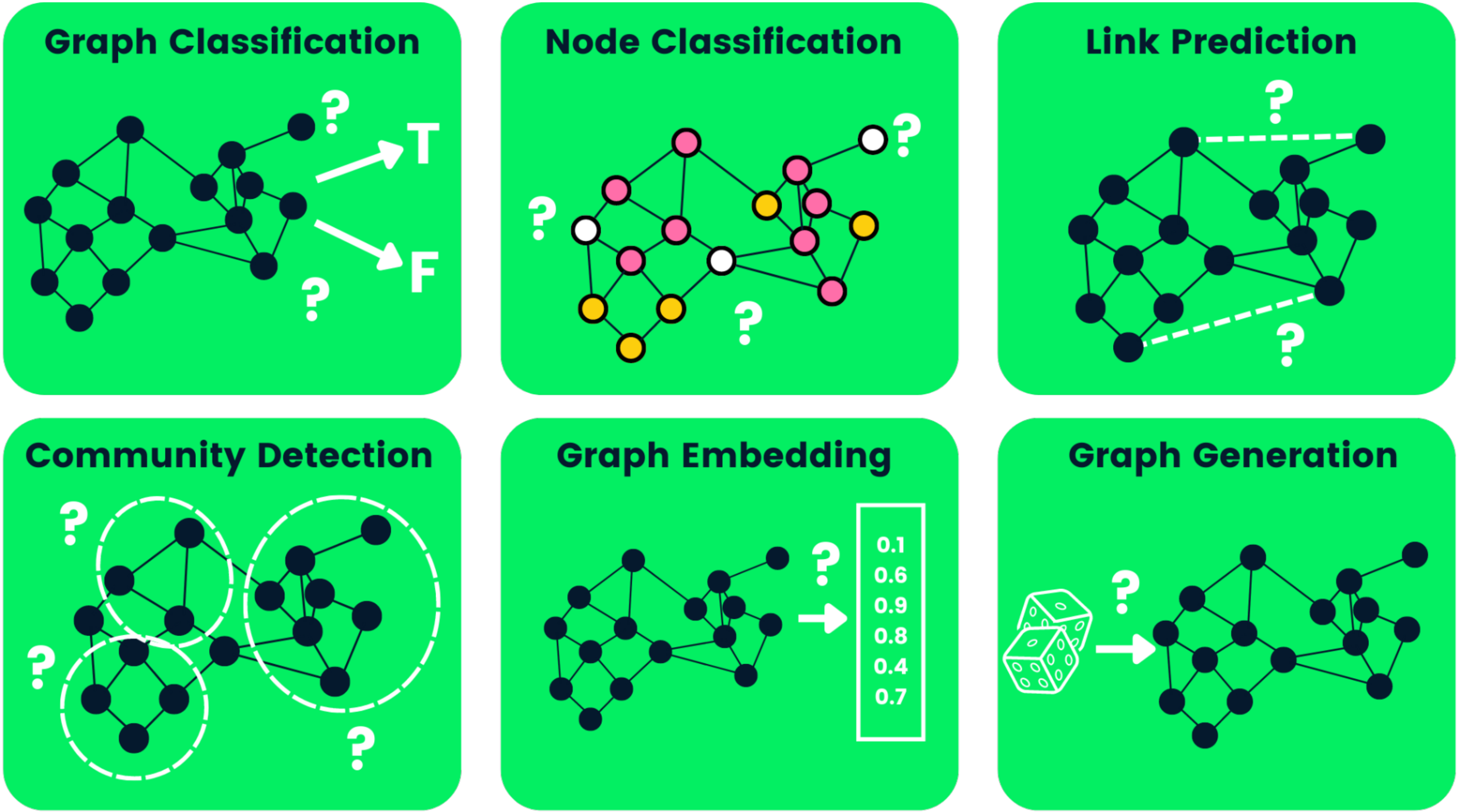


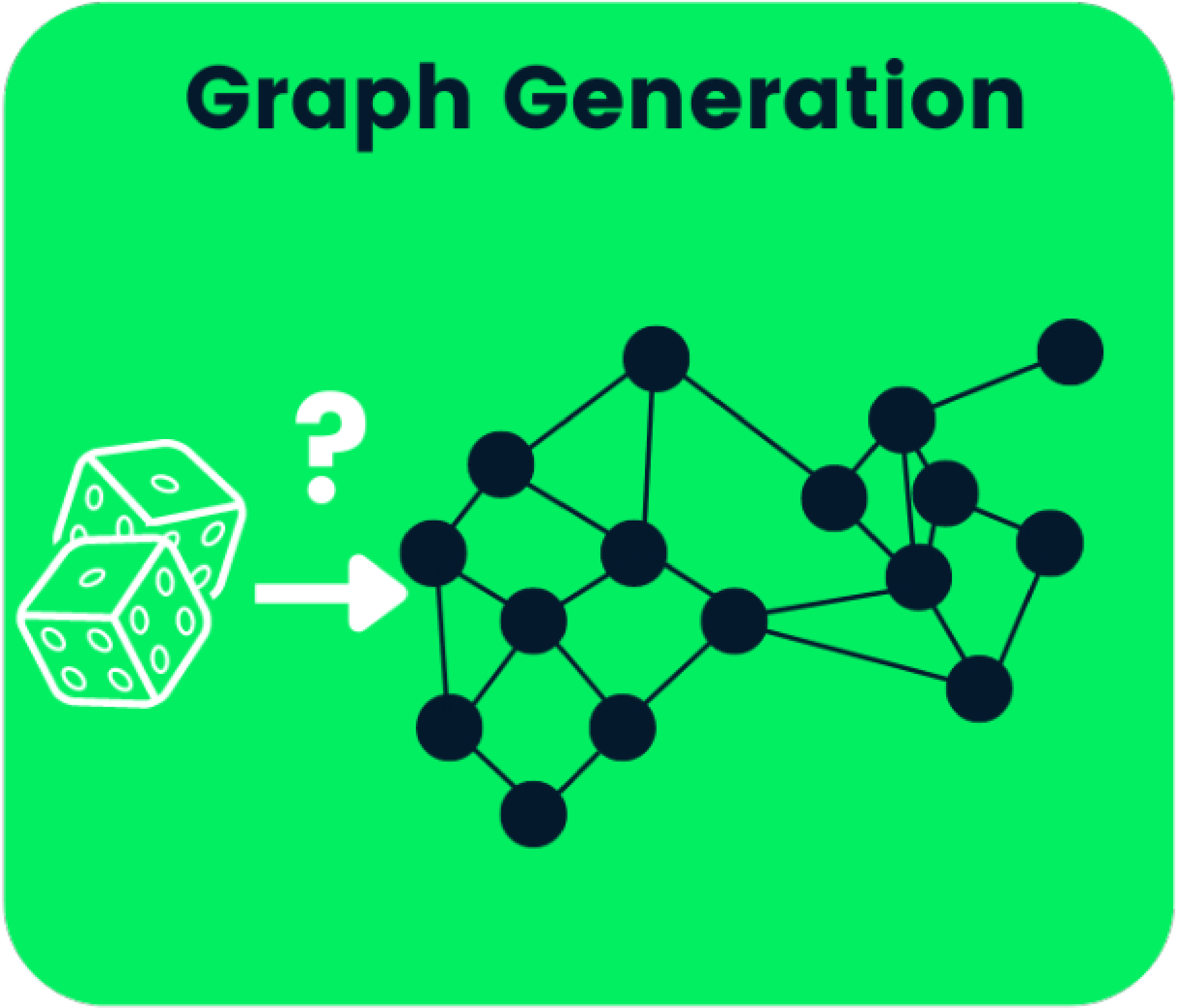
Agent-based Graph Neural Networks

Karolis Martinkus[1], Pál András Papp[2], Benedikt Schesch[1], Roger Wattenhofer[1]



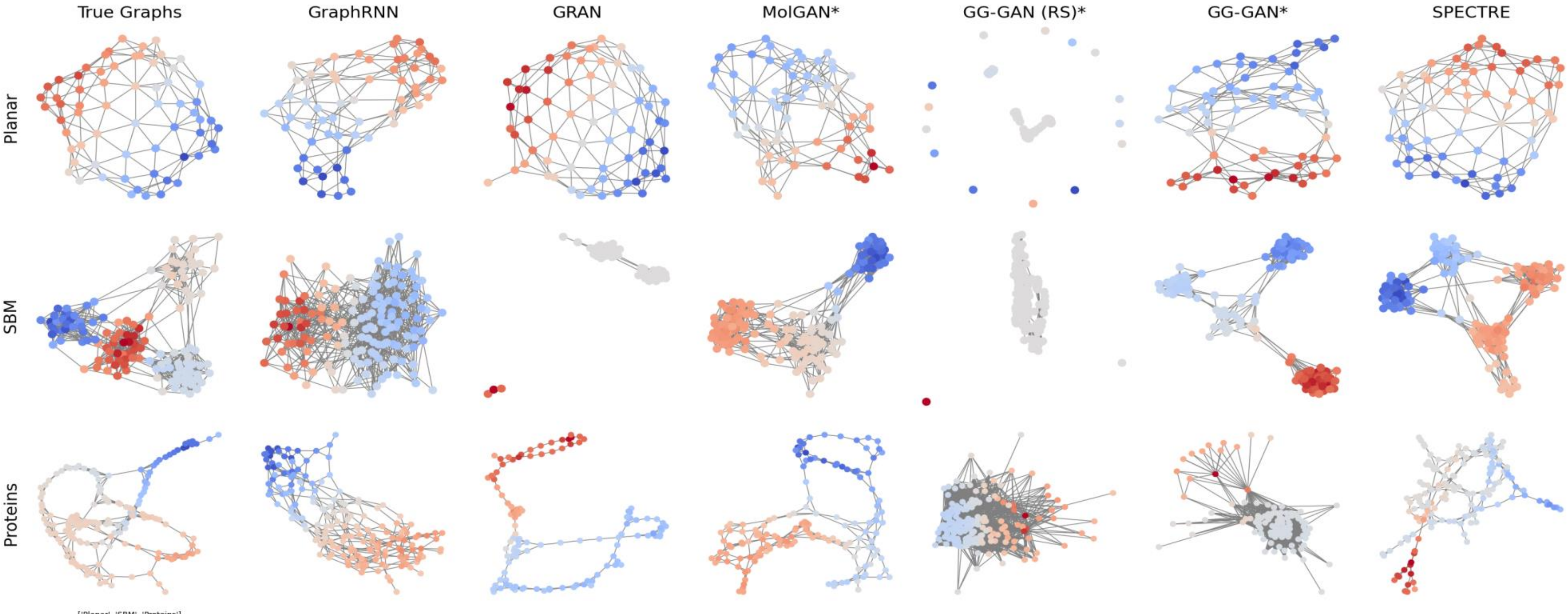
Model	4-CYCLES [59]	CIRCULAR SKIP LINKS [15]	2-WL
GIN [75]	50.0 ±0.0	10.0 ±0.0	50.0 ±0.0
GIN with random features [64; 1]	99.7 ±0.4	95.8 ±2.1	92.4 ±1.6
SMP [71]	**100.0 ±0.0**	**100.0 ±0.0**	50.0 ±0.0
DROPGIN [59]	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**
ESAN [8]	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0***
1-2-3 GNN [53]	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0†**
PPGN [51]	**100.0 ±0.0**	**100.0 ±0.0**	50.0 ±0.0
CRAWL [67]	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**
RANDOM WALK AGENTNET	**100.0 ±0.0**	**100.0 ±0.0**	50.5 ±4.5
SIMPLIFIED AGENTNET	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**
AGENTNET	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**


Graph Classification
?
T
F
?
Node Classification
?
?
?
Link Prediction
?
?
Community Detection
?
?
?
Graph Embedding
?
0.1
0.6
0.9
0.8
0.4
0.7
Graph Generation
?


