A neuro-symbolic approach to learn
How to solve (serious) puzzles from solutions

Thomas Schiex
Joint work with S. Barbe, M. Defresne and R. Gambardella

Contextual optimisation

Imagine...

- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω

ω $\mathbf{x}$

Sudoku grid with solution

ω $\mathbf{x}$

Protein structure with its sequence

The model is written as a pairwise Graphical Model

Contextual optimisation

Imagine...

- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω

ω $\mathbf{x}$

Sudoku grid with solution

ω $\mathbf{x}$

Protein structure with its sequence

The model is written as a pairwise Graphical Model

Contextual optimisation

Imagine...

- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω

ω $\mathbf{x}$

Sudoku grid with solution

ω $\mathbf{x}$

Protein structure with its sequence

The model is written as a pairwise Graphical Model

Contextual optimisation

Imagine...

- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω

ω $\mathbf{x}$

Sudoku grid with solution

ω $\mathbf{x}$

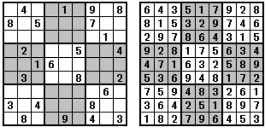
Protein structure with its sequence

The model is written as a pairwise Graphical Model

Contextual optimisation

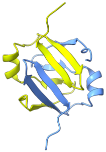
Imagine...

- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω



ω $\mathbf{x}$

Sudoku grid with solution



GPMANSSVELRVAE
AYPEDVGRGIVRMD
KQTRAKLGVSVGDY
VEVKKVD GPMANS
SVELRVAEAYPEDV
GRGIVRMDKQTRAK
LGVSVGDYVEVKKVD

ω $\mathbf{x}$

Protein structure with its sequence

The model is written as a **pairwise Graphical Model**

ANITI INRAE

Contextual optimisation

Imagine...

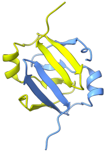
- You observe a set of designs $\{\mathbf{x}_i\}$
- Each design $\mathbf{x}_i$ is a solution of an unknown optimisation model
- This model depends on an observed context ω_i
- You'd like to learn how to generate new designs from new contexts ω

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

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

ω $\mathbf{x}$

Sudoku grid with solution



GPMANSSVELRVAE
AYPEDVGRGIVRMD
KQTRAKLGVSVGDY
VEVKKVD GPMANS
SVELRVAEAYPEDV
GRGIVRMDKQTRAK
LGVSVGDYVEVKKVD

ω $\mathbf{x}$

Protein structure with its sequence

The model is written as a **pairwise Graphical Model**

Pairwise graphical model (weighted CP)

Reminder

- A set $\mathbf{X}$ of variables — n variables
- Variable x_i has domain D_i — max. size d
- A set of cost functions — $c_{ij} : D_i \times D_j \to \mathbb{R}^+ \cup \{\infty\}$

Variables and parameters/costs

- The joint cost $C(\cdot)$ is the sum of all cost functions — Cost Function Network
- It defines a probability distribution: $P(\mathbf{x}) \propto \exp(-C(\mathbf{x}))$ — Markov Random Field
- Normalizing constant is #P-hard to compute

Pairwise graphical model (weighted CP)

Reminder

- A set $\mathbf{X}$ of variables — n variables
- Variable x_i has domain D_i — max. size d
- A set of cost functions — $c_{ij} : D_i \times D_j \to \mathbb{R}^+ \cup \{\infty\}$

Variables and parameters/costs

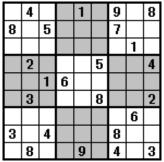
- The joint cost $C(\cdot)$ is the sum of all cost functions — Cost Function Network
- It defines a probability distribution: $P(\mathbf{x}) \propto \exp(-C(\mathbf{x}))$ — Markov Random Field
- Normalizing constant is #P-hard to compute

The neuro-symbolic architecture

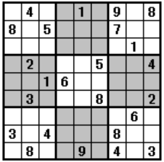
	4			1		9		8
8		5				7		
							1	
	2				5			4
		1	6					
	3				8			2
							6	
3		4				8		
	8			9		4		3

