Neuro-Symbolic Learning of Graphical Models

Symbol Grounding through Data Imputation

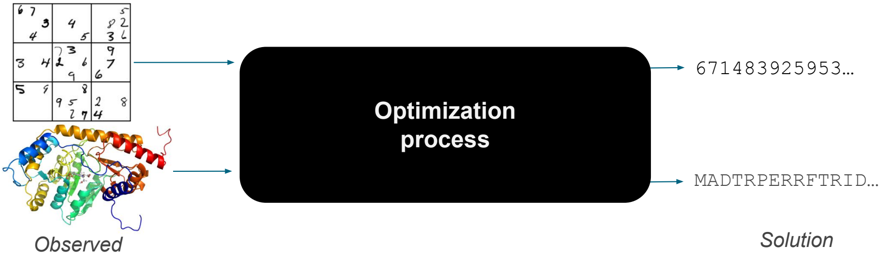
CompAI 2025 workshop

Marianne Defresne (KU Leuven), Romain Gambardella (Telecom Paris), Sophie Barbe (TBI, INRAE) and Thomas Schiex (MIAT, INRAE)
marianne.defresne@kuleuven.be

18 August 2025

INRAE
ANITI ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE
INSA TOULOUSE
tbi Toulouse Biotechnology Institute Bio & Chemical Engineering
MIA TOULOUSE
BioEco Graduate School Biotechnology Building Bio-Based Economy

Learning how to reason



Observed

Optimization process

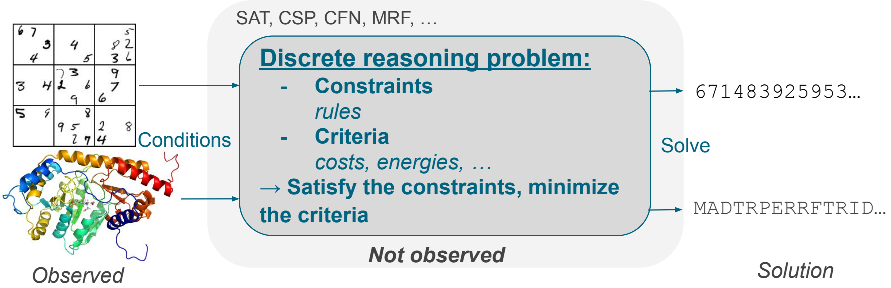
671483925953…

MADTRPERRFTRID…

Solution

- **Observation:** natural & structured objects
- Result from an **optimization** process → aim to re-purpose it

Learning how to reason



SAT, CSP, CFN, MRF, …

Discrete reasoning problem:

- **Constraints**
  rules
- **Criteria**
  costs, energies, …

→ **Satisfy the constraints, minimize the criteria**

Not observed

Conditions

Observed

Solve

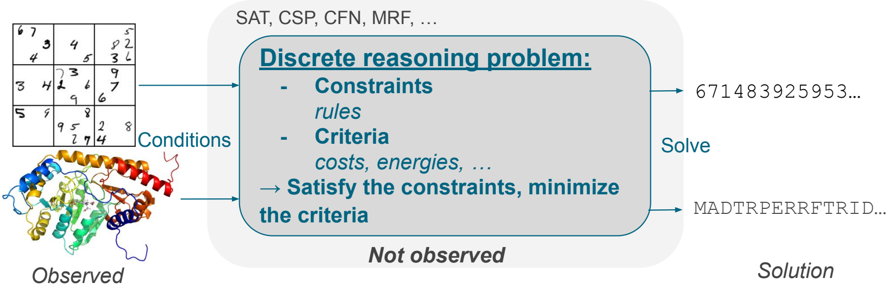
671483925953…

MADTRPERRFTRID…

Solution

- **Goal:** solve new instances with no access to the discrete model parameters
  - Learn to predict the underlying constraints & criteria
  - Decision-focused learning

Learning how to reason



SAT, CSP, CFN, MRF, …

Discrete reasoning problem:

- **Constraints**
  rules
- **Criteria**
  costs, energies, …

→ **Satisfy the constraints, minimize the criteria**

Not observed

- **Goal:** solve new instances with no access to the discrete model parameters
  - Learn to predict the underlying constraints & criteria
  - Decision-focused learning
- **How?** By interfacing two branches of AI:
  - Deep Learning (DL)
  - Discrete reasoning (CFN)