Graph Generation
?

SPECTRE : Spectral Conditioning Helps to Overcome the Expressivity Limits of One-shot Graph Generators

Karolis Martinkus [1] **Andreas Loukas** [* 2] **Nathanaël Perraudin** [* 3] **Roger Wattenhofer** [1]

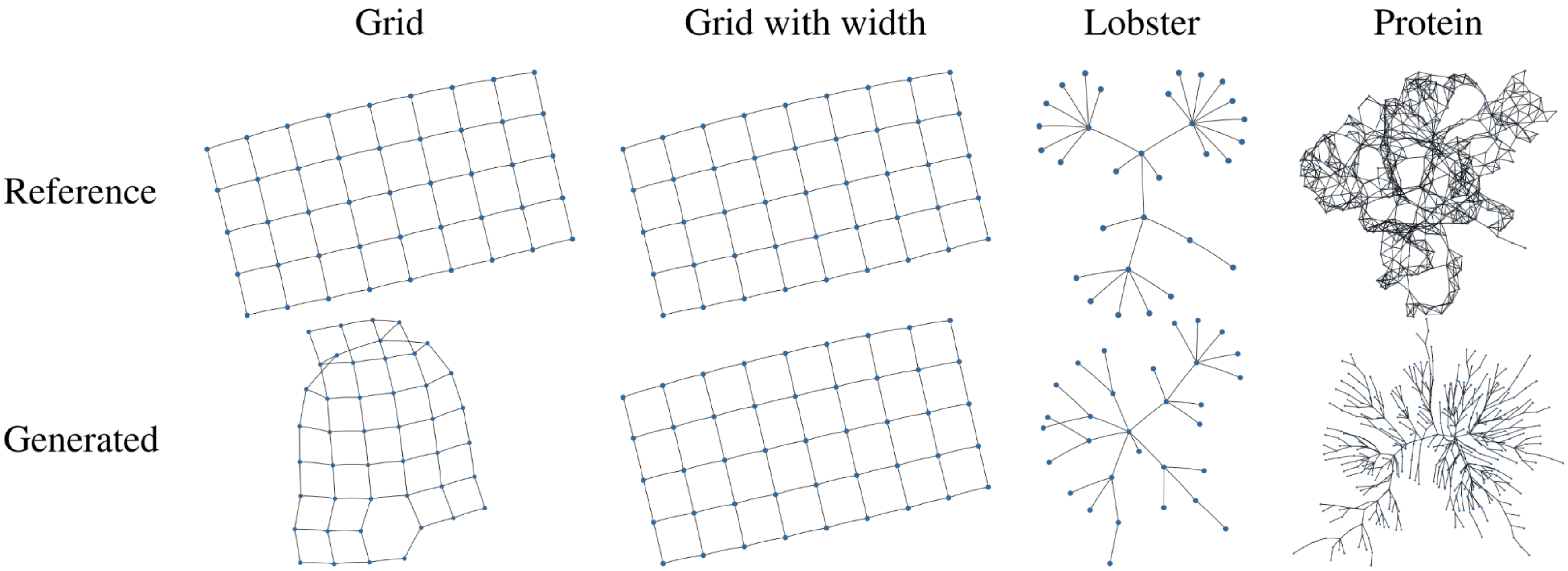


DISCOVERING GRAPH GENERATION ALGORITHMS

Mihai Babiac, Karolis Martinkus & Roger Wattenhofer
ETH Zurich
{mbabiac,martinkus,wattenhofer}@ethz.ch

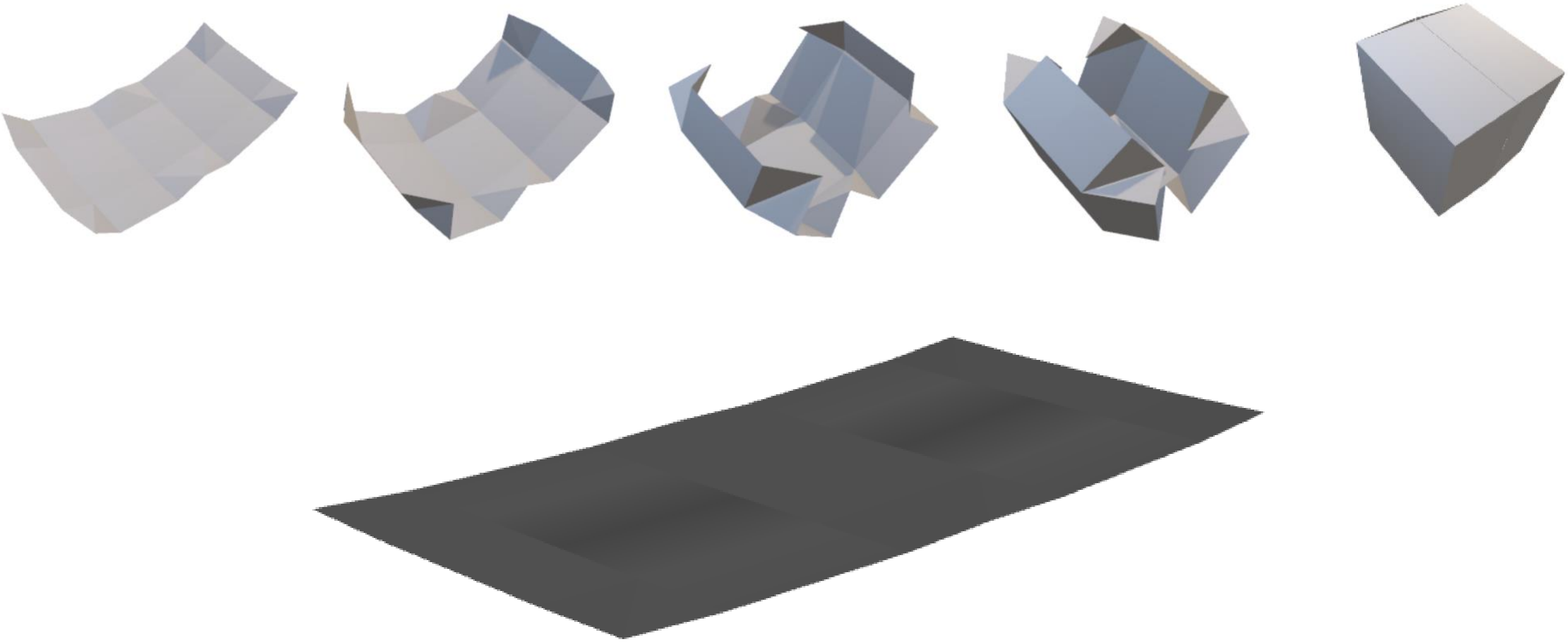
```
def outer_loop():
    for i in range(N):
        inner_loop()

def inner_loop():
    for j in range(i):
        float00 = random(0, 1)
        bool00 = float00 < 0.4
        if bool00:
            add_edge(i, j)

outer_loop()
```

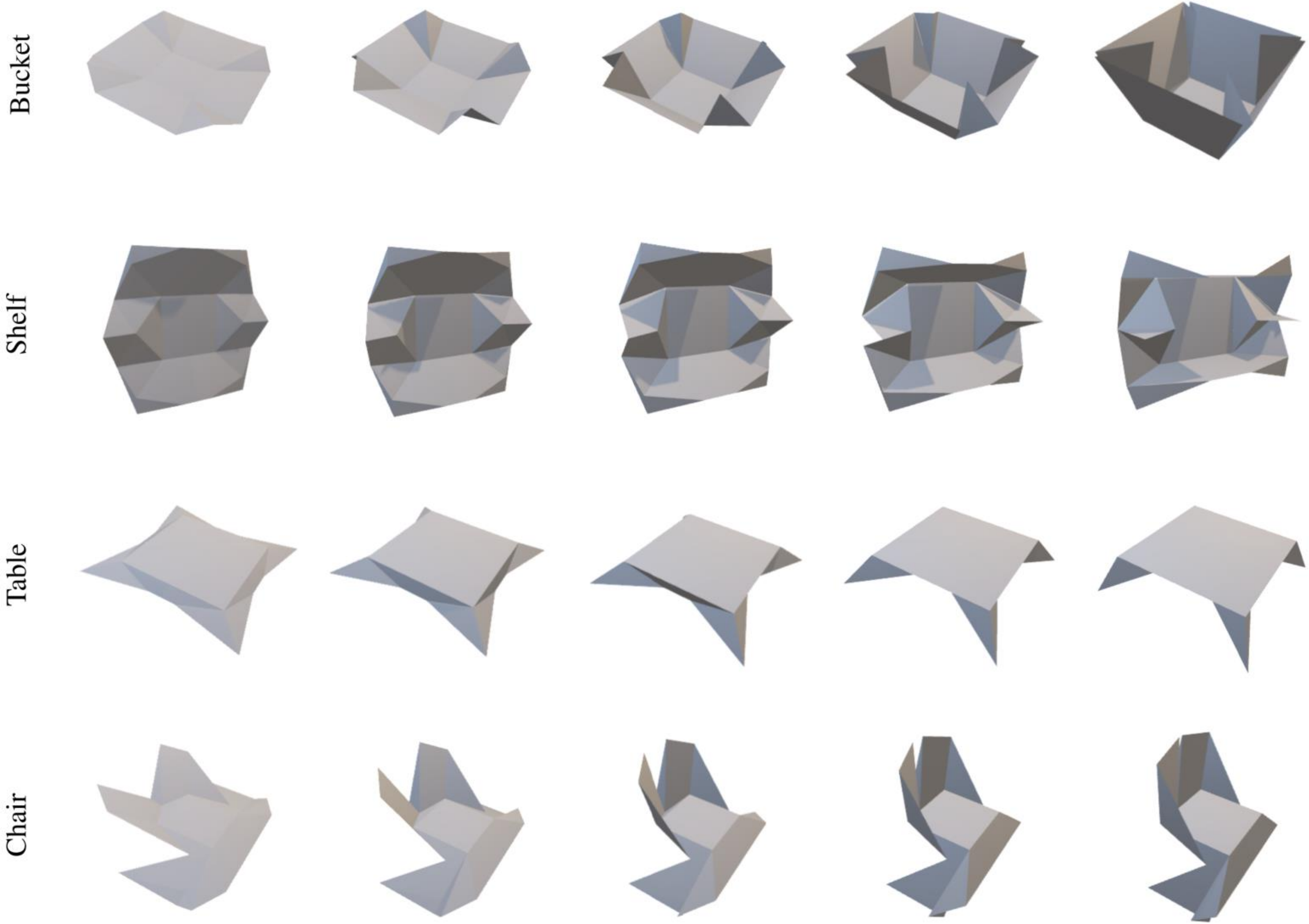
```
def outer_loop():
    for i in range(N):
        int00 = i + n
        add_edge(i, int00)

        int01 = i % n
        bool00 = int01 == 0
        if not bool00:
            int01 = i + 1
            add_edge(i, int01)

outer_loop()
```


