Scalable Coupling of Deep Learning with Logical Reasoning: from Sudoku to new functional molecules

Thomas Schiex

Joint work with S. Barbe, M. Defresne (PhD student)

Human reasoning and scientific discovery

Inductive and deductive reasoning

- From observations (solutions) we construct a theory ($F = m\gamma$)
- We then use the theory to make predictions and design objects
- Until the theory is proven to be incorrect

Sudoku grid with solution

Protein structure with its sequence

The theory is written as binary Cost Function Network

Human reasoning and scientific discovery

Inductive and deductive reasoning

- From observations (solutions) we construct a theory ($F = m\gamma$)
- We then use the theory to make predictions and design objects
- Until the theory is proven to be incorrect

Sudoku grid with solution

Protein structure with its sequence

The theory is written as binary Cost Function Network

Human reasoning and scientific discovery

Inductive and deductive reasoning

- From observations (solutions) we construct a theory ($F = m\gamma$)
- We then use the theory to make predictions and design objects
- Until the theory is proven to be incorrect

Sudoku grid with solution

Protein structure with its sequence

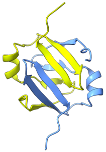
The theory is written as binary Cost Function Network

Inductive and deductive reasoning

- From observations (solutions) we construct a theory ($F = m\gamma$)
- We then use the theory to make predictions and design objects
- Until the theory is proven to be incorrect

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

Sudoku grid with solution



Protein structure with its sequence

The theory is written as binary Cost Function Network

Cost Function Networks

Binary Cost Function Network

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

Costs and probabilities

- The cost $C(t)$ of an assignment t is the sum of all cost functions on t
- Toulbar2 solves the Weighted Constraint Satisfaction Problem — $\min_t C(t)$
- A CFN defines a probability distribution: $P(t) \propto \exp(-C(t))$ — Markov Random Fields
- Normalizing constant is #P-hard to compute

Cost Function Networks

Binary Cost Function Network

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

Costs and probabilities

- The cost $C(t)$ of an assignment t is the sum of all cost functions on t
- Toulbar2 solves the Weighted Constraint Satisfaction Problem — $\min_t C(t)$
- A CFN defines a probability distribution: $P(t) \propto \exp(-C(t))$ — Markov Random Fields
- Normalizing constant is #P-hard to compute

Injecting ML / intuition[1]

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

Φ

Injecting ML / intuition[1]

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

$\Phi \longrightarrow$ Neural net

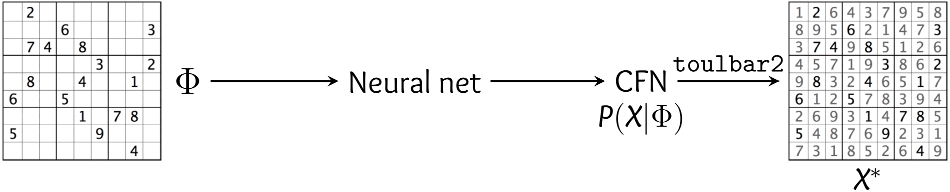
Injecting ML / intuition[1]

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

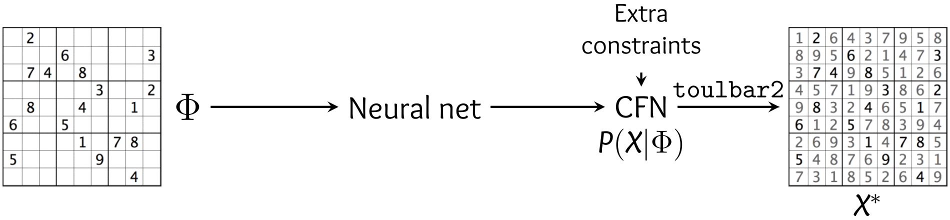
$\Phi \longrightarrow$ Neural net $\longrightarrow$ CFN

$P(X|\Phi)$

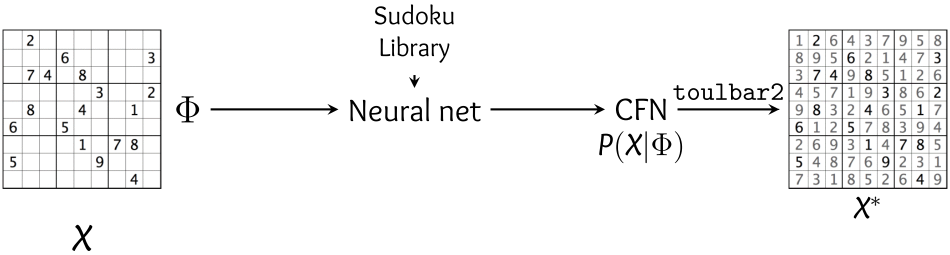
Injecting ML / intuition[1]