Zoom on the Sudoku toy problem

- Sudoku as a pairwise **Graphical Model**
  - One variable X_i per cell
  - Domain $D_i = 9$ digits
  - Cost functions = rules $\qquad c_{ij} : D_i \times D_j \to \mathbb{R}^+ \cup \{\infty\}$
- Joint cost $C(\cdot)$: sum of all cost functions
  - Cost Function Network (CFN)
  - Probability distribution: $P(\mathbf{s}) \propto \exp(-C(\mathbf{s}))$
- **Solving** a grid: $\mathbf{s}^* = \arg\min_s C(\mathbf{s}) = \operatorname{argmax}_s P(\mathbf{s})$
  - NP-hard, solved by toulbar2 [6]

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

$$\begin{array}{rc} & \overbrace{\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}}^{\text{values of variable } j} \\ \text{values of variable } i \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right. & \begin{pmatrix} \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \infty & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \infty & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \infty & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \infty & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \infty \end{pmatrix} \end{array}$$

Zoom on the Sudoku toy problem

- Sudoku as a pairwise **Graphical Model**
  - > One variable X_i per cell
  - > Domain D_i = 9 digits
  - > Cost functions = rules $\quad c_{ij} : D_i \times D_j \to \mathbb{R}^+ \cup \{\infty\}$
- Joint cost $C(\cdot)$: sum of all cost functions
  - > Cost Function Network (CFN)
  - > Probability distribution: $P(\mathbf{s}) \propto \exp(-C(\mathbf{s}))$
- **Solving** a grid: $\mathbf{s}^* = \arg\min_s C(\mathbf{s}) = \text{argmax}_s P(\mathbf{s})$
  - > NP-hard, solved by toulbar2 [6]

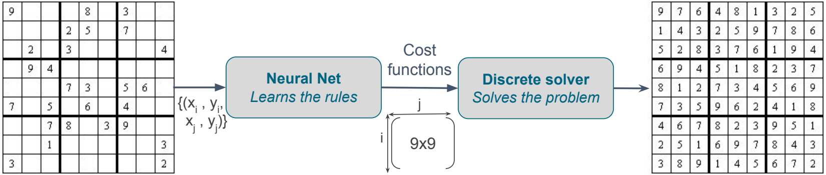
9	7	6	4	8	1	3	2	5
1	4	3	2	5	9	7	8	6
5	2	8	3	7	6	1	9	4
6	9	4	5	1	8	2	3	7
8	1	2	7	3	4	5	6	9
7	3	5	9	6	2	4	1	8
4	6	7	8	2	3	9	5	1
2	5	1	6	9	7	8	4	3
3	8	9	1	4	5	6	7	2

$$\text{values of variable i}\left\{\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\overbrace{\hphantom{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}}^{\text{values of variable j}}} \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & \infty & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & \infty & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & \infty & 0 & 0 \\
8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \infty & 0 \\
9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \infty
\end{array}\right.$$

What if the rules are unknown? → Learn from examples [2]

Learning how to play Sudoku

- Aim: learning a representation of the Sudoku rules
  - Data: (initial grid, solved grid)
  - Rules (cost functions) are **unknown**



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

$\{(x_i , y_i, x_j , y_j)\}$

Neural Net
Learns the rules

Cost functions

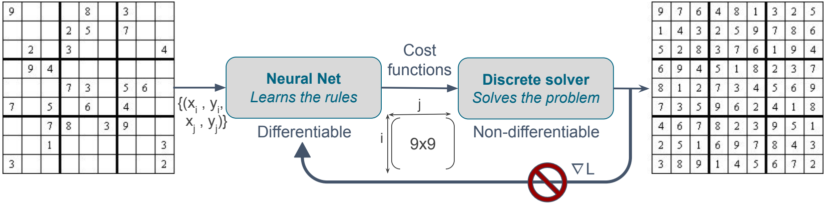
Discrete solver
Solves the problem

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

Hybrid encoder-decoder architecture

Learning how to play Sudoku

- Aim: learning a representation of the Sudoku rules
  - Data: (initial grid, solved grid)
  - Rules (cost functions) are **unknown**



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

$\{(x_i, y_i, x_j, y_j)\}$

Neural Net
Learns the rules

Differentiable

Cost functions

j

i

9x9

Discrete solver
Solves the problem

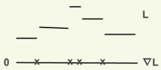
Non-differentiable

∇L

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

Aiming to minimize the decision error

$$L = Hamming(y, \hat{y}) = \frac{1}{81} \sum_{i=1}^{81} \mathbb{1}[y_i \neq \hat{y}_i]$$