ω

The neuro-symbolic architecture

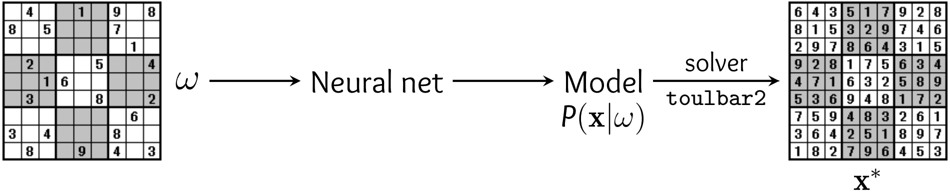


$\omega \longrightarrow$ Neural net

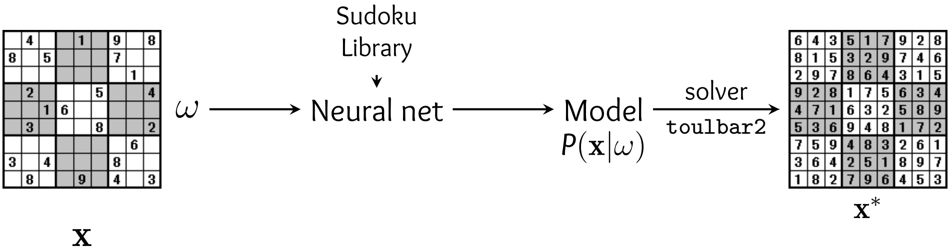


$\omega \longrightarrow$ Neural net $\longrightarrow$ Model $P(\mathbf{x}|\omega)$

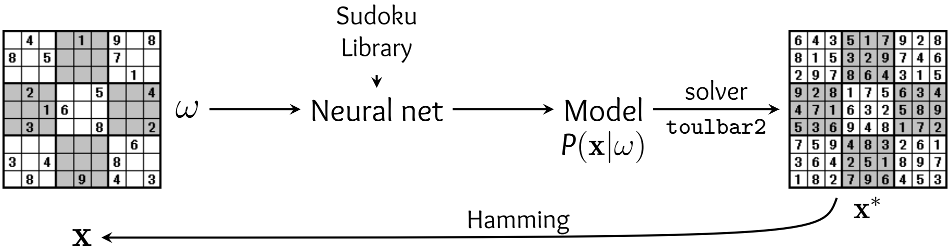
The neuro-symbolic architecture



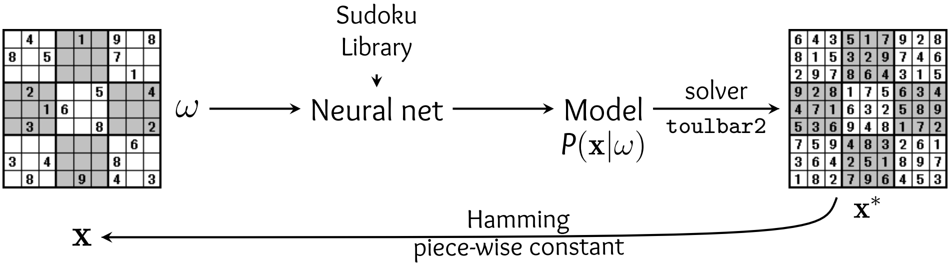
The neuro-symbolic architecture



The neuro-symbolic architecture



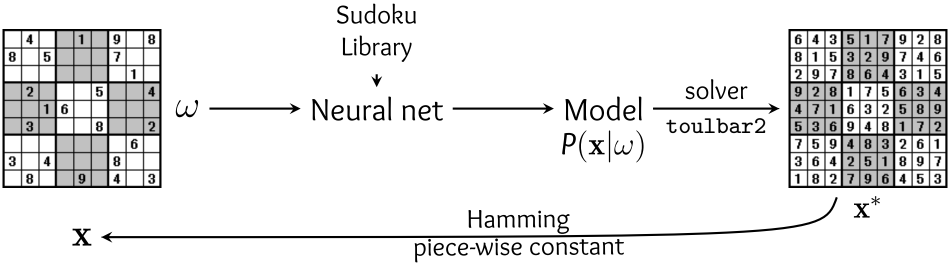
The neuro-symbolic architecture



Issues

- Gradients either zero or undefined
- Requires to repeatedly solve random NP-hard instances

The neuro-symbolic architecture



Natural choice: the negative loglikelihood #P-hard

- **E-PLL** loss: improves Besag's pseudo-loglikelihood [1] (sublinear) IJCAI23 [6]
- Kicks the solver out of the training loop (scalable training)