Grid
Grid with width
Lobster
Protein
Reference
Generated

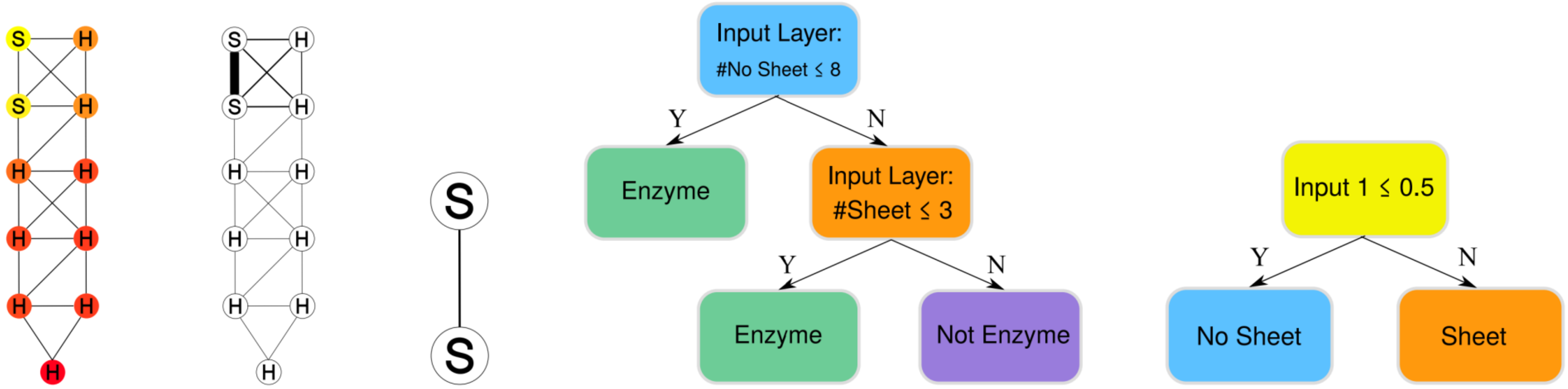
Automating Rigid Origami Design

Jeremia Geiger, Karolis Martinkus, Oliver Richter, Roger Wattenhofer



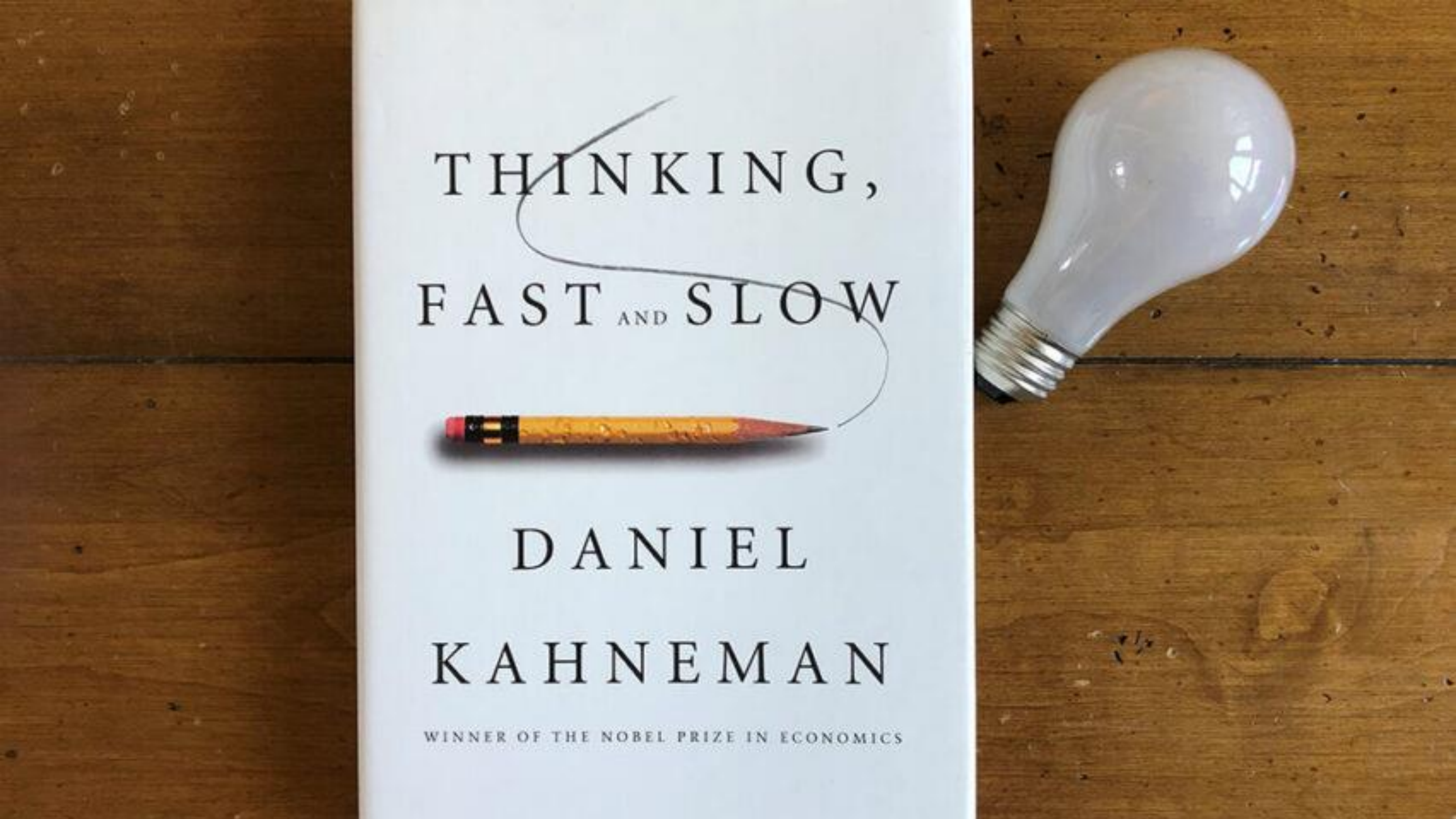

Bucket
Shelf
Table
Chair

Graph Explanation

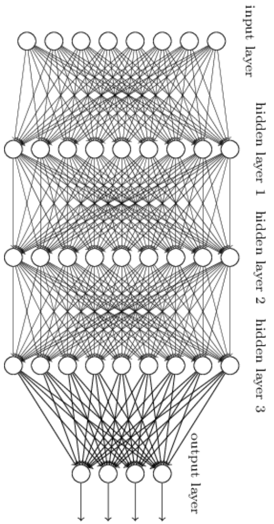

S
H
S
H
H
H
H
H
H
H
H
S
H
S
H
H
H
H
H
H
H
H
S
S
Input Layer:
#No Sheet ≤ 8
Y
N
Enzyme
Input Layer:
#Sheet ≤ 3
Y
N
Enzyme
Not Enzyme
Input 1 ≤ 0.5
Y
N
No Sheet
Sheet

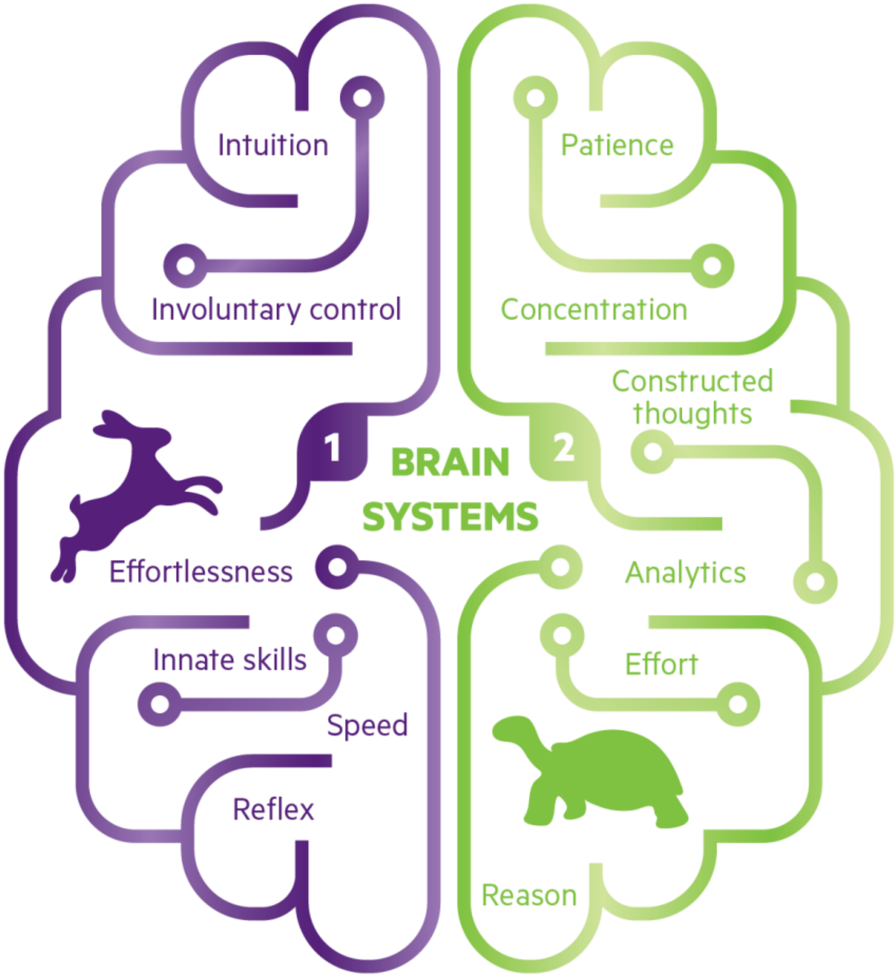
(a) (b) (c) (d) (e)

The Bigger Picture


THINKING,
FAST AND SLOW
DANIEL
KAHNEMAN
WINNER OF THE NOBEL PRIZE IN ECONOMICS

Deep Neural Networks





?