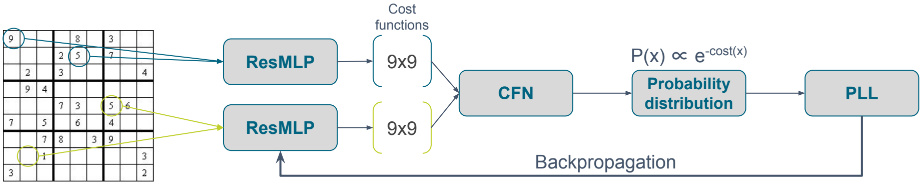


- Issue: **discrete objective** vs **gradient descent** → ∇L is 0 or non-existant
- Repeated NP-hard solve calls → no scalability

2-stage approach with the Emmental-PLL (E-PLL) [4]

Scaling: no solver during training

- How to assess the learned discrete problem without solving it?
  - **Log likelihood** (log-probability of the training set) Intractable (#P)
  - **Pseudo-log likelihood** [1]: $-\sum_i log\ P(y_i|y_{-i})$



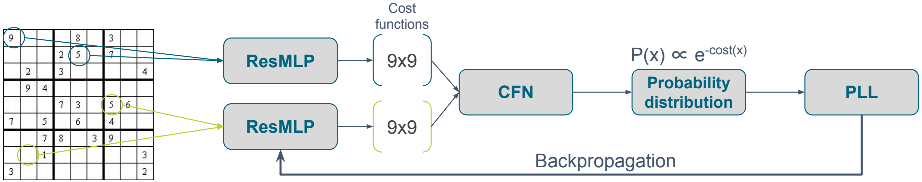
9				8		3		
			2	5		7		
	2		3					4
	9	4						
			7	3		5	6	
7		5		6		4		
		7	8		3	9		
		1						3
3								2

[3] 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

2-stage approach with the Emmental-PLL (E-PLL) [4]

Scaling: no solver during training

- How to assess the learned discrete problem without solving it?
  - **Log likelihood** (log-probability of the training set) — Intractable (#P)
  - **Pseudo-log likelihood** [1]: $-\sum_i log\ P(y_i|y_{-i})$ — 0% grid solved



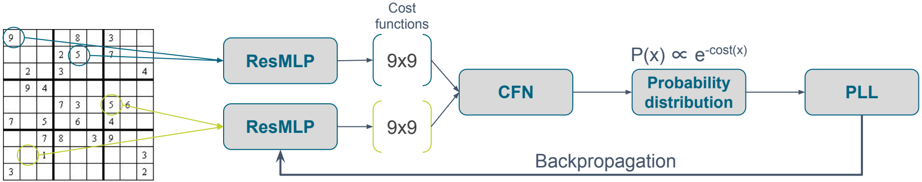
9				8		3		
			2	5		7		
	2		3					4
	9	4						
			7	3		5	6	
7		5		6		4		
		7	8		3	9		
		1						3
3								2

[3] 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

2-stage approach with the Emmental-PLL (E-PLL) [4]

Scaling: no solver during training

- How to assess the learned discrete problem without solving it?
  - > **Log likelihood** (log-probability of the training set) Intractable (#P)
  - > **Pseudo-log likelihood** [1]: $-\sum_i log\ P(y_i|y_{-i})$ 0% grid solved



- PLL enhanced to learn constraints [4], **E-PLL**: $-\sum_i log\ P(y_i|y_{-(i \cup M(i))})$

[3] 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

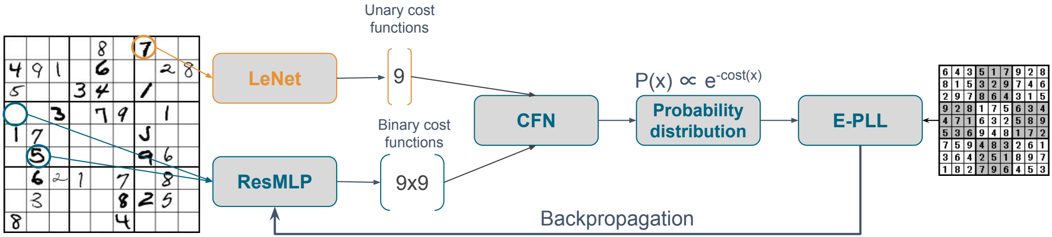
Many approaches for learning how to play Sudoku

Type	Approach	Acc.	#given	Train set	Train time (h)
DL	RRN NeurIPS18[8]	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+ML	GM/APLL CP2020[2]	**100%**	17	9,000	1.5
CO+DL	Hinge IJCAI23[4]	**100%**	17	1,000	>50
	E-NPLL IJCAI23[4]	**100%**	17	**100**	0.05

