# Learning how to design biomolecules using a neuro-symbolic architecture

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

INRAE

ANITI

Université de Toulouse

## Inductive and deductive reasoning

- From observations 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 pairwise Cost Function Network

# Human reasoning and scientific discovery

## Inductive and deductive reasoning

- From observations 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 pairwise Cost Function Network

# Human reasoning and scientific discovery

## Inductive and deductive reasoning

- From observations 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 pairwise Cost Function Network

## Inductive and deductive reasoning

- From observations 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 pairwise Cost Function Network

# Cost Function Networks

## Pairwise Cost Function Network

(Ising/Potts/Graphical model)

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

## Costs and probabilities

- The cost $E(t)$ of an assignment $t$ is the sum of all cost functions on $t$
- Toulbar2 finds $\operatorname{argmin}_t E(t)$ and proves optimality.
- A CFN defines a probability distribution: $P(t) \propto \exp(-E(t))$ — Markov Random Fields
- Normalizing constant is #P-hard to compute

# Cost Function Networks

## Pairwise Cost Function Network

(Ising/Potts/Graphical model)

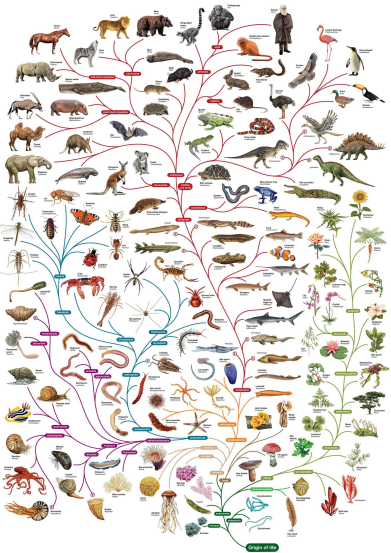
- A set $X$ of variables — $n$ variables
- Variable $x_i$ has domain $D_i$ — max. size $d$
- a set of cost/energy functions — $e_{ij} : D_i \times D_j \rightarrow \mathbb{R} \cup \{\infty\}$

## Costs and probabilities

- The cost $E(t)$ of an assignment $t$ is the sum of all cost functions on $t$
- Toulbar2 finds $\mathrm{argmin}_t E(t)$ and proves optimality.
- A CFN defines a probability distribution: $P(t) \propto \exp(-E(t))$ — Markov Random Fields
- Normalizing constant is #P-hard to compute

# The engines of Life

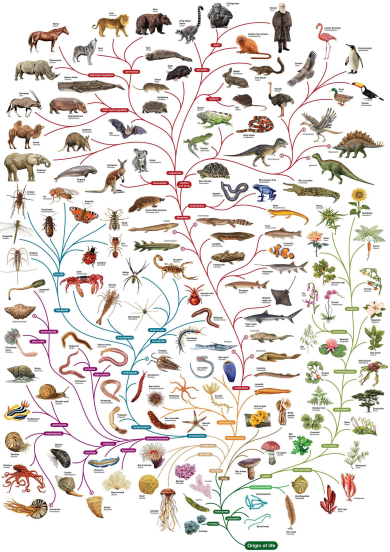
- Most active molecules of life (virus to humans)
- Useful in health to green chemistry



# Designing Proteins

## The engines of Life

- Most active molecules of life (virus to humans)
- Useful in health to green chemistry

# Protein folding

DVVGKVVDGKDD···GVKVGDKVKVKKV

Organizes different types of atoms in 3D

Sequence ⇝ Structure ⇝ Function

# Protein folding

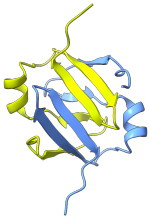
DVVGKVVDGKDD···GVKVGDKVKVKKV

Organizes different types of atoms in 3D

Sequence ⇝ Structure ⇝ Function

# Protein folding

DVVGKVVDGKDD···GVKVGDKVKVKKV



Organizes different types of atoms in 3D

Sequence ⇝ Structure ⇝ Function

# Protein folding

$\chi$
Amino acid sequence
(20 letters alphabet)

