ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Model Agnostic Solution of CSPs via Deep Learning: a Preliminary Study

Andrea Galassi
Michele Lombardi
Paola Mello
Michela Milano

It All Started with a Question

Can a Deep Neural Network learn to solve a combinatorial problem?

A blackbox view of a CSP: $\exists x \in \mathbb{N}^n \mid f(x) = \top$

- *f(x)* is non-linear
- *f(x)* is non-smooth
- x is discrete

DNNs can deal effectively with 2 out of 3 issues

It All Started with a Question

Of course we could also ask:

Why would you do it in the first place?!?

- Is it going to generalize?!?
- How much initial data will we need?!?
- What about the overhead?!?

They are all good points!   …But we will (mostly) set them aside

- Still we have an interesting research question

Not a Brand-new Idea

There have been other attempts:

- Adorf, H.M., Johnston, M.D.: *A discrete stochastic neural network algorithm for constraint satisfaction problems* [1990]
- Lee, J.H.M., Leung, H.F., Won, H.W.: *Extending genet for non-binary csp's* [1995]
- Wang, C.J., Tsang, E.P.K.: *Solving constraint satisfaction problems using neural networks* [1991]
- Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, Samy Bengio: *Neural Combinatorial Optimization with Reinforcement Learning* [2016]
- …

What's different here?

- Existing approaches: problem-specific (better performance)
- We will be problem-agnostic (no human prior)

Getting to it...

How do we solve a CSP?

We iteratively:

- Evaluate the current partial solution
- Choose a new variable-value assignment

How do we solve a puzzle/solitaire?

We iteratively:

- Consider the current state of the board
- Choose a new move

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

Humans can learn the game by watching someone else play…

Can DNNs do the same?

The Learning Problem

The ML task:

- Input: a partial solution
- Output: a feasible assignment

Feasible?

- Local feasibility: GAC or similar level of consistency
- Global feasibility $\boxed{\text{guaranteed extension to full solution}}$

Representation:

Boolean vectors, one-hot encoding

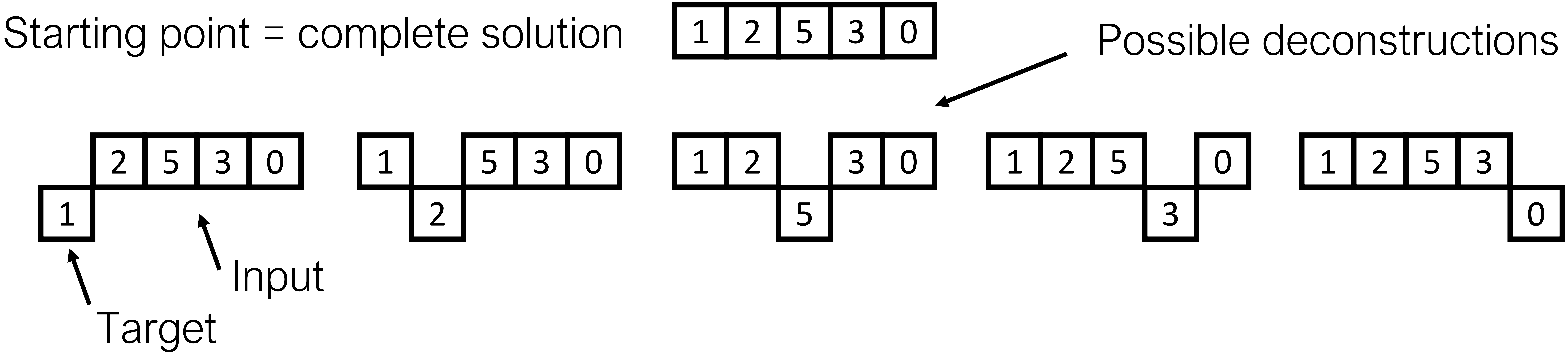
$$x_i \in \{1..n\}, x_i = j \longleftrightarrow$$