Scalable Coupling of Deep Learning with Logical Reasoning*

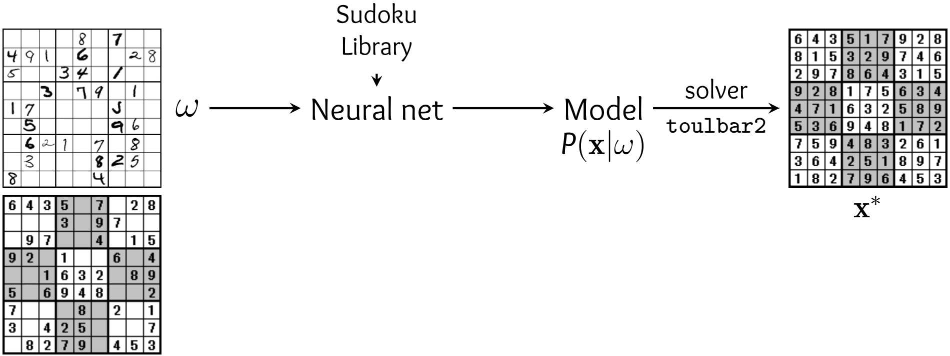
Marianne Defresne[1,2], **Sophie Barbe**[2] and **Thomas Schiex**[1]
[1]Université Fédérale de Toulouse, ANITI, INRAE, UR 875, 31326 Toulouse, France
[2]TBI, Université de Toulouse, CNRS, INRAE, INSA, ANITI, 31077 Toulouse, France
{marianne.defresne, sophie.barbe}@insa-toulouse.fr, thomas.schiex@inrae.fr

Many approaches

The **hardness** of Sudoku grids depends on the number of given (17 is minimal).

Type	Approach	Acc.	#given	Train set	Train time (h)
DL	RRN NeurIPS18 [11]	96.6%	17	180,000	>50
	Rec. Trans. ICLR23 [13]	96.7%	17	180,000	>50
	DDPM ICLR25 [14]	99.2–100%	33.8	100,000	13.6
	DDPM	0.2%	17	-	-
Relax+DL	SATNet ICML19 [12]	95.1–99.8%	36.2	9,000	2.9
	SATNet	86.1–86.2%	17	-	-
CO	MaxSAT IJCAI23 [2]	**100%**	-	200	0.01
CO + ML	GM/APLL CP2020 [3]	**100%**	17	9,000	1.5
CO+DL	Struc. Perceptron IJCAI23 [6]	**100%**	17	1,000	>50
	E-NPLL IJCAI23 [6]	**100%**	17	**100**	0.05

Learning to play Visual Sudoku



Learns symbols and rules simultaneously

- The meaning of the given is removed from the training set label
- For the E-PLL, we use their optimal values given the currently learned model

Comparison with state-of-the-art (same dataset)

Approach	Solved	Training (h)
Rec. Trans. ICLR23 [13]	75.6%	5.1
NeSy. Prog. NeurIPS23 [8]	92.2–94.4%	4.7
E-PLL IJCAI23 [6]	93.4%	3.2

ANITI INRAE

Decision Focused Learning

- Assumes constraints are known
- Data: pairs $\langle \omega, \mathbf{c} \rangle$ where $\mathbf{c}$ define the criterion parameters
- Aim: minimize regret (difference in real cost of the predicted and optimal solution)

We can compute the optimal solution $\mathbf{x}^*$ and compare to a DFL approach

Decision Focused Learning

- Assumes constraints are known
- Data: pairs $\langle \omega, \mathbf{c} \rangle$ where $\mathbf{c}$ define the criterion parameters
- Aim: minimize regret (difference in real cost of the predicted and optimal solution)

We can compute the optimal solution $\mathbf{x}^*$ and compare to a DFL approach

Application to Decision-focused learning

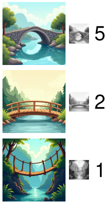
Decision Focused Learning

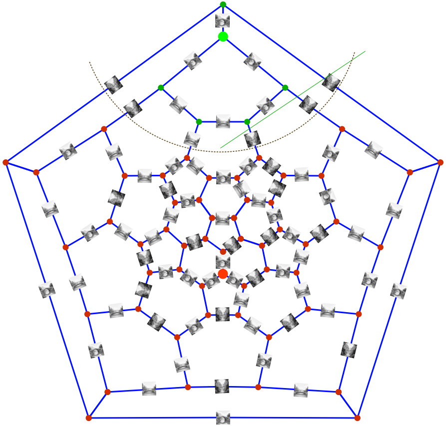
- Assumes constraints are known
- Data: pairs $\langle \omega, \mathbf{c} \rangle$ where $\mathbf{c}$ define the criterion parameters
- Aim: minimize regret (difference in real cost of the predicted and optimal solution)