$\Phi$
Continuous SE(3)-invariant
3D structure

Organizes different types of atoms in 3D

Sequence $\rightsquigarrow$ Structure $\rightsquigarrow$ Function

# Protein folding

$\chi$
Amino acid sequence
(20 letters alphabet)

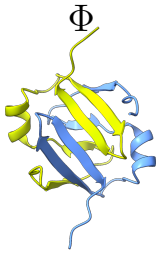
$\Phi$
Continuous SE(3)-invariant
3D structure

Organizes different types of atoms in 3D

Sequence $\rightsquigarrow$ Structure $\rightsquigarrow$ Function

# Designing Proteins with physics

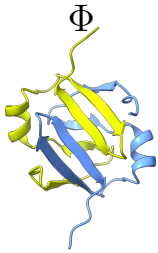
$\Phi$



A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Com. ACM 20, B. Donald et al.

# Designing Proteins with physics




Physics

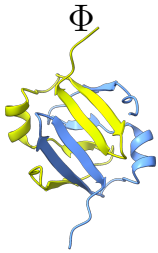
Rosetta
`beta_nov16`

Binary CFN
$P(X|\Phi)$

A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Comp. ACM 20, B. Donald et al.

# Designing Proteins with physics



$\Phi$

Physics

Rosetta
`beta_nov16`

Binary CFN
$P(X|\Phi)$

Reasoning

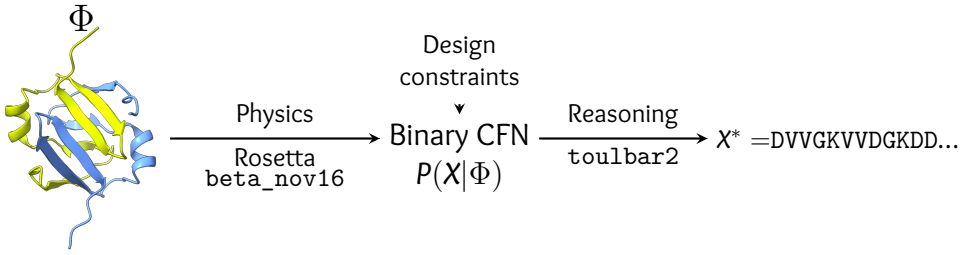
`toulbar2`

$X^*$ =`DVVGKVVDGKDD...`

A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Comp. ACM 20, B. Donald et al.

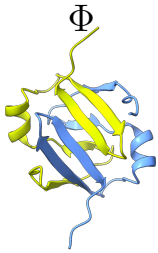
# Designing Proteins with physics




A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Comm. ACM 20, B. Donald et al.

# Designing Proteins with physics



$\Phi$

Physics

Rosetta
`beta_nov16`

Design constraints

Binary CFN

$P(X|\Phi)$

Reasoning

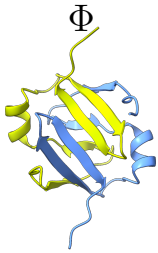
`toulbar2`

$X^*$ =`DVVGKVVDGKDD...`

A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Com. ACM-20, B. Donald et al.

# Designing Proteins with physics



$\Phi$

Physics

Rosetta
`beta_nov16`

Design constraints

Binary CFN
$P(X|\Phi)$

Reasoning

`toulbar2`

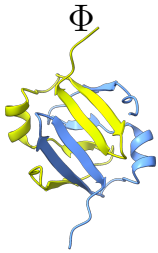
$X^*$ =`DVVGKVVDGKDD...`

XIX$^{th}$ century physics

A quite successful all physics+logic generative process

*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Com. ACM-20, B. Donald et al.