0	0	0	…	1	…	0
1	2	3		j		n

- Problem agnostic, but size-depedent

The Training Data

Example = partial solution + one (globally) feasible assignment

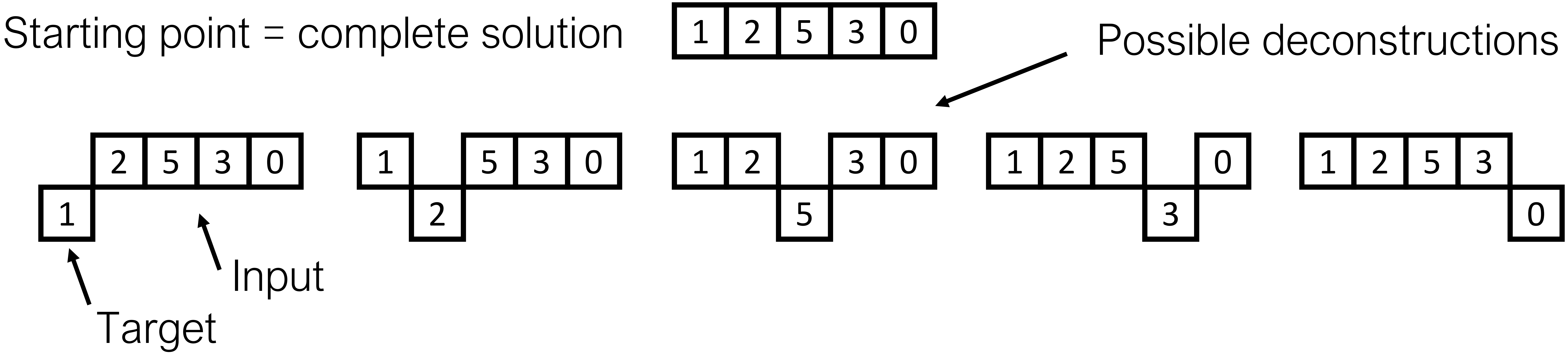


Two main approaches:

- Random: pick one deconstruction and repeat
- Systematic: consider all deconstructions and repeat

The Training Data

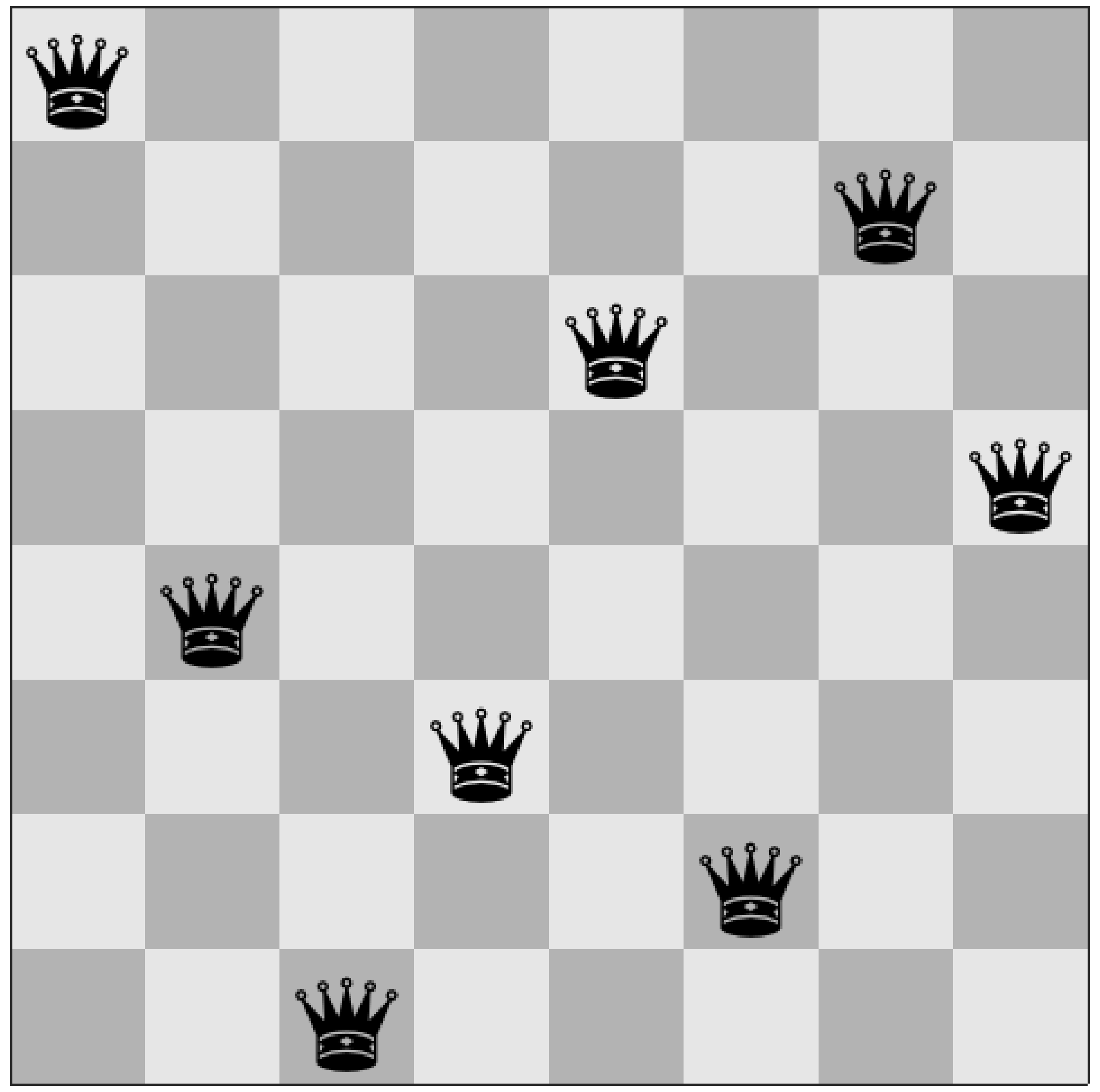
Example = partial solution + one (globally) feasible assignment



The target move is only one of the possible feasible choices!