User

what is 5 times 7?

Assistant

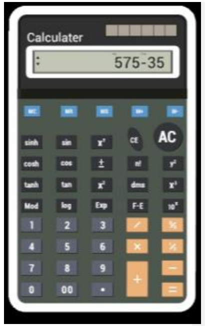
5 times 7 equals 35.

User

what is 384903720 times 538982734?

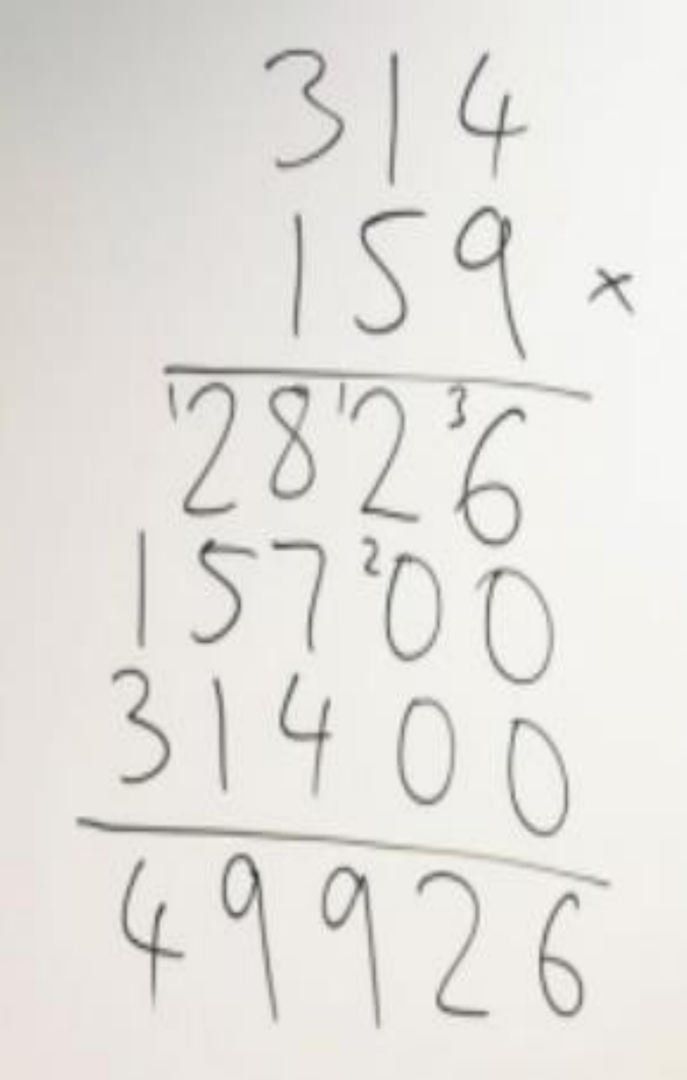
Assistant

384,903,720 times 538,982,734 equals 207,559,081,508,668,480.



$$384903720 \times 538982734 =$$

207,456,459,332,370,480



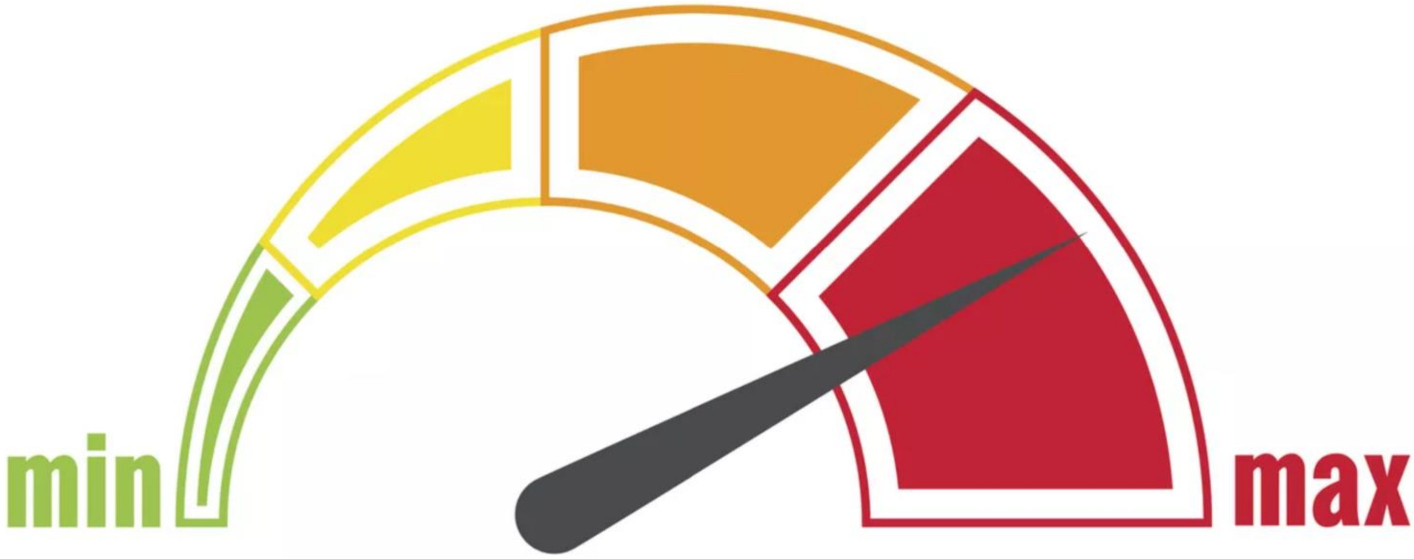
User

what is 384903720 times 538982734?

Assistant

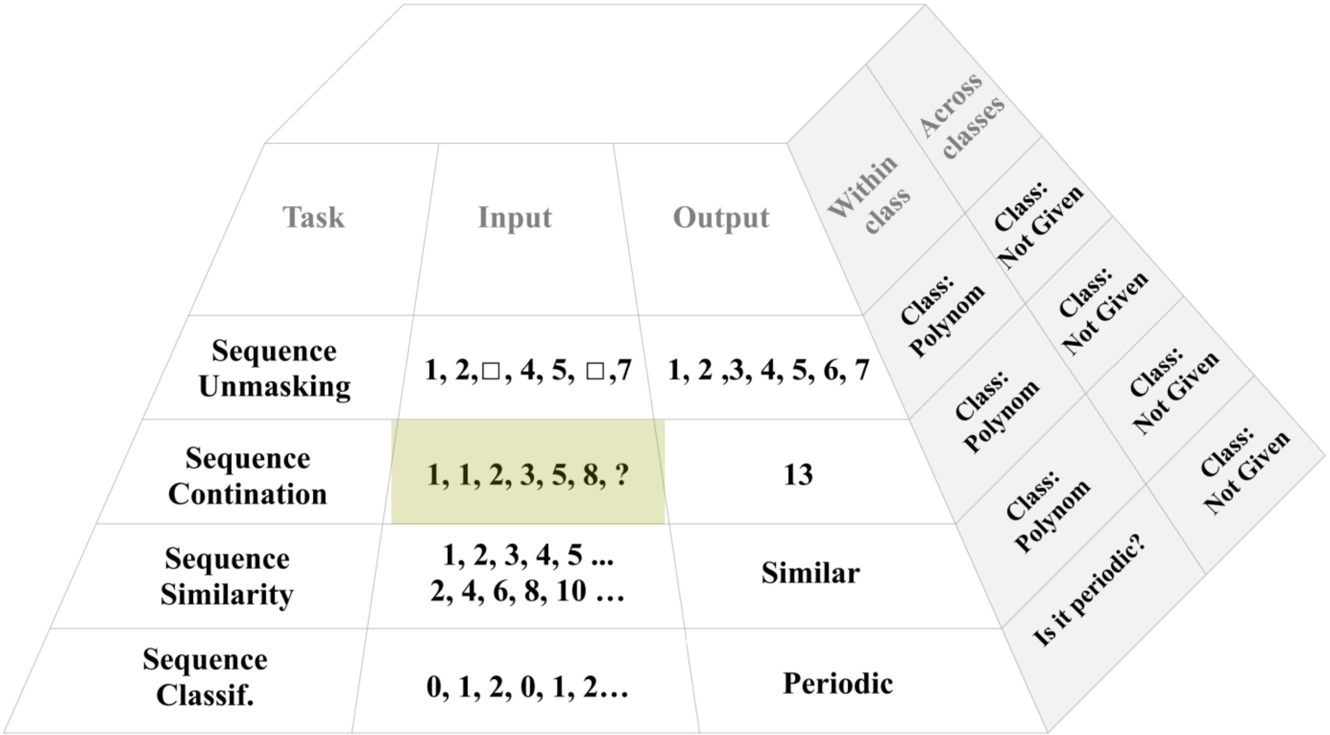
384,903,720 times 538,982,734 equals 207,559,081,508,668,480.

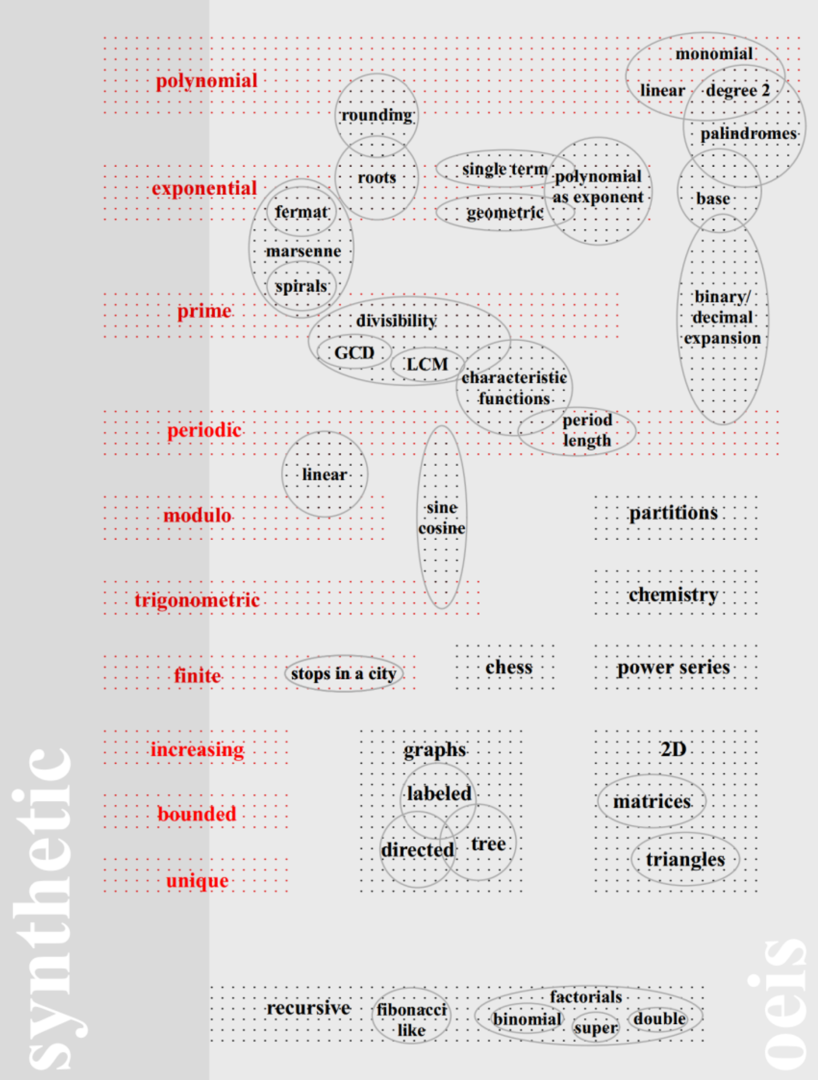
Thinking Slow Benchmark?