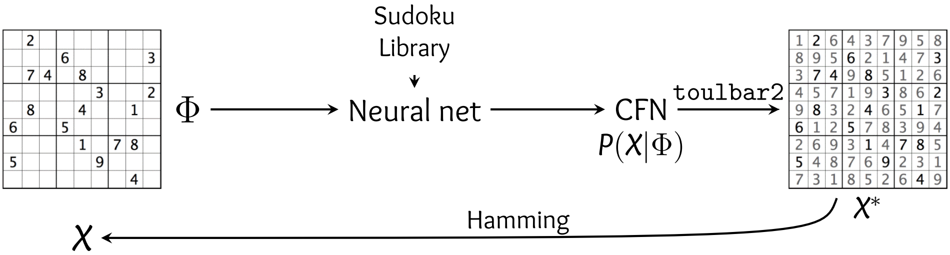
Injecting ML / intuition[1]

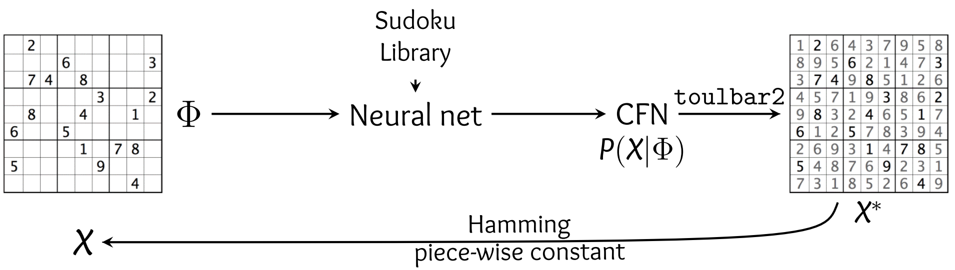


Injecting ML / intuition[1]



Injecting ML / intuition[1]

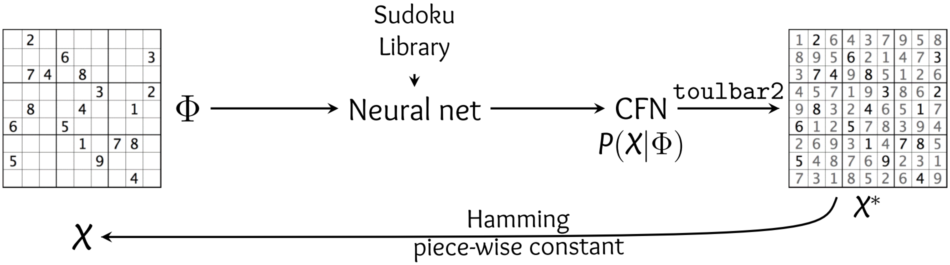




Issues

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

Injecting ML / intuition[1]

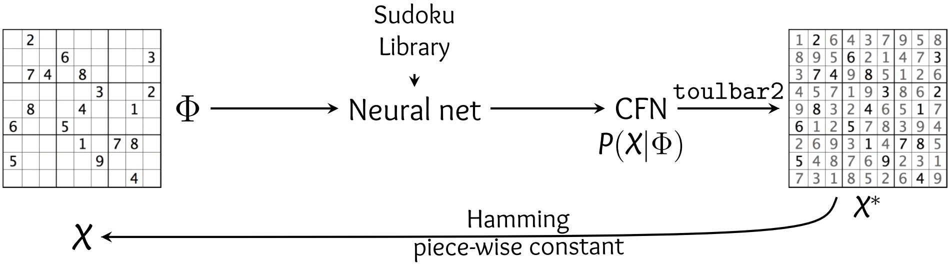


Our solution

IJCAI'2023

- Introduced a dedicated loss: the E-Pseudo Log Likelihood (Defresne et al. 2023)
- Kicked the solver out of the training loop (scalable training)

Injecting ML / intuition[1]