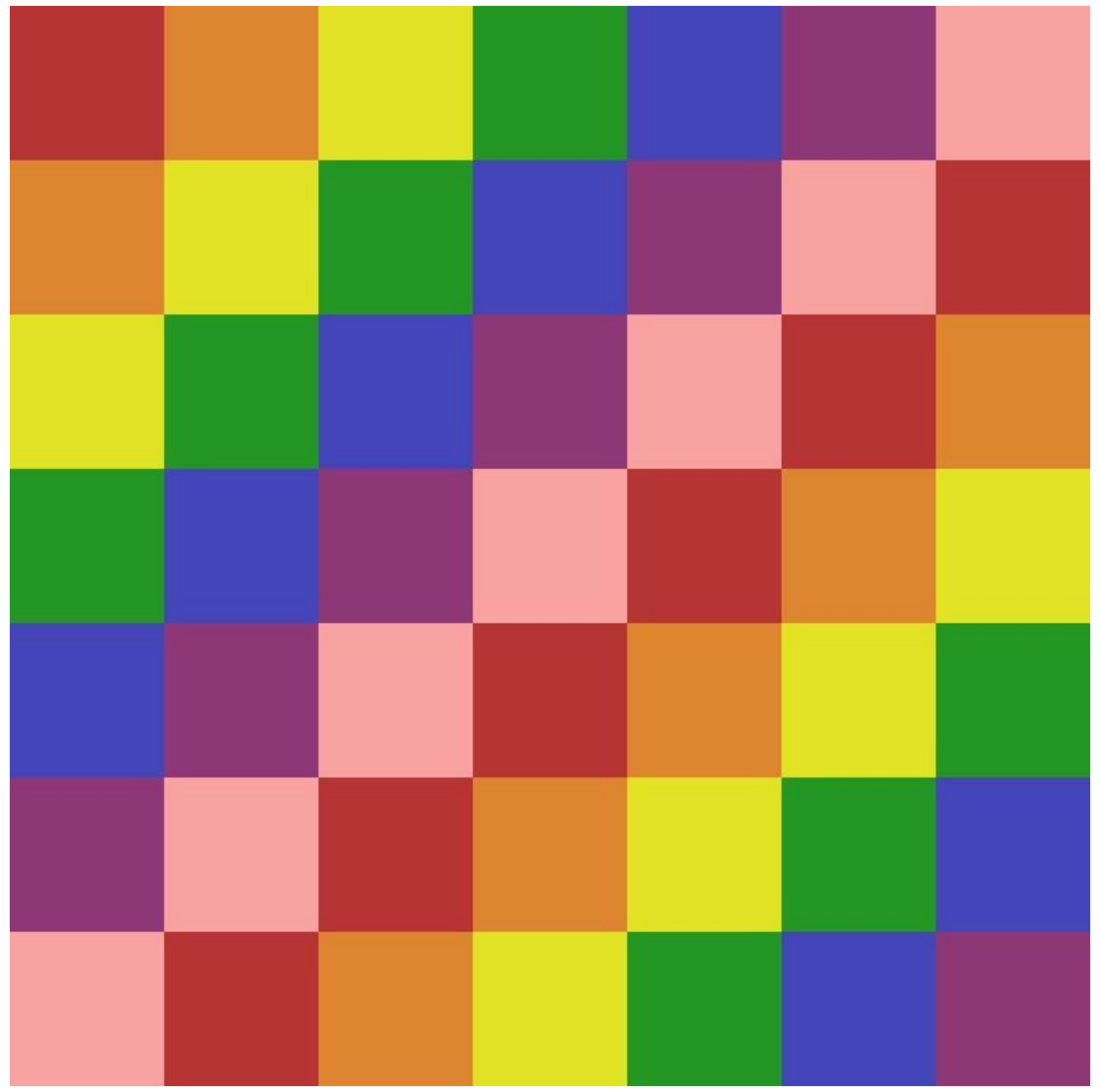
There may also be examples with conflicting output

Benchmarks



N-Queen Completion (8x8)

- Input: binary 64-vector
- Ouput: binary 64-vector
- Training (start): 8 solutions + all symmetries
- Test (start): 4 solutions + all symmetries
- Systematic deconstruction



Partial Latin Square (10x10)

- Input: binary 1000-vector
- Ouput: binary 1000-vector
- Training (start): 5k/10k solutions (over ~10^{31})
- Test (start): 5k/10k solutions (over ~10^{31})
- Random deconstruction