FACT: Learning Governing Abstractions Behind Integer Sequences

Peter Belcák, Ard Kastrati, Flavio Schenker, Roger Wattenhofer




synthetic
oeis
polynomial
exponential
prime
periodic
modulo
trigonometric
finite
increasing
bounded
unique
monomial
linear
degree 2
palindromes
rounding
roots
single term
geometric
polynomial as exponent
base
fermat
marsenne
spirals
divisibility
GCD
LCM
characteristic functions
binary/ decimal expansion
period length
linear
sine cosine
partitions
chemistry
stops in a city
chess
power series
graphs
labeled
directed
tree
2D
matrices
triangles
recursive
fibonacci like
factorials
binomial
super
double

Simon Tatham's Portable Puzzle Collection

Simon Tatham's Puzzles 4+

Greg Hewgill

Designed for iPad

★★★★★ 4.8 • 171 Ratings

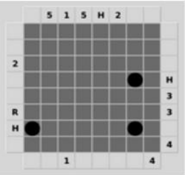
Free

Simon Tatham's Puzzles

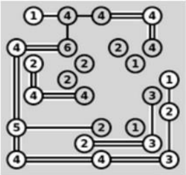
Chris Boyle

4.8★	500K+	E
14.5K reviews	Downloads	Everyone ⓘ

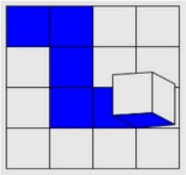
Black Box



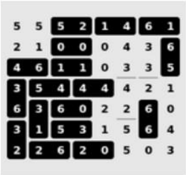
Bridges



Cube

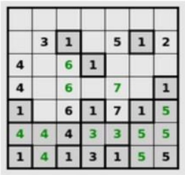


Dominosa

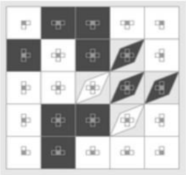


Fifteen

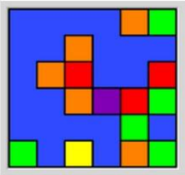
Filling



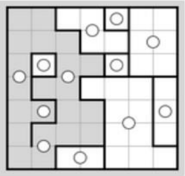
Flip



Flood

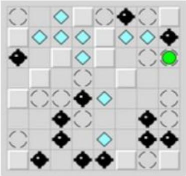


Galaxies



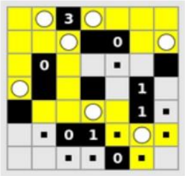
Guess

Inertia

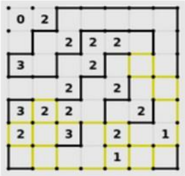