Approach	Architecture	Acc.	Grids	Training set
RRN NeurIPS18	GNN	96.6%	Hard	180,000
SATNet ICML19	Relaxation	99.8%	Easy	9,000
Hybrid IJCAI23	E-PLL	**100%**	Hard	**200**
Symbolic IJCAI23	MaxSAT	**100%**	Hard	**200**

Pseudo-Loglikelihood with holes

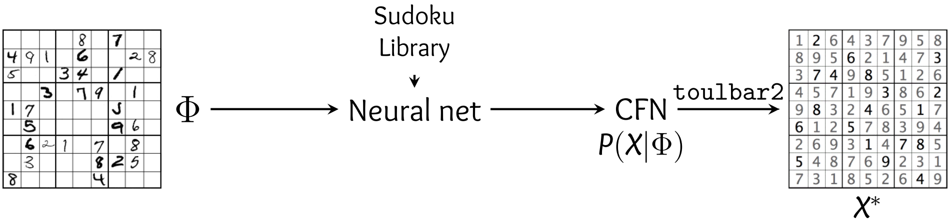
PLL from solution s and current predicted CFN

- $-\log(P(s))$ would be a nice loss — intractable
- For each variable x_i, we mask x_i in the solution
- Compute $P(x_i|X_{-i})$ — easy
- Use $-\sum_{x_i} \log(P(x_i|X_{-i}))$ as the pseudo-loss
- Excellent asymptotic properties — statistically consistent
- Does not work in practice (high costs) — Montanari and Pereira 2009

The Emmental Pseudo-loglikelihood

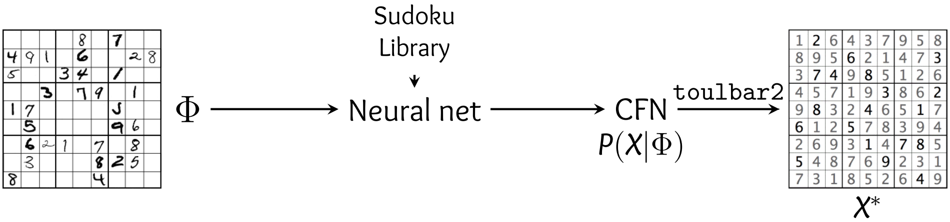
- When we compute $P(x_i|X_{-i})$
- We ignore a random fraction of the neighbors
- Strongly accelerates stochastic gradient optimization

Learning to play Visual Sudoku



Simultaneously learns to recognize digits and to play the Sudoku

SATNet	Theoretical (no corrections)	Hybrid
63.2 %	74.2%	$94.1 \pm 0.8\%$



Simultaneously learns to recognize digits and to play the Sudoku

SATNet	Theoretical (no corrections)	Hybrid
63.2 %	74.2%	**94.1** $\pm$ 0.8%

Reading numbers without cheating (grounding)

Grounding issue: a nasty form of data leakage (Chang et al. 2020)

- The training set contains images and associated decoded digits (hints).
- Corrected in (Topan et al. 2021) using a complex architecture (GAN+clustering+Distillation)

Easily solvable in our architecture

- In the training set, we replace hints with a new "missing value" (0)
- We connect LeNet output to the CFN generator input
- LeNet learns a permutation of the digits
- The CFN-generator corrects this permutation in its output
- Much longer training (few hours)

Reading numbers without cheating (grounding)

Grounding issue: a nasty form of data leakage (Chang et al. 2020)

- The training set contains images and associated decoded digits (hints).
- Corrected in (Topan et al. 2021) using a complex architecture (GAN+clustering+Distillation)

Easily solvable in our architecture

- In the training set, we replace hints with a new"missing value" (0)
- We connect LeNet output to the CFN generator input
- LeNet learns a permutation of the digits
- The CFN-generator corrects this permutation in its output
- Much longer training (few hours)

Learning to play Many-Solutions Sudokus

Sudokus have only one solution (single target for DL)

- Existing DL architectures fail on many-solutions Sudokus (Nandwani et al. 2021)
- 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)

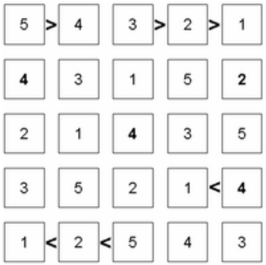
- Existing DL architectures fail on many-solutions Sudokus (Nandwani et al. 2021)
- Corrected using a Reinforcement learning approach
- Training set with 5 solutions per instance
- Ability to generate additional solutions