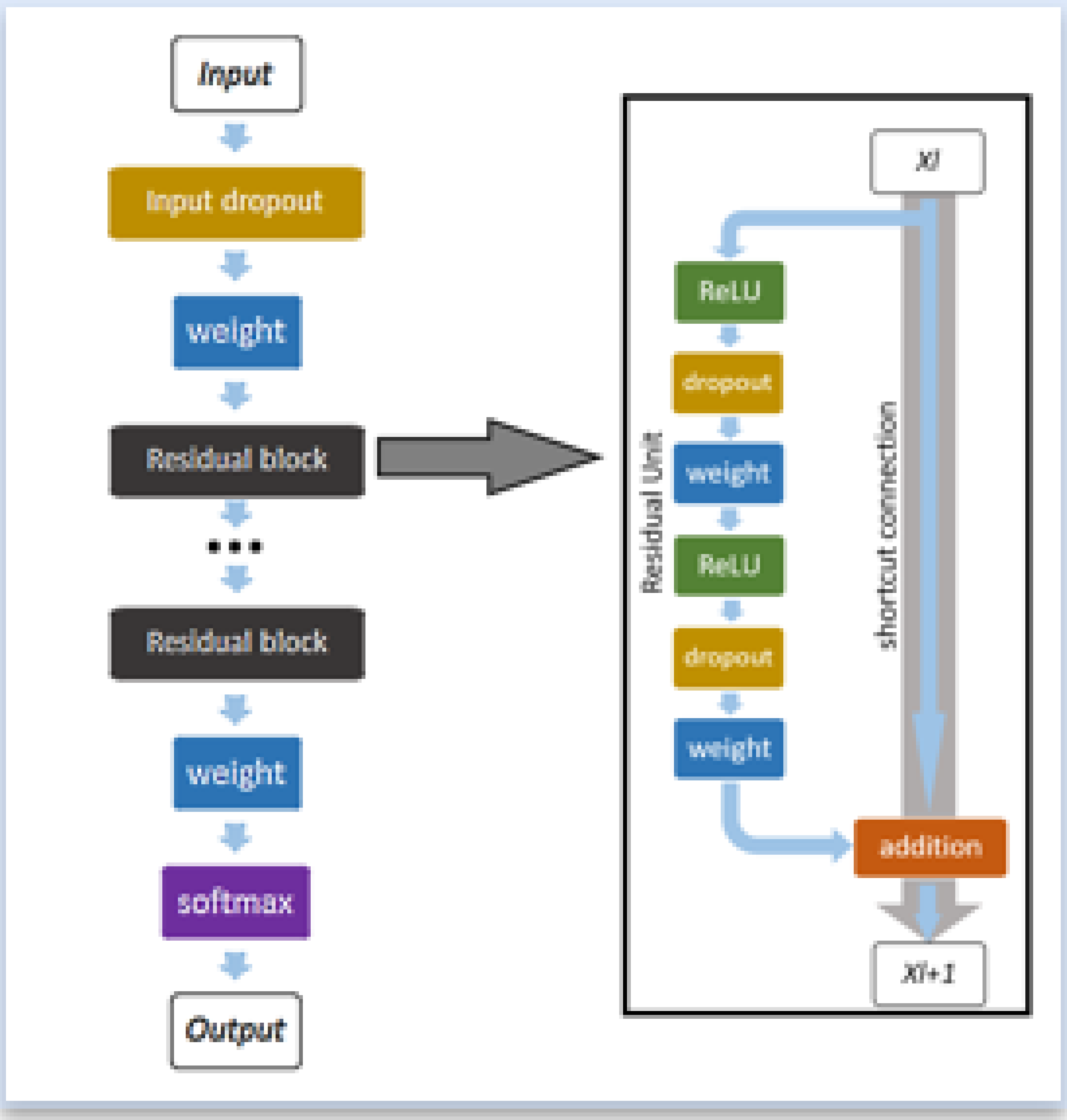
Deep Neural Network

Network architecture:

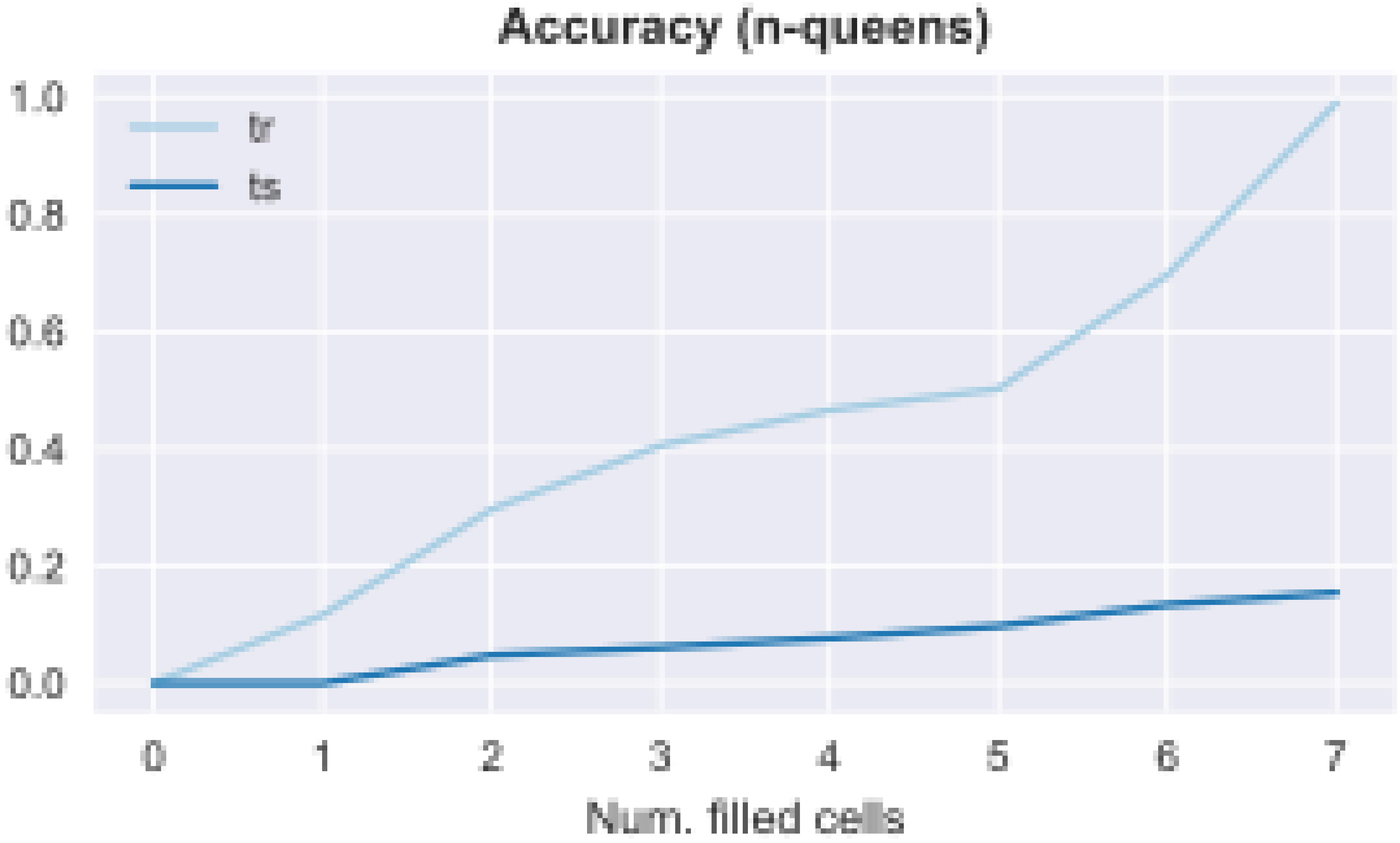
- Pre-activated residual networks
- > 22 layers (benchmark dependent)
- Feed-forward,
- Fully connected
- Width: 100-500 (benchmark dependent)

Training:

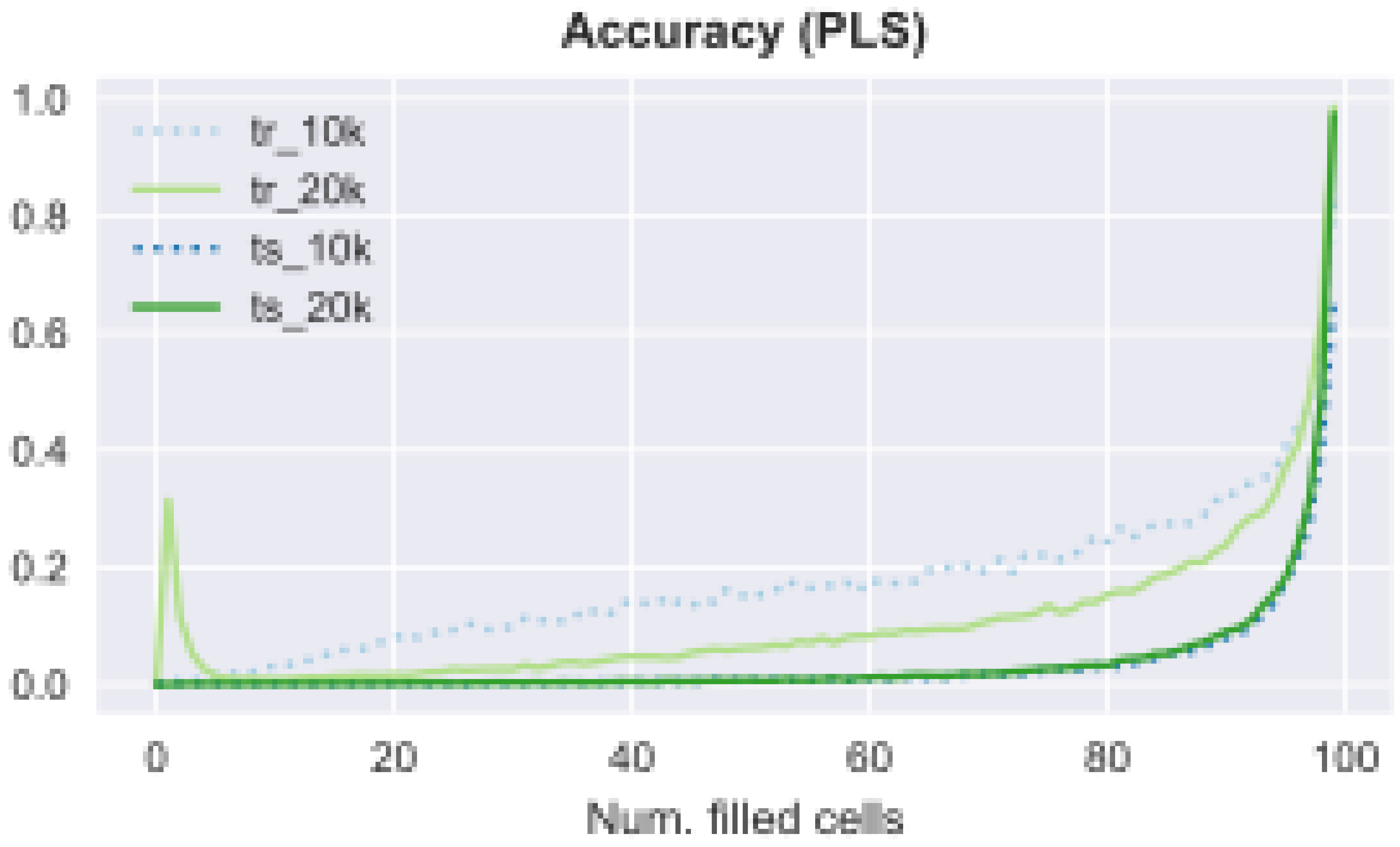
- Mini-batch optimization with shuffling and dropout
- Validation data: 10% of the training set
- Early-stop after 50 epochs without improvements



Can DNNs Imitate the Original Player?