We can compute the optimal solution $\mathbf{x}^*$ and compare to a DFL approach

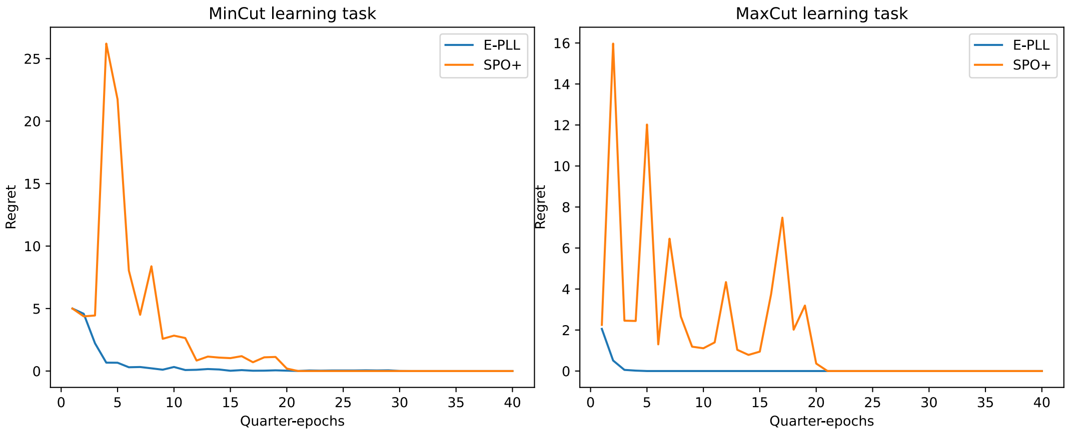
Simple model

- Node variables: side of the cut
- MaxCut cost matrix $\propto$ identity (I)
- MinCut cost matrix $\propto (1 - I)$
- Predicted multipliers, E-PLL

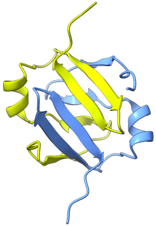




MinCut, MaxCut, Regret and SPO+ [7]



Learning to design proteins: Effie



$\mathbf{x}$

PDB
200,000 $(\omega, \mathbf{x})$

Design constraints

ω → Neural net → Model $P(\mathbf{x}|\omega)$ → (solver, `toulbar2`) → $\mathbf{x}^*$

Comparison with recent models

Outperforms all-atoms Rosetta scoring functions

- Metric: **Native Sequence Recovery** rate (NSR)

Approach	Rosetta	Effie
NSR	17.9%	32.8%

Outperforms recent pure Deep Learning scoring functions

	ProteinMPNN	Effie
NSR	45.9%	48.4%

Comparison with recent models

Outperforms all-atoms Rosetta scoring functions

- Metric: **Native Sequence Recovery** rate (NSR)

Approach	Rosetta	Effie
NSR	17.9%	32.8%

Outperforms recent pure Deep Learning scoring functions

	ProteinMPNN	Effie
NSR	45.9%	**48.4%**

Predicting SARS-CoV2 variants

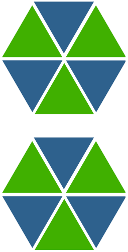
Enumerate CoViD variants with a bounded number of mutations

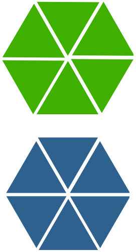
- Uses only the initial March 2020 RBD-ACE2 structure + Effie/toulbar2
- Relies on a global constraint to bound mutations [9]
- Predicts all the first SARS-CoV2 VoCs ($\alpha, \beta, \gamma, \delta, \kappa, \iota, \lambda$ and μ)
- In a few seconds, on one CPU-thread.