Our architecture directly learns how to solve many-solutions Sudokus

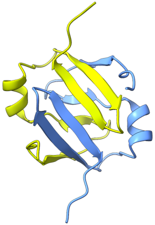
Learning to play Futoshiki

Sudoku is easy, only one type of constraints

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



Learning to design proteins: Effie



X

PDB
170,000 (Φ, X)
↓
Design constraints
↓

$$\Phi \longrightarrow \text{Neural net} \longrightarrow \underset{P(X|\Phi)}{\text{CFN}} \xrightarrow{\texttt{toulbar2}} X^*$$

Optimizing a complete protein sequence

Full redesign of large proteins in the test set

- Guaranteed `toulbar2` solution expensive
- Using LR-BCD instead (Durante et al. 2022)

Outperforms all-atoms XIXth-century physics

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

Approach	Rosetta	Effie
NSR	17.9%	32.8%

Optimizing a complete protein sequence

Full redesign of large proteins in the test set

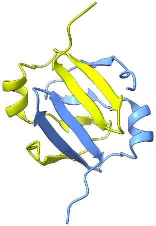
- Guaranteed `toulbar2` solution expensive
- Using LR-BCD instead (Durante et al. 2022)

Outperforms all-atoms XIXth-century physics

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

Approach	Rosetta	Effie
NSR	17.9%	32.8%

Effie vs. ProteinMPNN (Dauparas et al. 2022)

GPT-style  $\Phi \longrightarrow$ Autoregressive NN $\longrightarrow X_1, X_2, \ldots$

$$\text{PDB} \downarrow$$

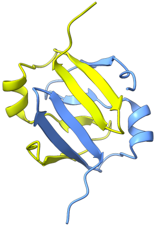
$$P(X_i | X_{i-1}, \ldots, X_1, \Phi)$$

Pros and cons

- Heuristic score instead of NP-hard solving
- Capacity to capture higher-order interactions
- Limited control for design constraints

	ProteinMPNN	Effie
NSR	45.9%	48.4%

Effie vs. ProteinMPNN (Dauparas et al. 2022)

GPT-style 

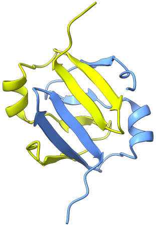
$$\Phi \longrightarrow \underset{P(X_i|X_{i-1},\ldots,X_1,\Phi)}{\overset{\text{PDB}\ \downarrow}{\text{Autoregressive NN}}} \longrightarrow X_1, X_2, \ldots$$

Pros and cons

- Heuristic score instead of NP-hard solving
- Capacity to capture higher-order interactions
- Limited control for design constraints

	ProteinMPNN	Effie
NSR	45.9%	48.4%

GPT-style



$\Phi \longrightarrow$ Autoregressive NN $\longrightarrow X_1, X_2, \ldots$

PDB $\downarrow$

$P(X_i | X_{i-1}, \ldots, X_1, \Phi)$

Pros and cons

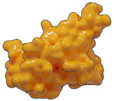
- Heuristic score instead of NP-hard solving
- Capacity to capture higher-order interactions
- Limited control for design constraints

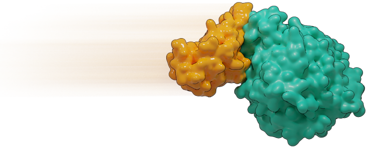
	ProteinMPNN	Effie
NSR	45.9%	**48.4%**

Predicting SARS-CoV2 variants

(Colom et al. 2024)

RBD





ACE2

Enumerate CoViD variants with a bounded number of mutations

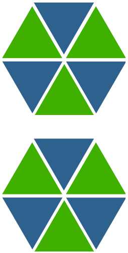
- Uses only the initial March 2020 RBD-ACE2 structure + Effie/toulbar2
- Relies on (Montalbano et al. 2022) global constraint to bound mutations
- Predicts all the first SARS-CoV2 VoCs ($\alpha, \beta, \gamma, \delta, \kappa, \iota, \lambda$ and μ)
- In a few seconds, on one CPU-thread.