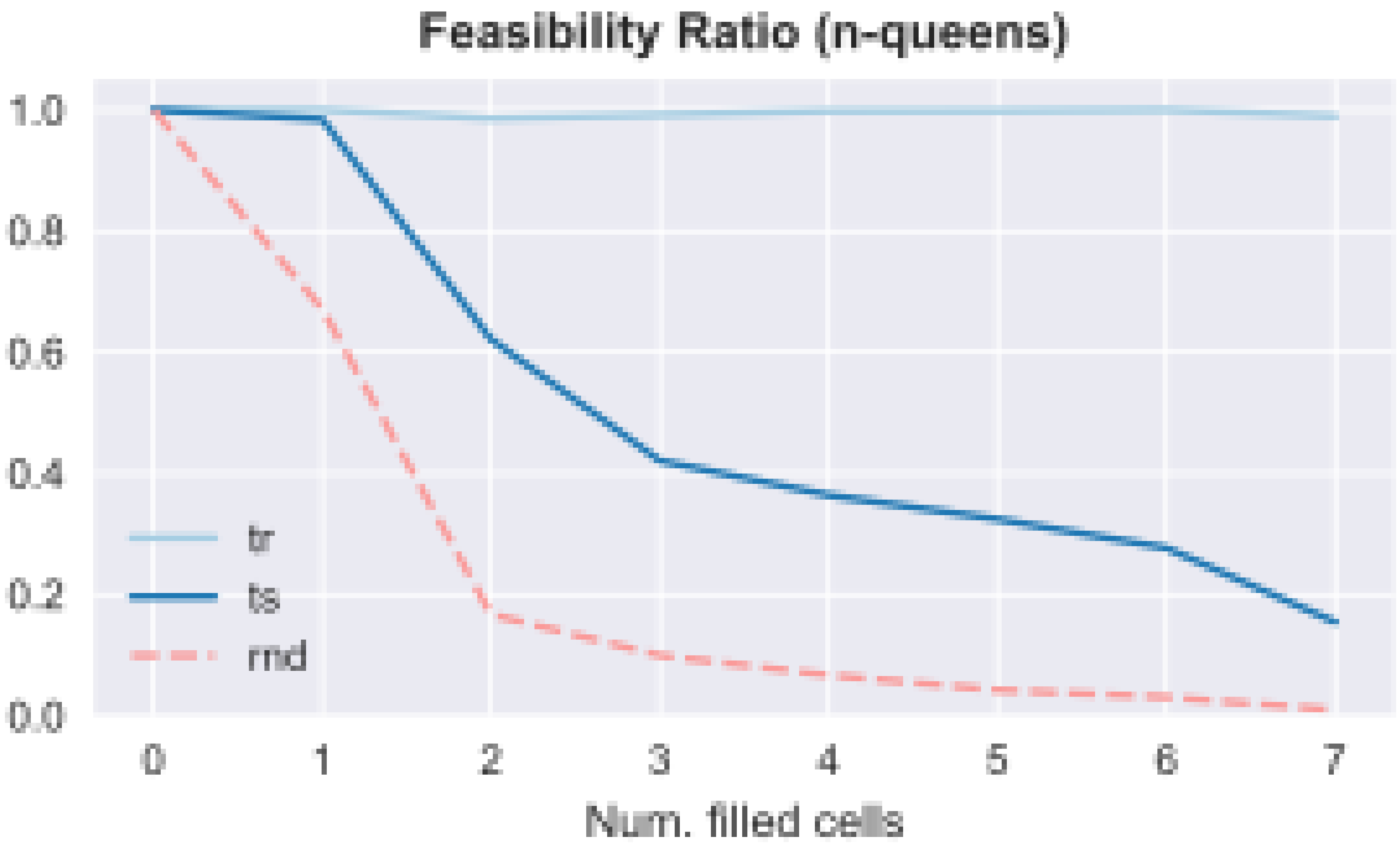


Can DNNs Imitate the Original Player?

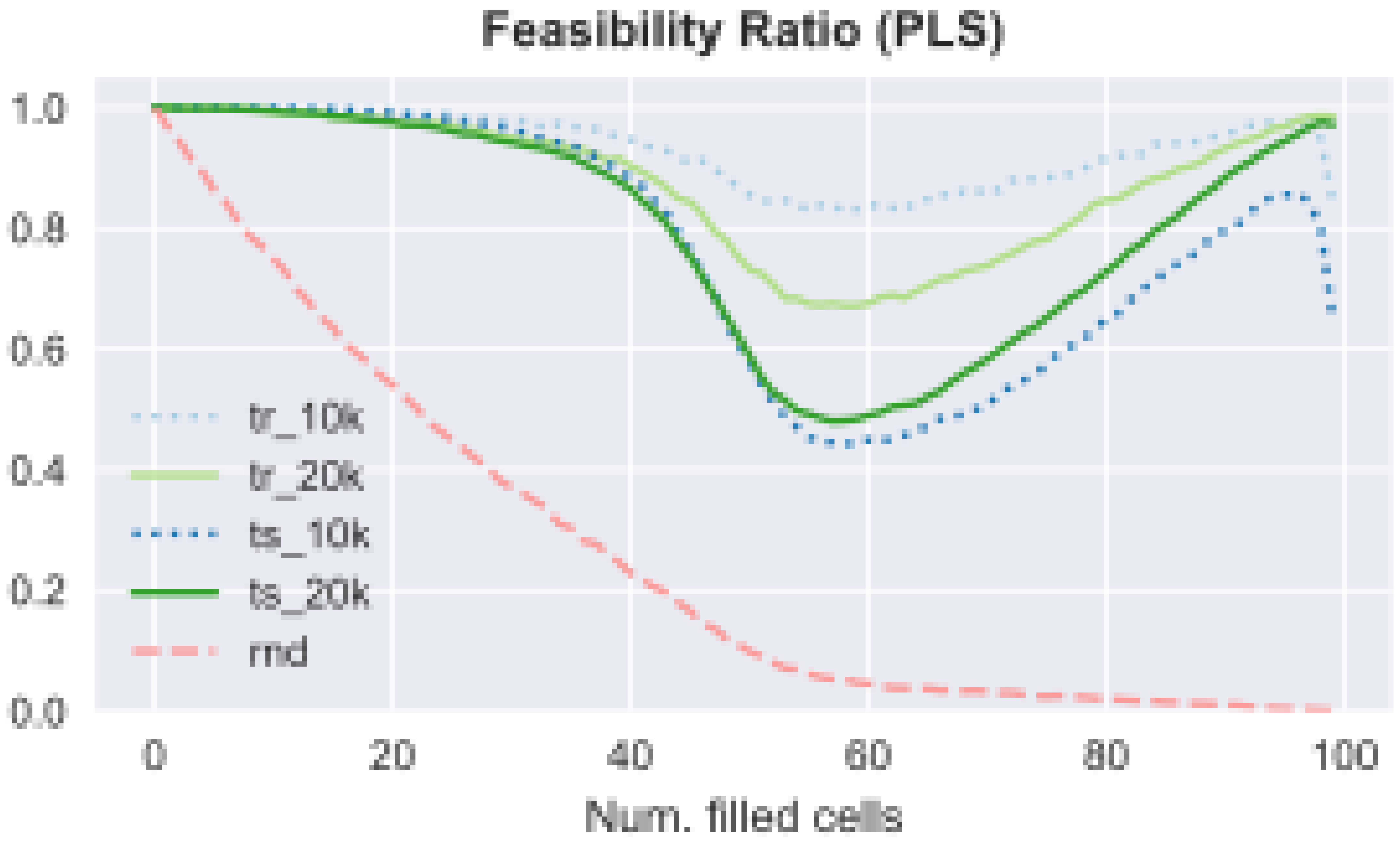


- For almost filled boards, only on the training set
- No generalization on the test set (as expected)

Can DNNs Choose Feasible Assignments?