- E-PLL is **exact**, **data-efficient** and **scales** to large instances

Learning to play Visual Sudoku

Visual sudoku: Learn symbols and rules simultaneously

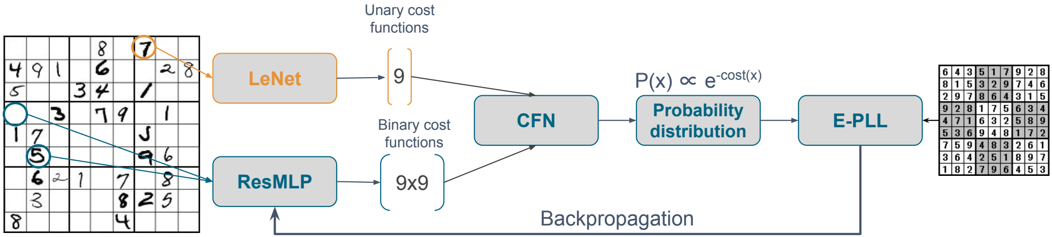


				8		7		
4	9	1		6			2	8
5			3	4		1		
		3		7	9		1	
1	7					5		
	5					9	6	
	6	2	1		7		8	
	3				8	2	5	
8					4			

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

Learning to play Visual Sudoku

Visual sudoku: Learn symbols and rules simultaneously

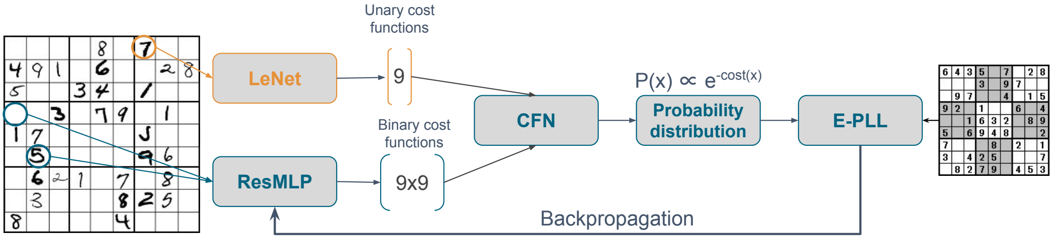


6	4	3	5	1	7	9	2	8
8	1	5	3	2	9	7	4	6
2	9	7	8	6	4	3	1	5
9	2	8	1	7	5	6	3	4
4	7	1	6	3	2	5	8	9
5	3	6	9	4	8	1	7	2
7	5	9	4	8	3	2	6	1
3	6	4	2	5	1	8	9	7
1	8	2	7	9	6	4	5	3

- Cheating! Direct supervision of the digit
  - > **Symbol grounding problem**: learning the mapping image ↦ symbols

Symbol grounding problem

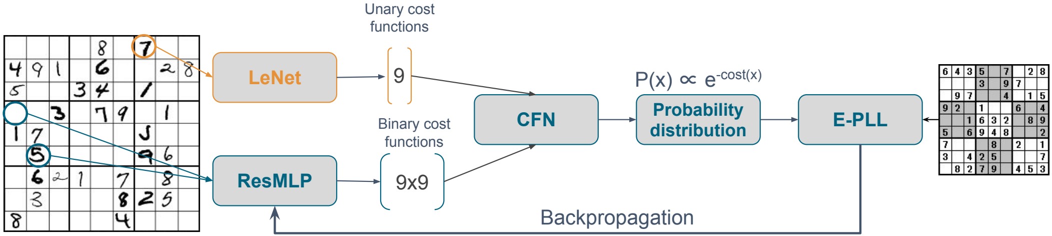
Visual sudoku: Learn symbols and rules simultaneously



- **Mask hints** from solutions in train set [3]
- Issue: computing the E-PLL requires a **complete** assignment
- Our solution: **data imputation**
  - Solve predicted CFN to infer missing variables

Symbol grounding problem

Visual sudoku: Learn symbols and rules simultaneously



- Solver able to correct digit mis-classification

Approach	Solved	Training (*h*)
Rec. Trans. ICLR23[13]	75.6%	5.1
NeSy. Prog. NeurIPS23[7]	92.2–94.4%	4.7
E-PLL IJCAI23[4]	93.4%	3.2

Learning to reason on real-world problems

Key properties of learning CFN with Emmental PLL

- **Contextual** optimization
- **Scales** to large instances
- *A posteriori* **control** (adding constraints or criteria)