Approach experimentally shown to predict contagious antibody-resistant variants [5]
Also designed experimentally tested anti-SARS-CoV-2 nanobody binders

Design of an heteromeric hexamer

- Design  and  that self-assemble as  but not as  or 
- Compare Effie+tb2 (bi-criteria [4]) with ProteinMPNN (multi-state)





Design of an heteromeric hexamer

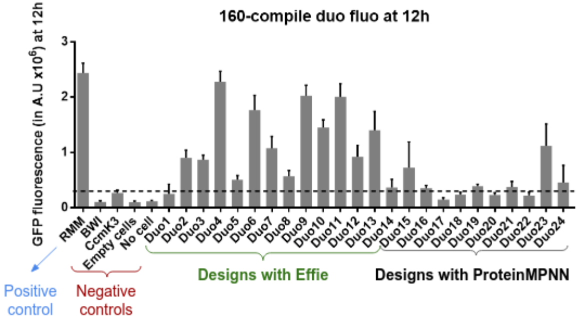
- Design and that self-assemble as but not as or
- Compare Effie+tb2 (bi-criteria [4]) with ProteinMPNN (multi-state)

How often is [icon] better than [icon]?

Scoring →	Effie	PMPNN
Effie	100 %	99.5 %
PMPNN	3.0 %	82.6 %

How often is better than ?

Scoring →	Effie	PMPNN
Effie	100 %	99.5 %
PMPNN	3.0 %	82.6 %



Conclusions

A Neural Net, a GM, and a discrete optimizer in a NeSy autoencoder

- A NeSy Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(\mathbf{x}|\omega)$ from raw inputs
- Represented as a GM in a fully explorable and controllable latent layer
- Using discrete optimisation (toulbar2) that accepts side constraints
- All this with scalable training

ANITI INRAE

Conclusions

A Neural Net, a GM, and a discrete optimizer in a NeSy autoencoder

- A NeSy Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(\mathbf{x}|\omega)$ from raw inputs
- Represented as a GM in a fully explorable and controllable latent layer
- Using discrete optimisation (toulbar2) that accepts side constraints
- All this with scalable training

Conclusions

A Neural Net, a GM, and a discrete optimizer in a NeSy autoencoder

- A NeSy Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(\mathbf{x}|\omega)$ from raw inputs
- Represented as a GM in a fully explorable and controllable latent layer
- Using discrete optimisation (toulbar2) that accepts side constraints
- All this with scalable training

ANITI INRAE

Conclusions

A Neural Net, a GM, and a discrete optimizer in a NeSy autoencoder

- A NeSy Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(\mathbf{x}|\omega)$ from raw inputs
- Represented as a GM in a fully explorable and controllable latent layer
- Using discrete optimisation (toulbar2) that accepts side constraints
- All this with scalable training

Conclusions

A Neural Net, a GM, and a discrete optimizer in a NeSy autoencoder

- A NeSy Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(\mathbf{x}|\omega)$ from raw inputs
- Represented as a GM in a fully explorable and controllable latent layer
- Using discrete optimisation (toulbar2) that accepts side constraints
- All this with scalable training

ANITI INRAE

Acknowledgments

AI/toulbar2

S. de Givry (INRA)
G. Katsirelos (INRA)
M. Zytnicki (PhD, INRA)
D. Allouche (INRA)
M. Ruffini (INRA)
V. Durante (ANITI, PhD)
H. Nguyen (PhD, INRA)
C. Brouard (ML, INRA)
S. Buchet (INRAE/ANITI)
P. Montalbano (ANITI, PhD)
M. Cooper (IRIT, Toulouse)
J. Larrosa (UPC, Spain)
F. Heras (UPC, Spain)
M. Sanchez (Spain)
E. Rollon (UPC, Spain)
P. Meseguer (CSIC, Spain)
G. Verfaillie (ONERA, ret.)
JH. Lee (CU. Hong Kong)
C. Bessiere (LIMM, Montpellier)
JP. Métivier (GREYC, Caen)
S. Loudni (GREYC, Caen)
M. Fontaine (GREYC, Caen),...

DL/Protein Design

A. Voet (KU Leuven)
A. Olichon (INSERM)
D. Simoncini (UFT, Toulouse)
S. Barbe (INSA, Toulouse)
M. Defresne (INRAE, PhD)
Y. Bouchiba (INSA, PhD)
C. Dumont (INSA, Toulouse)
J. Vucinic (INRA/INSA)
S. Traoré (PhD, CEA)
C. Viricel (PhD)
K. Zhang (Riken, CBDR)
S. Yagi (Riken, CBDR)
S. Tagami (Riken, CBDR)
RosettaCommons (U. Washington)
W. Sheffler (U. Washington)
V. Mulligan (Flatiron Institute, NY)
C. Bahl (IPI, Boston)
PyRosetta (U. John Hopkins)
B. Donald (U. North Carolina)
K. Roberts (U. North Carolina)
T. Simonson (Polytechnique)
J. Cortes (LAAS/CNRS),...