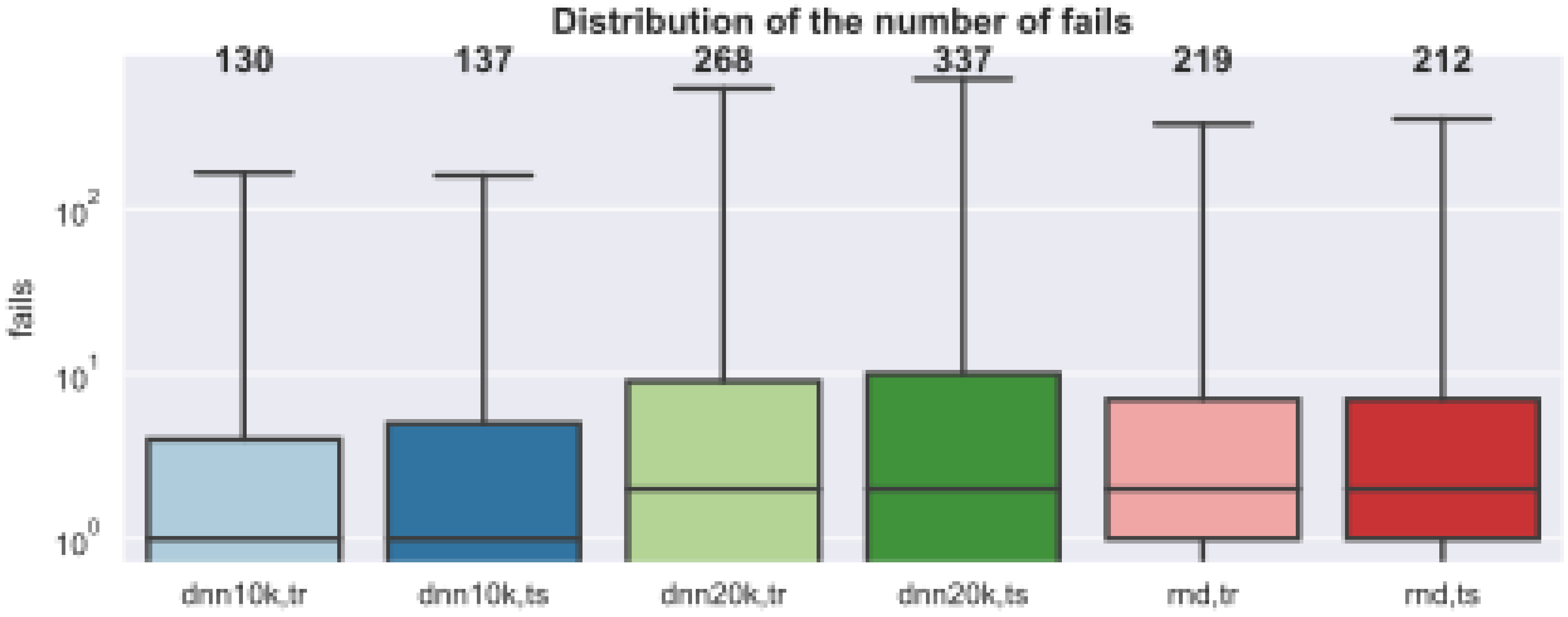


Can DNNs Choose Feasible Assignments?



- Yes! Quite surprisingly
- Random selection shown as a baseline

Can DNNs Improve Search?



- Modest improvement
- Feasibility does not necessarily translates to performance
- …But this is not the right setup

Final Remarks and Open Questions

Main facts

- Problem agnostic approach to solve a CSP via a DNN
- Focus on feasibility rather than optimality

On the practical side

- Generalize between different problem sizes (e.g. pointer networks)?
- Use fewer solutions?
- Add some human prior information?
- Make the DNN search aware?

On the scientific side

- Has the network learned some “abstract” rules?
- Is there a preference for certain constraints?
- Why the performance dip for average fill levels?

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

AI Group @DISI
http://ai.unibo.it

Michela Milano
<michela.milano@unibo.it>
Michele Lombardi
<michele.lombardi2@unibo.it>

Andrea Galassi
<a.galassi@unibo.it@unibo.it>
Paola Mello
<paola.mello@unibo.it@unibo.it>