# Designing Proteins with physics



$\Phi$

Physics

Rosetta
`beta_nov16`

Design constraints

Binary CFN
$P(X|\Phi)$

Reasoning

`toulbar2`

$X^*$ =`DVVGKVVDGKDD...`

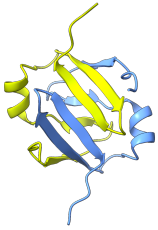
XIX$^{th}$ century physics

NP-hard

A quite successful all physics+logic generative process

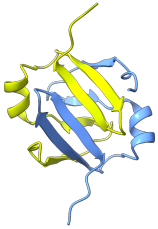
*The Toulbar package [...] significantly improved the state-of-the-art efficiency for protein design*
Com. ACM-20, B. Donald et al.

# Injecting ML / intuition[1]



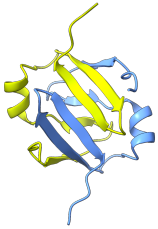
$\Phi$

# Injecting ML / intuition[1]



$\Phi \longrightarrow$ Neural net

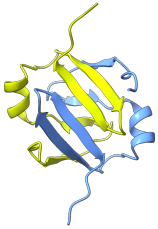
# Injecting ML / intuition[1]



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

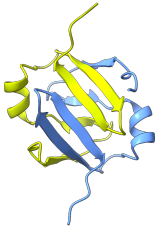
$P(X|\Phi)$

# Injecting ML / intuition[1]



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

# Injecting ML / intuition[1]

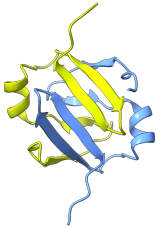


$\Phi$ → Neural net → CFN → $X^*$ (toulbar2)

Design constraints → CFN

$P(X|\Phi)$

# Injecting ML / intuition[1]

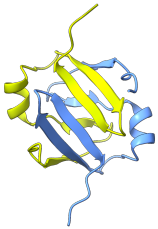


$X$

PDB
195,000 $(\Phi, X)$

$\Phi$ → Neural net → CFN $P(X|\Phi)$ → (toulbar2) → $X^*$

# Injecting ML / intuition[1]



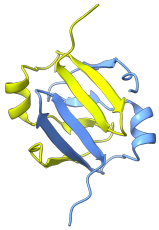
PDB
195,000 $(\Phi, X)$

$\Phi$ → Neural net → CFN $P(X|\Phi)$ —toulbar2→ $X^*$

Hamming

$X$

# Injecting ML / intuition[1]



PDB
195,000 $(\Phi, X)$

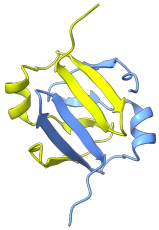
$\Phi$ → Neural net → CFN $P(X|\Phi)$ —toulbar2→ $X^*$