Keen

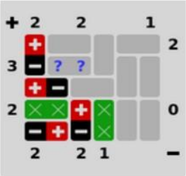
Light Up



Loopy



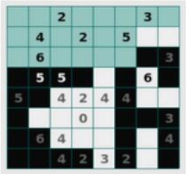
Magnets



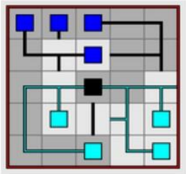
Map

Mines

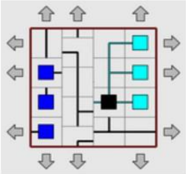
Mosaic



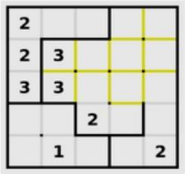
Net



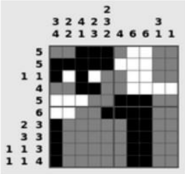
Netslide



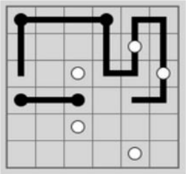
Palisade



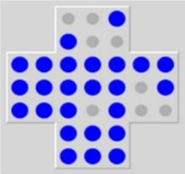
Pattern



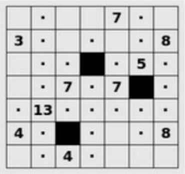
Pearl



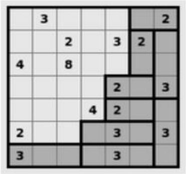
Pegs



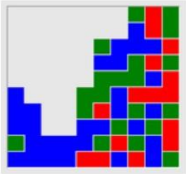
Range



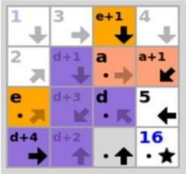
Rectangles



Same Game

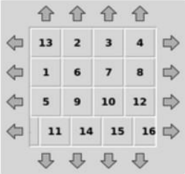


Signpost

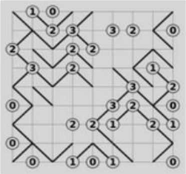


Singles

Sixteen

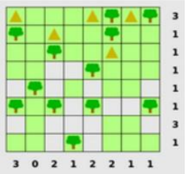


Slant

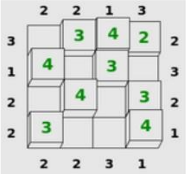


Solo

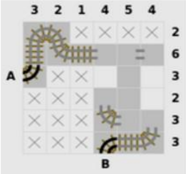
Tents



Towers

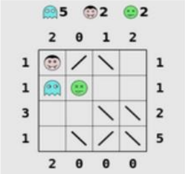


Tracks

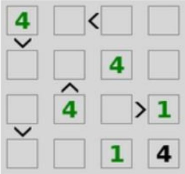


Twiddle

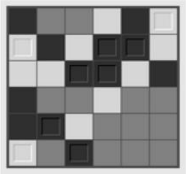
Undead



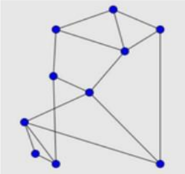
Unequal



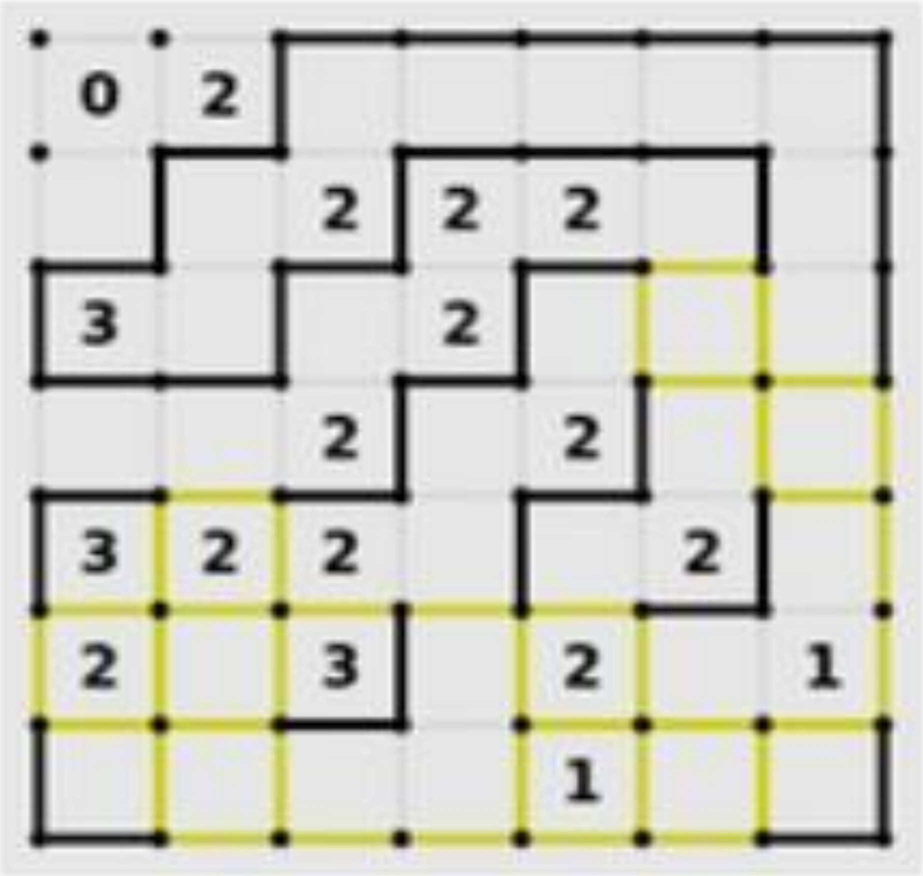
Unruly



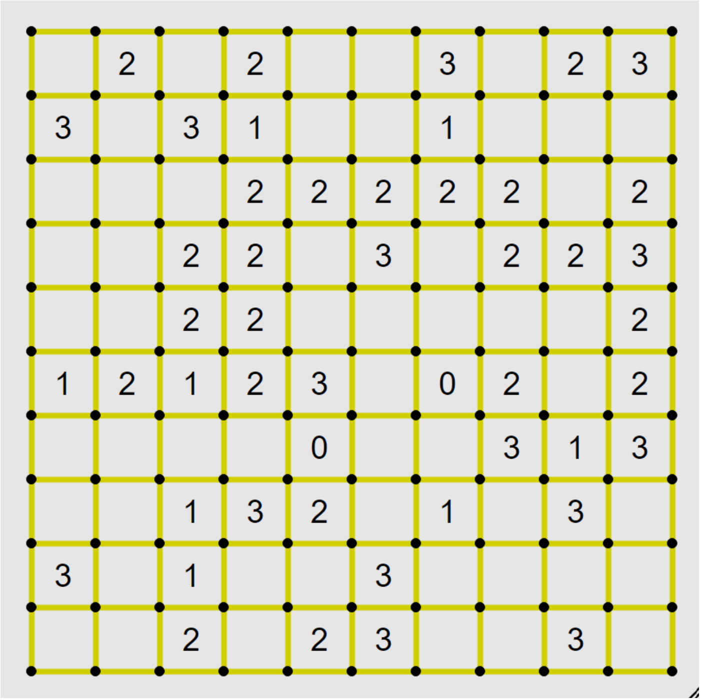
Untangle

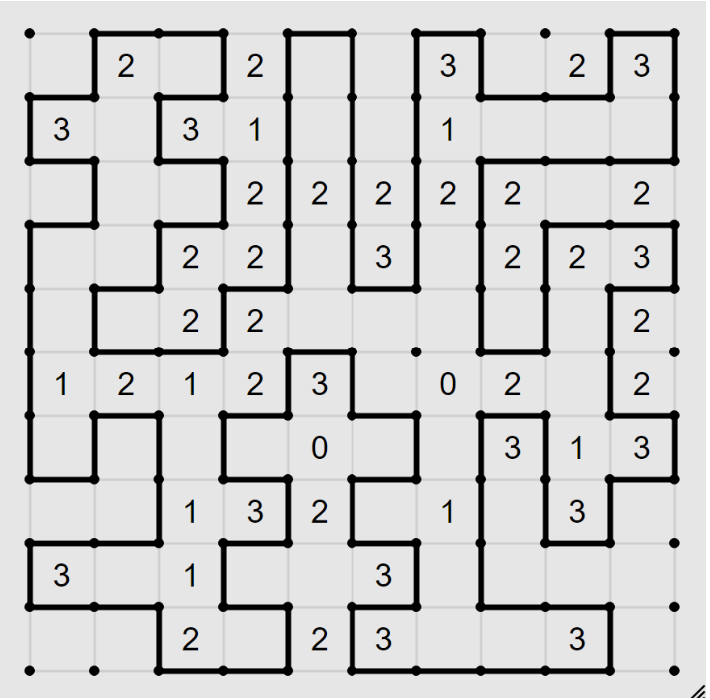


Loopy

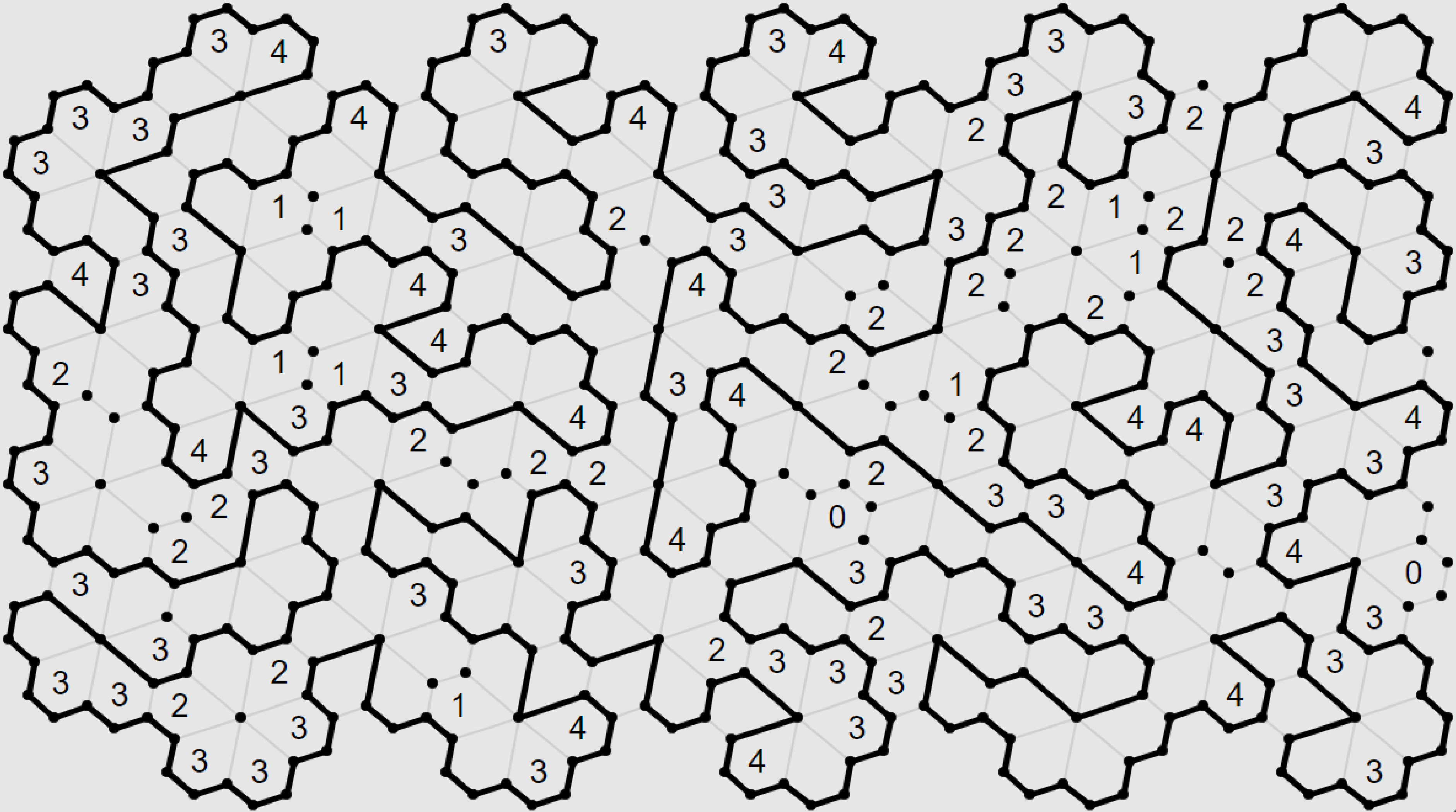


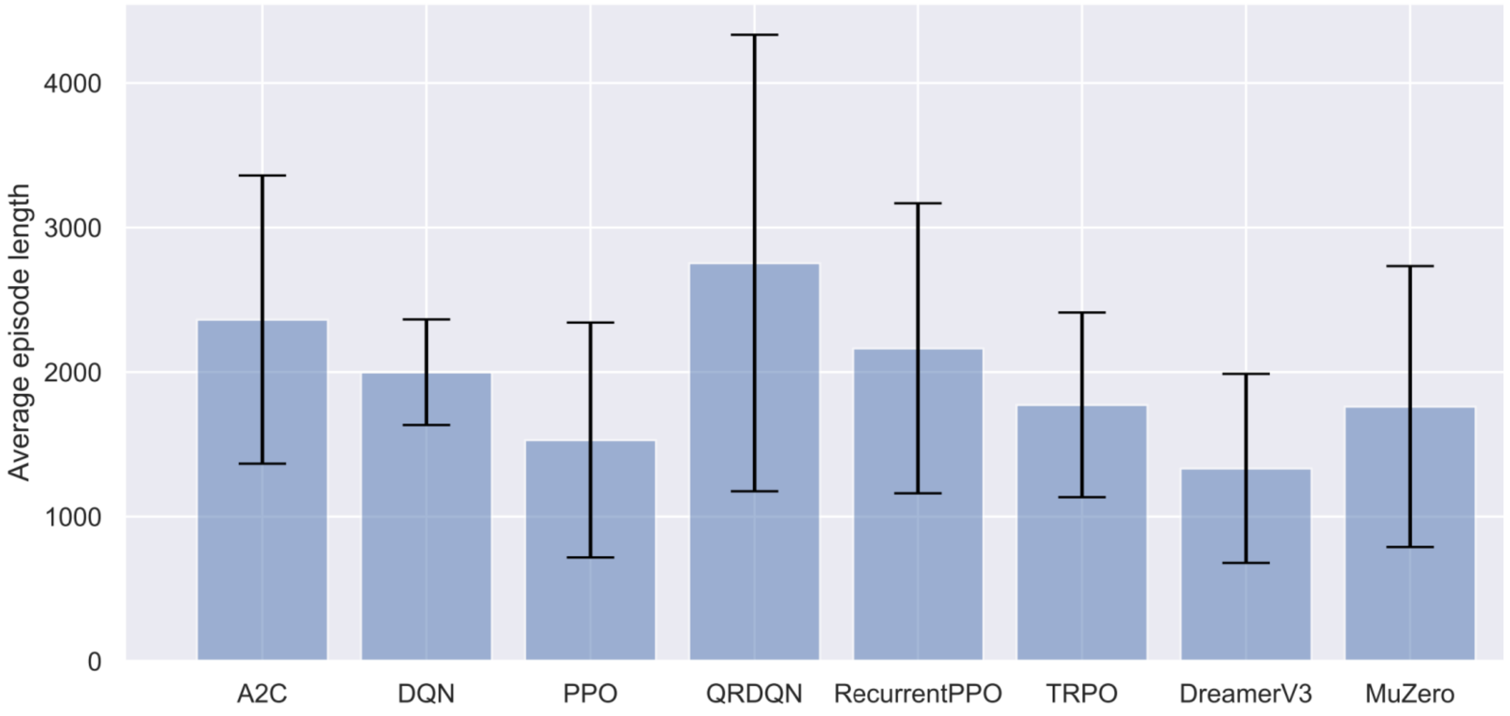
Loopy (Takegaki, Slitherlink, Ouroboros, Suriza, …)