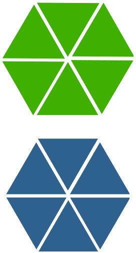
Not achievable by pure autoregressive models (ProteinMPNN)

Design of an enzyme organizing platform

Design of an heteromeric hexamer

- Design and that self-assemble as but not as or
- Physics+logic: requires bi-level optimization (NP^{NP}-complete) (Vucinic et al. 2020)
- Compare Effie+tb2 (NP-complete) with ProteinMPNN, bi-criteria (Buchet et al. 2024)





Design of an enzyme organizing platform

Design of an heteromeric hexamer

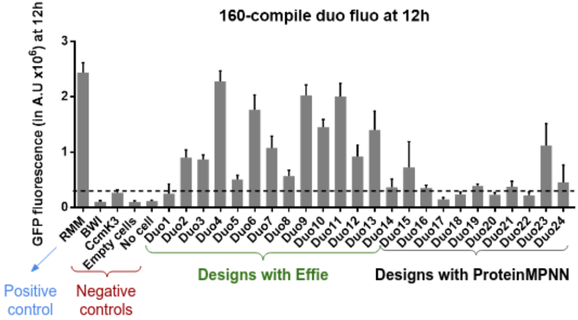
- Design  and  that self-assemble as  but not as  or 
- Physics+logic: requires bi-level optimization (NP^{NP}-complete) (Vucinic et al. 2020)
- Compare Effie+tb2 (NP-complete) with ProteinMPNN, bi-criteria (Buchet et al. 2024)

How often is better than ?

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 CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

Conclusions

A Neural Net, a CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

Conclusions

A Neural Net, a CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

Conclusions

A Neural Net, a CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

Conclusions

A Neural Net, a CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

Conclusions

A Neural Net, a CFN and a WCSP solver in a hybrid autoencoder

- A hybrid generic Generative AI that benefits from each component
- Neural Network: ideal to extract a representation of $P(X|\Phi)$ from raw inputs
- Represented as a CFN in a fully explorable and controllable latent layer
- Using decoding by discrete reasoning (toulbar2)
- All this with scalable training

Questions:

What is missing in this architecture to make you happy?

When is Predict-and-optimize useful?

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

Besag, Julian (1975). "Statistical Analysis of Non-Lattice Data". In: Journal of the Royal Statistical Society: Series D (The Statistician) 24.3, pp. 179–195.

Bessiere, Christian et al. (Aug. 2023). "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, pp. 1876–1883. DOI: 10.24963/ijcai.2023/208. URL: https://doi.org/10.24963/ijcai.2023/208.

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

Chang, Oscar et al. (2020). "Assessing SATNet's Ability to Solve the Symbol Grounding Problem". In: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, virtual. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/0ff8033cf9437c213ee13937b1c4c455-Abstract.html.

Colom, Mireia Solà et al. (Aug. 2024). "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, e5109. DOI: 10.1002/pro.5109. URL: https://hal.science/hal-04646616.

Dauparas, J. et al. (2022). "Robust deep learning-based protein sequence design using ProteinMPNN". In: Science 378.6615, pp. 49–56. DOI: 10.1126/science.add2187.

References II

Defresne, Marianne et al. (2023). “Scalable Coupling of Deep Learning with Logical Reasoning”. In: 32nd International Joint Conference on Artificial Intelligence, IJCAI 2023. Macao, SAR, China: ijcai.org, pp. 3615–3623. DOI: 10.24963/IJCAI.2023/402.

Durante, Valentin et al. (July 2022). “Efficient low rank convex bounds for pairwise discrete Graphical Models”. In: Thirty-ninth International Conference on Machine Learning.

Montalbano, Pierre et al. (2022). “Multiple-Choice Knapsack Constraint in Graphical Models”. In: Proc. of CPAIOR’22.

Montanari, Andrea and Jose Pereira (2009). “Which graphical models are difficult to learn?” In: Advances in Neural Information Processing Systems. Ed. by Y. Bengio et al. Vol. 22. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2009/file/22fb0cee7e1f3bde58293de743871417-Paper.pdf.

Nandwani, Yatin et al. (2021). “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. URL: https://openreview.net/forum?id=ATp1nW2FuZL.

Topan, Sever et al. (2021). “Techniques for Symbol Grounding with SATNet”. In: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, virtual. Ed. by Marc’Aurelio Ranzato et al., pp. 20733–20744. URL: https://proceedings.neurips.cc/paper/2021/hash/ad7ed5d47b9baceb12045a929e7e2f66-Abstract.html.

Vucinic, Jelena et al. (2020). “Positive multistate protein design”. In: Bioinformatics 36.1, pp. 122–130.