$X$ ← Hamming
piece-wise constant

## Issues

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

# Injecting ML / intuition[1]



PDB
195,000 $(\Phi, X)$

$\Phi$ → Neural net → CFN $P(X|\Phi)$ —toulbar2→ $X^*$

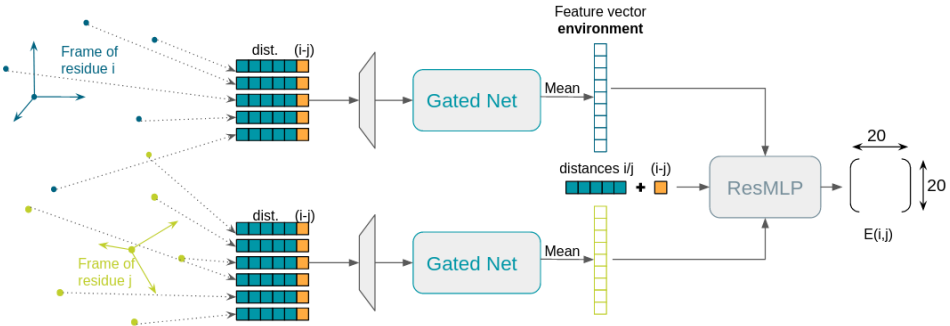
$X$ ← Hamming
piece-wise constant

## Our solution

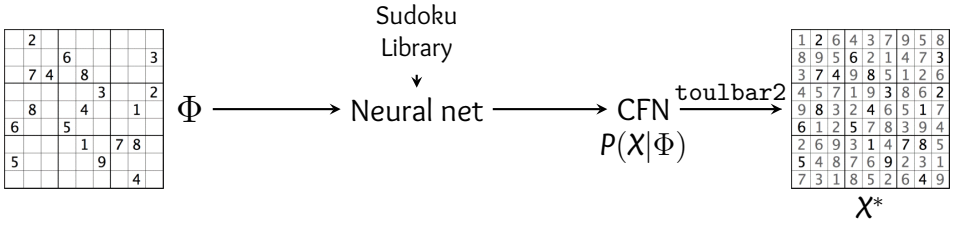
IJCAI'2023

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

# Protein Design architecture

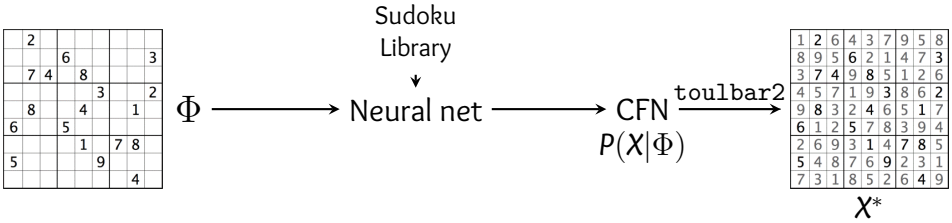


# Learning to play Sudoku



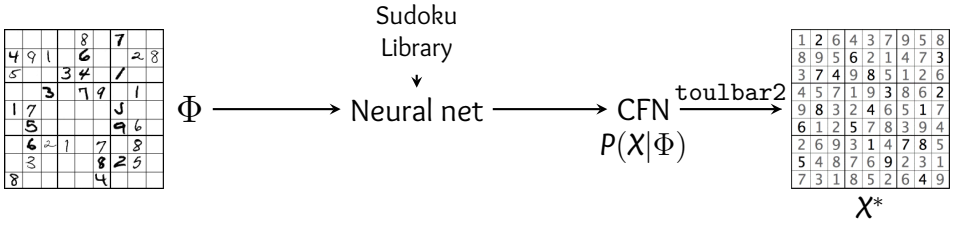

| Approach | Architecture | Acc. | Grids | Training set |
|---|---|---|---|---|
| RRN NeurIPS18 | GNN | 96.6% | Hard | 180,000 |
| SATNet ICML19 | Relaxation | 99.8% | Easy | 9,000 |
| Hybrid IJCAI23 | E-PLL | **100%** | Hard | **200** |

# Learning to play Sudoku




| 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** |

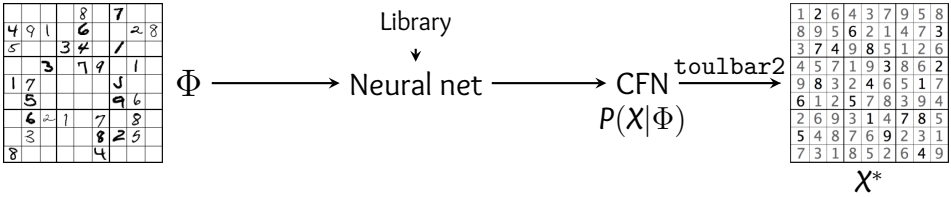
# Learning to play Visual Sudoku




Simultaneously learns to recognize digits and to play the Sudoku

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

# Learning to play Visual Sudoku



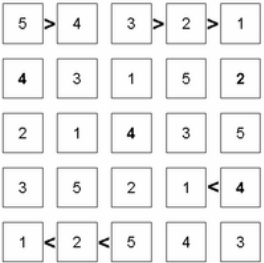

Simultaneously learns to recognize digits and to play the Sudoku

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

# Learning to play Futoshiki

## Sudoku is easy, only one type of constraints

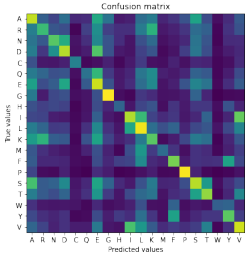
- Our architecture directly learns how to play Futoshiki
- Includes both difference and inequality constraints
- Perfect solving, expected constraints learned