Real-world applications

- Preference acquisition & car configuration [2]
- Estimating gene regulation networks [10]
- **Learning the laws of protein design**

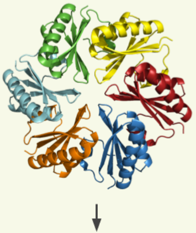
Application to a real-world problem

Learning the laws of protein design

Cost function = **pairwise interaction** score [11]

- Main challenges:
  - Train set **up to 10,000 variables**, variable size
  - **Varying context** = input structure
  - Observe **one of many** possible solutions
- Intractable inference → use an approximate solver [5]
- Outperforms existing decomposable score functions

	Rosetta [9]	Our
Similarity (↑)	17.9%	**33.0%**



SSNAIGLIETKGYVAA...

Experimentally validated

Acknowledgment

- My PhD **supervisors**: Sophie Barbe & Thomas Schiex
- My teams in **TBI** and **MIAT**
- **CALMIP** & **IDRIS** & **GenoToul** for computational resources
- The **organizers** of the workshop

Thanks for your attention!

BioEco
Graduate School
Biotechnology Building Bio-Based Economy

ANITI
ARTIFICIAL & NATURAL INTELLIGENCE
TOULOUSE INSTITUTE

This work has been supported by the Agence Nationale de la Recherche (ANR) [grant ANR-18-EURE-0021]

References I

[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] 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.

[3] Oscar Chang et al. “Assessing SATNet’s Ability to Solve the Symbol Grounding Problem”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1428–1439. URL: https://proceedings.neurips.cc/paper/2020/file/0ff8033cf9437c213ee13937b1c4c455-Paper.pdf.

References II

[4] 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.

[5] Valentin Durante, George Katsirelos, and Thomas Schiex. "Efficient low rank convex bounds for pairwise discrete Graphical Models". In: *Thirty-ninth International Conference on Machine Learning*. July 2022.

[6] Barry Hurley et al. "Multi-language evaluation of exact solvers in graphical model discrete optimization". In: *Constraints* 21 (2016), pp. 413–434.

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

References III

[8] 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.

[9] Hahnbeom Park et al. “Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules”. In: *Journal of Chemical Theory and Computation* 12.12 (2016), pp. 6201–6212.

[10] Youngsuk Park et al. “Learning the network structure of heterogeneous data via pairwise exponential Markov random fields”. In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1302–1310.

[11] Seydou Traoré et al. “A new framework for computational protein design through cost function network optimization”. In: *Bioinformatics* 29.17 (July 2013), pp. 2129–2136. ISSN: 1367-4803.

References IV

[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.

References V

[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.

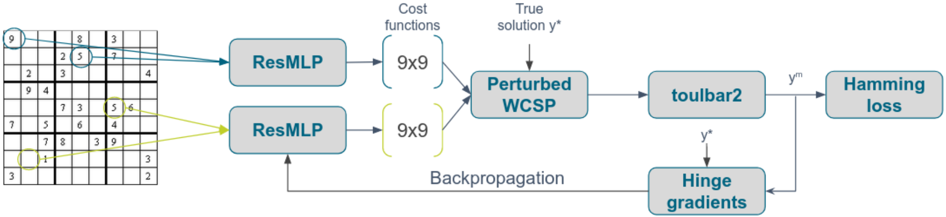
Annex: Details on the Hinge loss

$$L(\mathbf{y},\mathbf{y}^*) = Hamming(\mathbf{y},\mathbf{y}^*) = \sum_{i=1}^{n} \alpha \mathbb{1}[y_i \neq y_i^*] \text{ with } \alpha \in \mathbb{R}_+^*$$

$$Hinge(\omega,\mathbf{y}) = \max_{\mathbf{t}\in D^{\mathbf{Y}}} [L(\mathbf{y},\mathbf{t}) + (N(\omega)(\mathbf{y}) - N(\omega)(\mathbf{t}))]$$

For a pairwise-decomposable loss L, the Hinge loss becomes:

$$Hinge(\omega,\mathbf{y}) = N(\omega)(\mathbf{y}) - \underbrace{\min_{\mathbf{t}\in D^{\mathbf{Y}}} [N(\omega)(\mathbf{t}) - L(\mathbf{y},\mathbf{t})]}_{\text{arg min}=\mathbf{y}^m}$$


Cost functions
True solution y*
ResMLP
9x9
ResMLP
9x9
Perturbed WCSP
toulbar2
y^m
Hamming loss
y*
Hinge gradients
Backpropagation