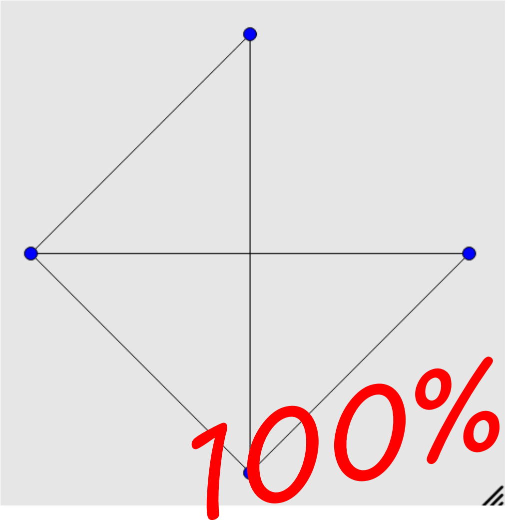


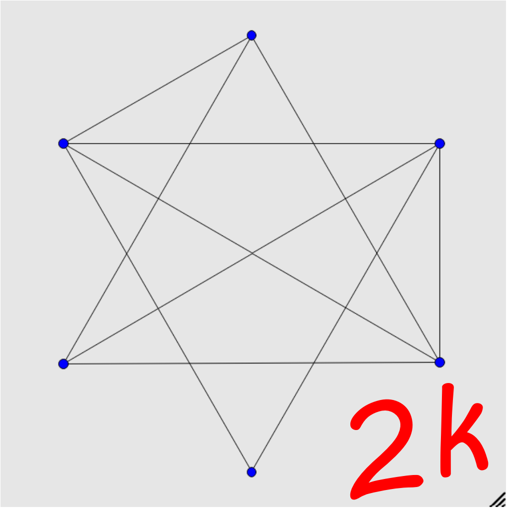
				2	2	0	2	3	1	1	2			1	2				2	1	2					0	1		2	3	1			1		0	2
		3		1	1		2				2	2	3	0			3									1				2			1	1			
3	2						2	2								2		2	3	1	2						2	0	3				2	3			
		2			1	1		1	1	1	0		2		2					1			2		1				2	1				2		1	3
	2	3		0		1	3		3		0					3			3		3	0	1		1	0	2				3	1	1	1			1
						2		2	2		1	2		2					2			1	1	2			2		3		1		3	3	3		
2	1	2					2	2	3			3			1					3		2					1		3				0				
	2	2	2		3		1		2			2													1			1	3				1		3		3
1			3			2					2	3		2	2	3	2	1	3		1		2	1	2	3	3			0				3	1		
				3				2	1	2					1		3	2		2	2						2	0		3	3						1
1	2	3	1	3					2	1		2			3				1	1		3	3	2	1		2							1	3		
				3	2		2	1		2	1	1	1	2		3		2	0	2	1		1		2	3		3				1		3			
		2	2					2		3	2						2										3		2				1	2	3		1
	3		2	2			2	2	2					2								2		1		1	2			1	2	2	0			2	1
		1	2	1	1	2	2	2	2	3									3	0	3				3	2				0			1	2		2	
		1	3	1	2	2	2	2	1		1		2		1			2		3			3	2				2	3							2	1
	2					2	2		2			2	0		1				2	2	2				2		2		2		3	3	2			2	
1	3		1	1				1				2			0	2		1					3		1		1	3	2			1			1		3
1		0	1	0	0	1	2	2	2	1	1	1		2	1			0	1	2							1	2	1		1	3		3		0	
2	3			1	1	0						3	2	2	2	1		2	2	2			3			2		3		2	2		2	1		1	
	1						3				2	1				2				2		1				1		1	2			1			3	1	

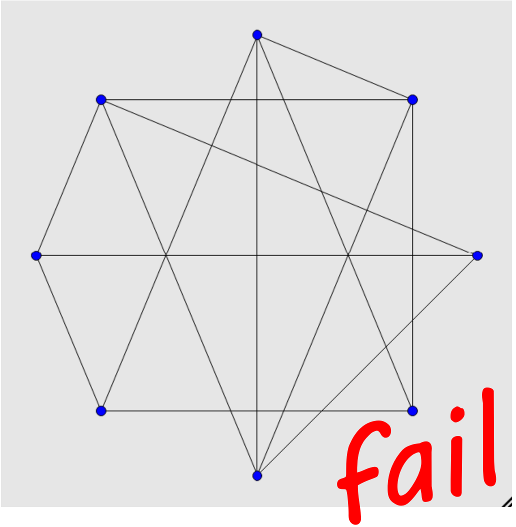



Average episode length
0
1000
2000
3000
4000
A2C
DQN
PPO
QRDQN
RecurrentPPO
TRPO
DreamerV3
MuZero

Puzzle	Parameters	PPO		DreamerV3	
Netslide	`2x3b1`	35.3 ± 0.7	(100.0%)	12.0 ± 0.4	(100.0%)
	`3x3b1`	4742.1 ± 2960.1	(9.2%)	3586.5 ± 676.9	(22.4%)
Same Game	`2x3c3s2`	11.5 ± 0.1	(100.0%)	7.3 ± 0.2	(100.0%)
	`5x5c3s2`	1009.3 ± 1089.4	(30.5%)	527.0 ± 162.0	(30.2%)
Untangle	`4`	34.9 ± 10.8	(100.0%)	6.3 ± 0.4	(100.0%)
	`6`	2294.7 ± 2121.2	(96.2%)	1683.3 ± 73.7	(82.0%)







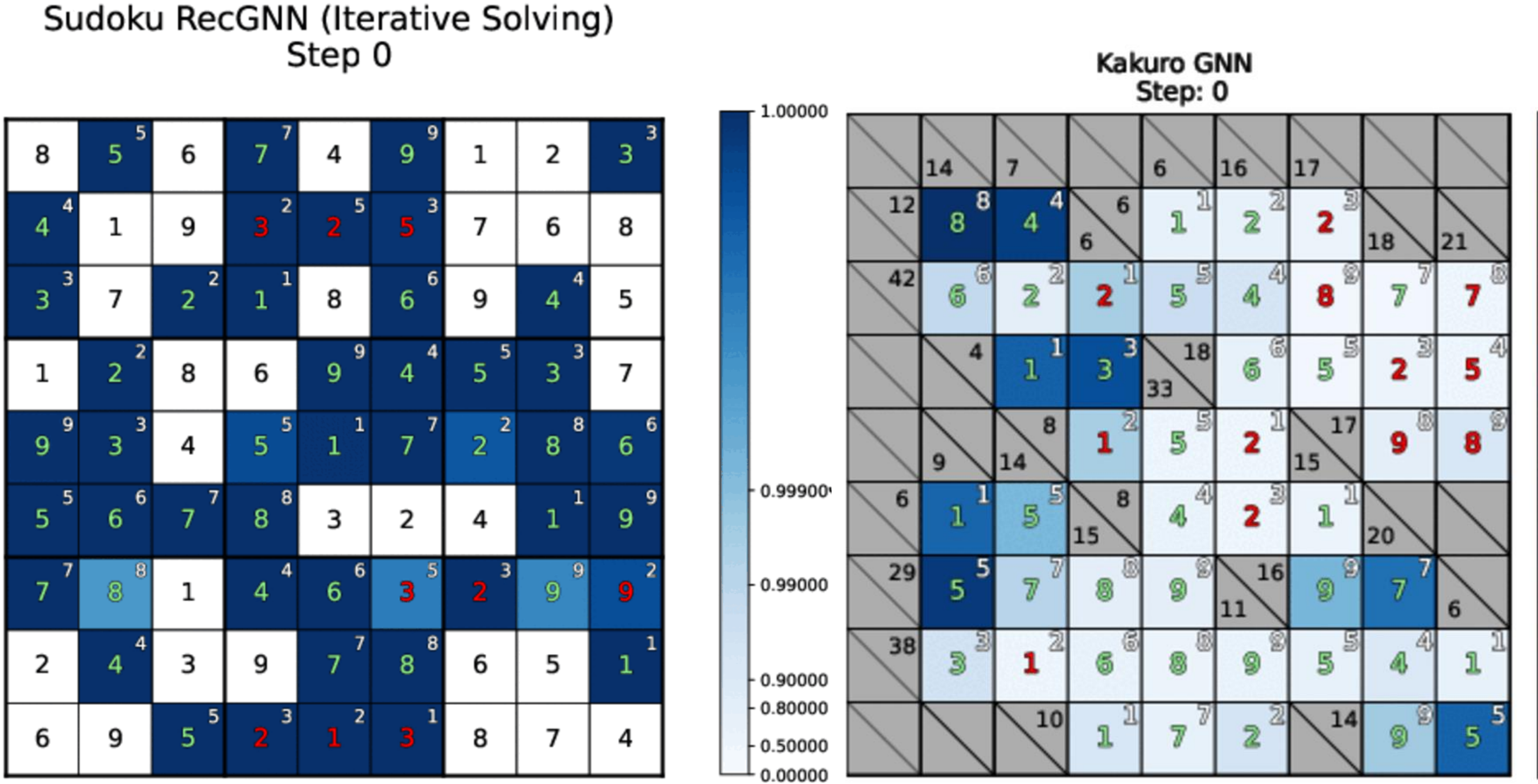
Sudoku

1		3	
	1		2

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

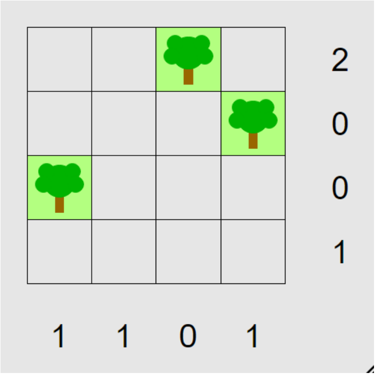
f		9		5			6						b	3	7
	6	b				1	f		7						8
				9		4						2	g		5
5	g	c	8			a	d	3		b		1			
		d	f			2				5		c			
9		4		c	5	g				d	1		e	7	
	b		5		9		3				a	6		f	4
								4	e				a	5	
	4	2				c	e								
a	d		e	1				9		4		f		8	
	5	g		f	2				d	6	8		7		9
			1		a				c			5	2		
			c		4		g	8	a			b	f	2	1
g		a	d						5		6				
7						9		e	b				6	d	
8	1	6						c			4		5		g

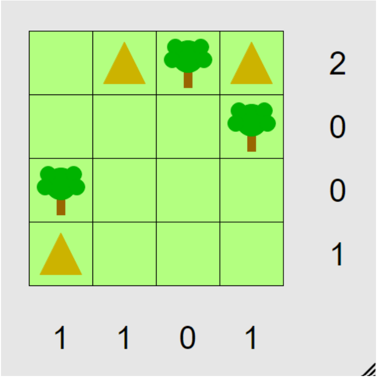
	4	6			2			
	3					7		
7		2		9	8			
		5					2	
	8		5		6		3	
	2					5		
			7	1		2	4	6
		9					5	
			4			8	9	

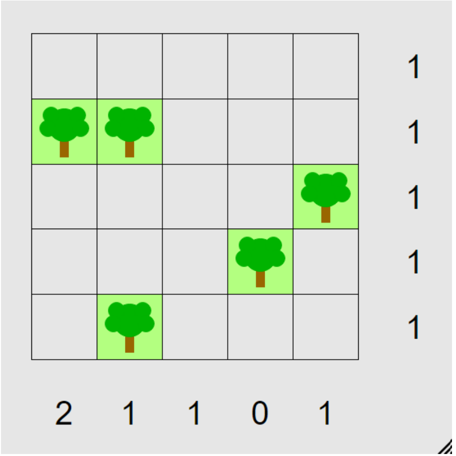


Sudoku RecGNN (Iterative Solving)
Step 0
Kakuro GNN
Step: 0
1.00000
0.99900
0.99000
0.90000
0.80000
0.50000
0.00000