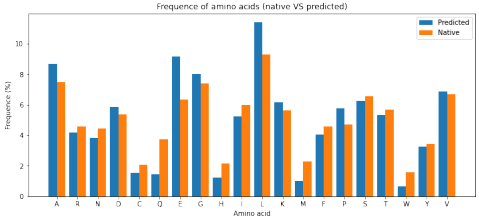


# Learning to design proteins: Effie

## Recovering amino acid properties

- Correctly predicts 51% of amino acids from their environment




## Zero-shot prediction of the effect of single mutations

- 79% accuracy on ATOM3D benchmark
- $0.4$ correlation stability score/predicted energy (Rocklin et al. 2017)

# 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 XIX$^{th}$-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)

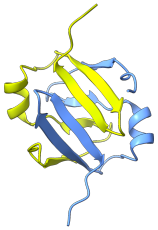
## Outperforms all-atoms XIX$^{th}$-century physics

- 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 $\longrightarrow X_1, X_2, \ldots$

PDB

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

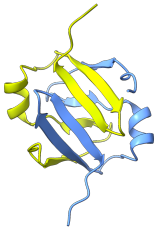
## Pros and cons

- heuristic guide 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



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

PDB

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

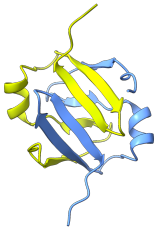
## Pros and cons

- heuristic guide 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



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

PDB

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

## Pros and cons

- heuristic guide 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)


## 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 $\mu$)
- In a few seconds, on one CPU-thread.

Not achievable by pure autoregressive models (ProteinMPNN)

# Design of an enzyme organizing platform

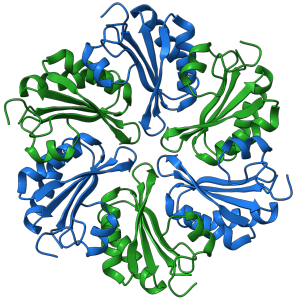
## Design of an heteromeric hexamer

- 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)

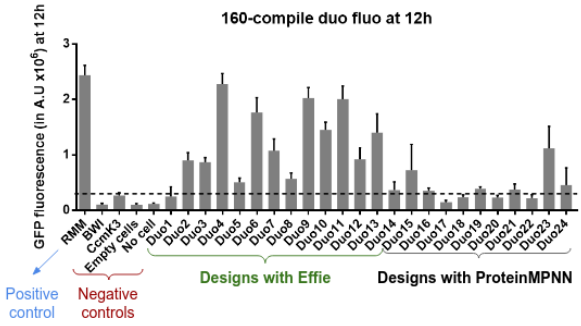


# How often is better than ?

| 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 CFN prover 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

# Conclusions

## A Neural Net, a CFN and a CFN prover 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

# Conclusions

## A Neural Net, a CFN and a CFN prover 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

# Conclusions

## A Neural Net, a CFN and a CFN prover 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

# Conclusions

## A Neural Net, a CFN and a CFN prover 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

# 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


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.

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.

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.

# References II


- Montalbano, Pierre et al. (2022). “Multiple-Choice Knapsack Constraint in Graphical Models”. In: Proc. of CPAIOR’22.
- Rocklin, Gabriel J et al. (2017). “Global analysis of protein folding using massively parallel design, synthesis, and testing”. In: Science 357.6347, pp. 168–175.
- Vucinic, Jelena et al. (2020). “Positive multistate protein design”. In: Bioinformatics 36.1, pp. 122–130.