My apologies to those missing in these lists. Even imperfect lists seem better than no list

References

[1] Julian Besag. "Statistical analysis of non-lattice data". In: Journal of the Royal Statistical Society: Series D (The Statistician) 24.3 (1975), pp. 179–195.

[2] Christian Bessiere, Clément Carbonnel, and Areski Himeur. "Learning Constraint Networks over Unknown Constraint Languages". In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. Ed. by Edith Elkind. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 1876–1883. DOI: 10.24963/ijcai.2023/208. URL: https://doi.org/10.24963/ijcai.2023/208.

[3] Céline Brouard, Simon de Givry, and Thomas Schiex. "Pushing Data into CP Models Using Graphical Model Learning and Solving". In: International Conference on Principles and Practice of Constraint Programming. Springer. 2020, pp. 811–827.

[4] Samuel Buchet et al. "Bi-objective Discrete Graphical Model Optimization". In: International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer. 2024, pp. 136–152.

[5] Mireia Solà Colom et al. "Complete combinatorial mutational enumeration of a protein functional site enables sequence-landscape mapping and identifies highly-mutated variants that retain activity". In: Protein Science 33.8 (Aug. 2024), e5109. DOI: 10.1002/pro.5109. URL: https://hal.science/hal-04646616.

[6] Marianne Defresne, Sophie Barbe, and Thomas Schiex. "Scalable Coupling of Deep Learning with Logical Reasoning". In: Thirty-second International Joint Conference on Artificial Intelligence, IJCAI'2023. 2023.

[7] Adam N. Elmachtoub and Paul Grigas. "Smart "Predict, then Optimize"". In: Management Science 68.1 (2022), pp. 9–26. DOI: 10.1287/mnsc.2020.3922.

[8] Zenan Li et al. "Neuro-symbolic learning yielding logical constraints". In: Advances in Neural Information Processing Systems 36 (2023), pp. 21635–21657.

[9] Pierre Montalbano, Simon de Givry, and George Katsirelos. "Multiple-Choice Knapsack Constraint in Graphical Models". In: Proc. of CPAIOR'22. 2022.

[10] Yatin Nandwani et al. "Neural Learning of One-of-Many Solutions for Combinatorial Problems in Structured Output Spaces". In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL: https://openreview.net/forum?id=ATp1nW2FuZL.

[11] Rasmus Palm, Ulrich Paquet, and Ole Winther. "Recurrent Relational Networks". In: Advances in Neural Information Processing Systems. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[12] Po-Wei Wang et al. "SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver". In: Proceedings of the 36th International Conference on Machine Learning. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6545–6554.

[13] Zhun Yang, Adam Ishay, and Joohyung Lee. "Learning to Solve Constraint Satisfaction Problems with Recurrent Transformer". In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL: https://openreview.net/forum?id=udNhDCr2KQe.

[14] Jiacheng Ye et al. "Beyond Autoregression: Discrete Diffusion for Complex Reasoning and Planning". In: The Thirteenth International Conference on Learning Representations. 2025. URL: https://openreview.net/forum?id=NRYgUzSPZz.

Learning to play Many-Solutions Sudokus

Sudokus have only one solution (single target for DL)

- Existing DL architectures fail on many-solutions Sudokus [10]
- Corrected using a Reinforcement learning approach
- Training set with 5 solutions per instance
- Ability to generate additional solutions

Our architecture directly learns how to solve many-solutions Sudokus

Learning to play Many-Solutions Sudokus

Sudokus have only one solution (single target for DL)

- Existing DL architectures fail on many-solutions Sudokus [10]
- Corrected using a Reinforcement learning approach
- Training set with 5 solutions per instance
- Ability to generate additional solutions

Our architecture directly learns how to solve many-solutions Sudokus

Learning to play Futoshiki

Sudoku is easy, only one type of constraint

- Our architecture directly learns how to play Futoshiki
- Includes both difference and inequality constraints
- Perfect solving, expected constraints learned