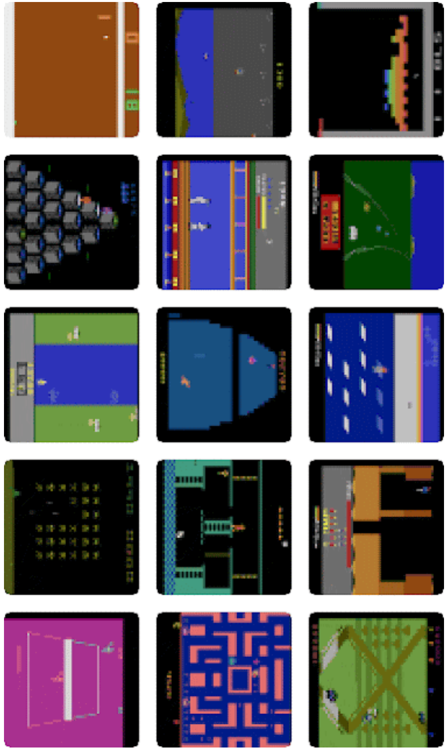
Tents

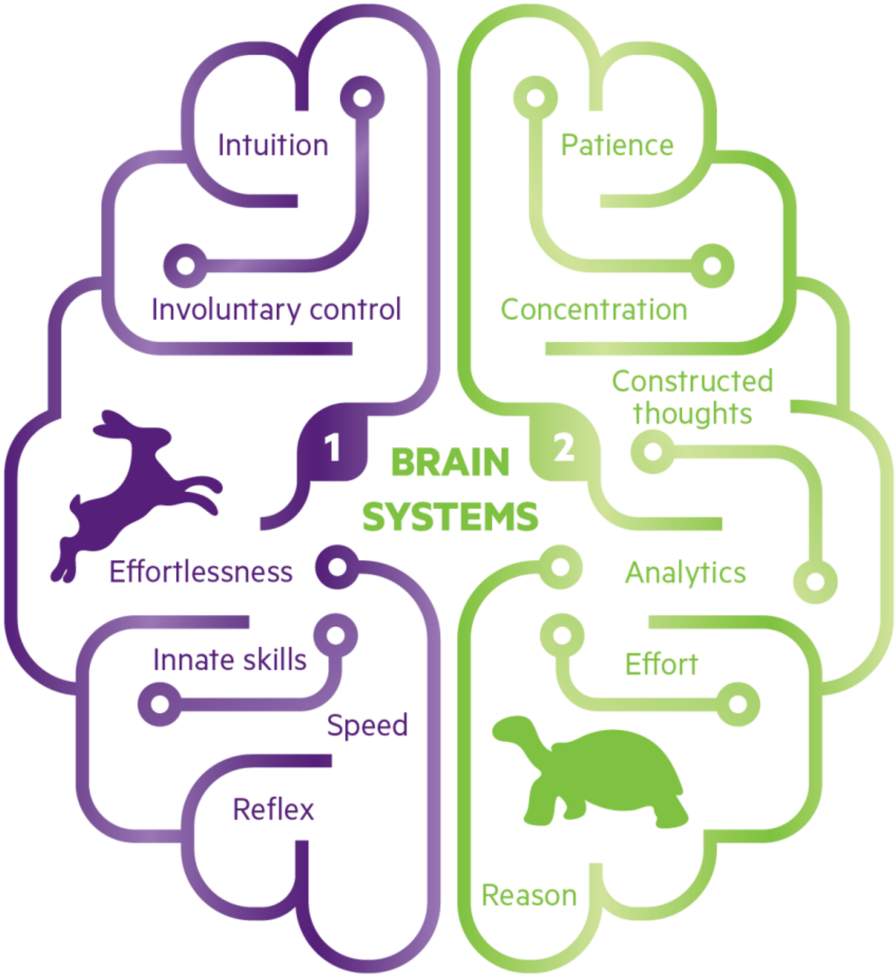




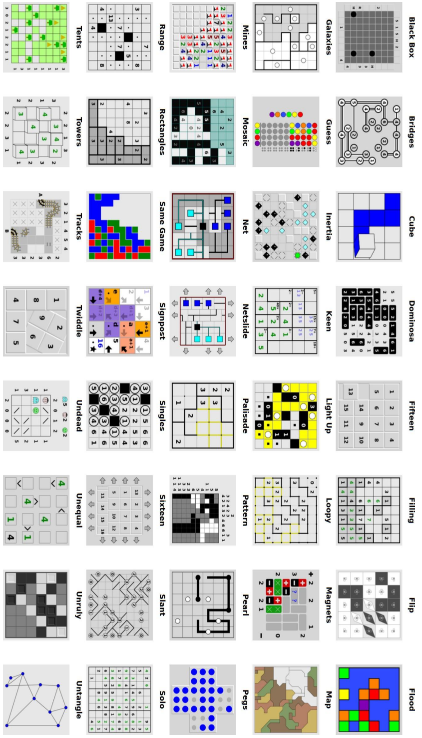


Atari Games

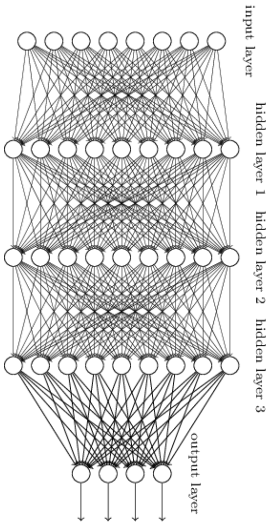


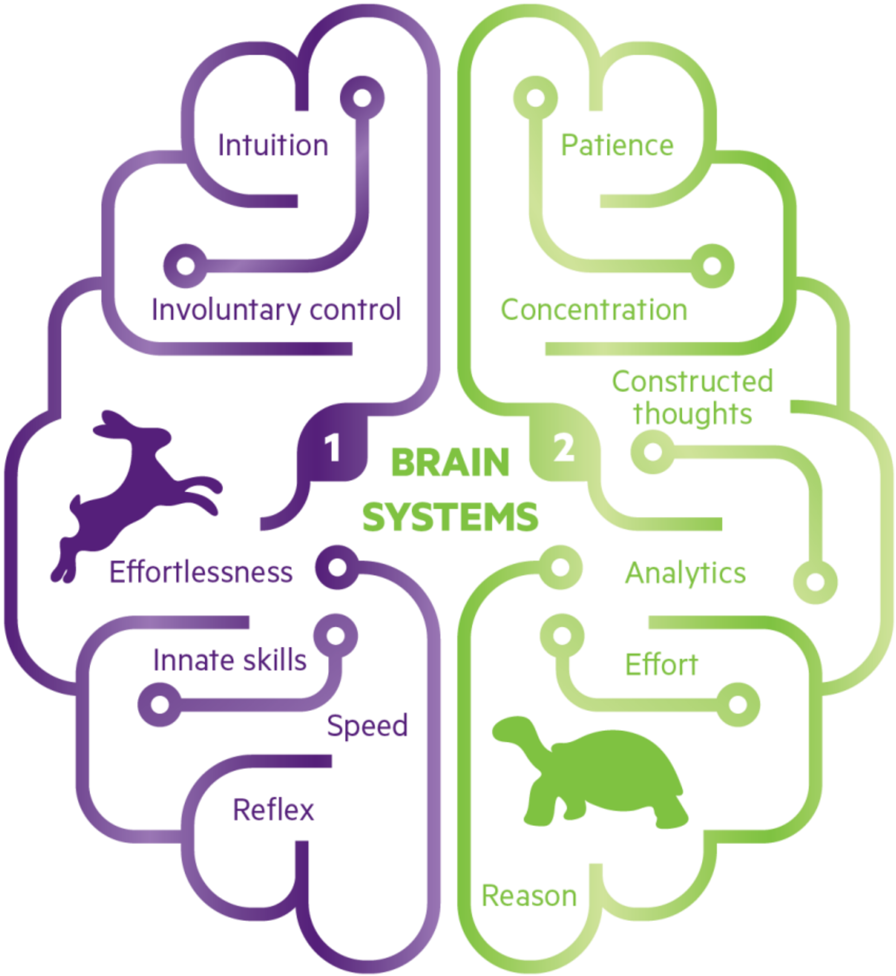


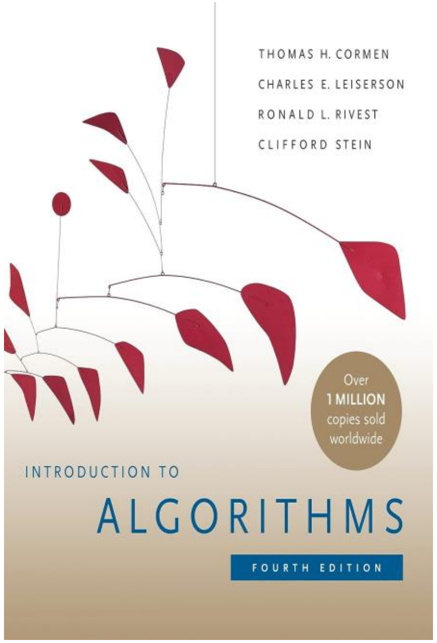
Puzzle Games



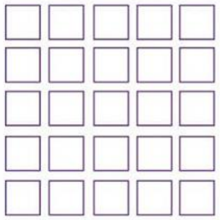
Deep Neural Networks

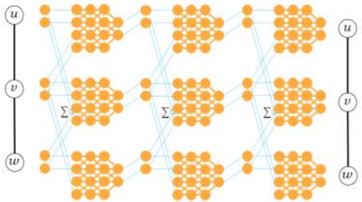


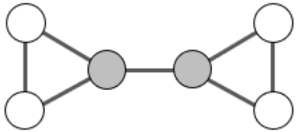


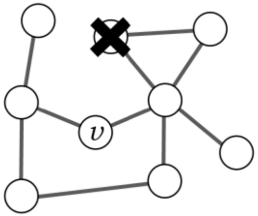


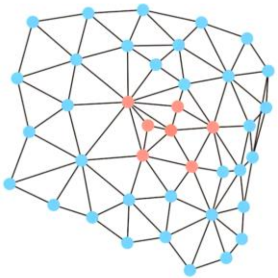
Summary

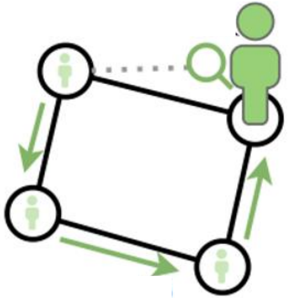


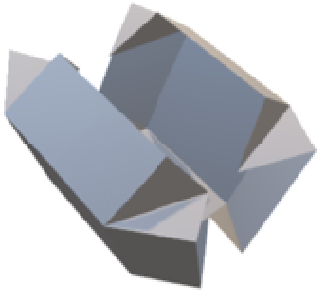

u
v
w
Σ
Σ
Σ
u
v
w

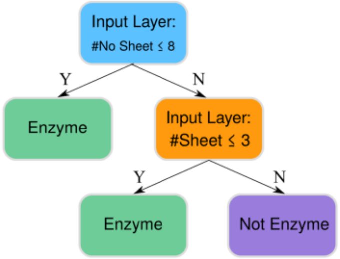



v








Input Layer:
#No Sheet ≤ 8
Y
N
Enzyme
Input Layer:
#Sheet ≤ 3
Y
N
Enzyme
Not Enzyme

Thank You!

Any questions or comments?

Thanks to co-authors: Peter Belcak, Benjamin Estermann, Lukas Faber, Florian Grötschla, Luca Lanzendörfer, Karolis Martinkus, Joël Mathys, Pal Andras Papp, etc.

Roger Wattenhofer