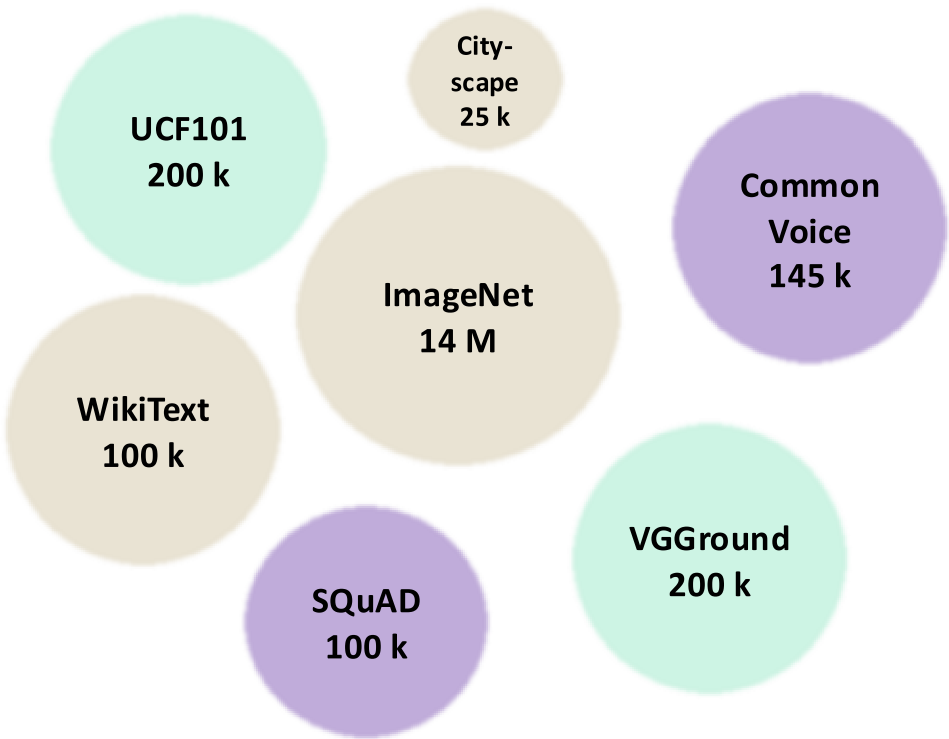
PURDUE UNIVERSITY

DisjunctiveNet.jl: Neural Symbolic Learning via Differentiable Convexified Optimization Layers

Shraman Pal, **Can Li**
Davidson School of Chemical Engineering
Purdue University

Limitations of Deep Neural Networks

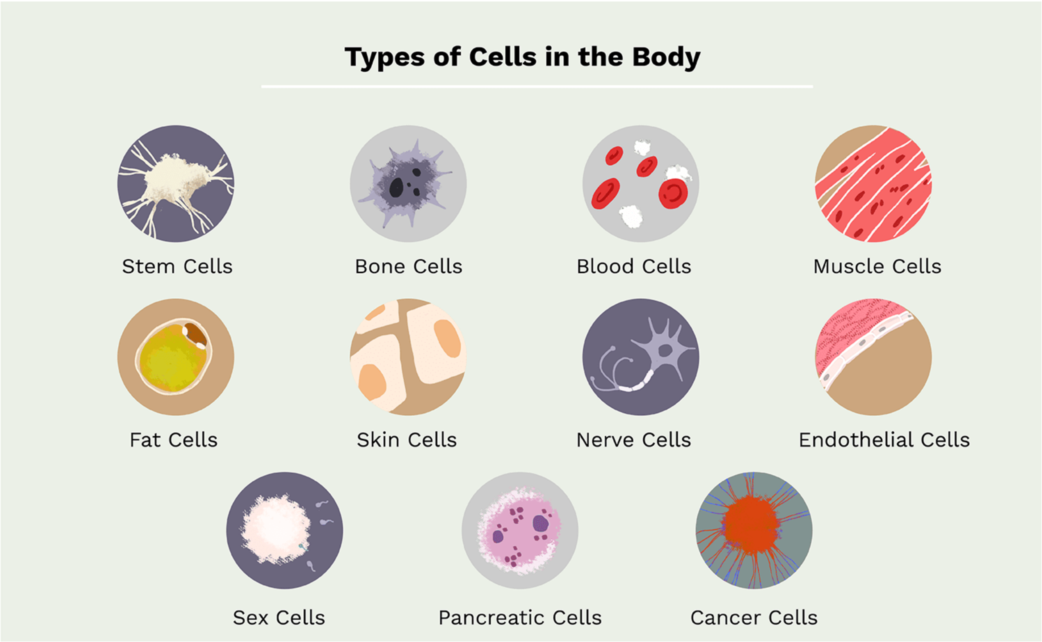
- **Data efficiency**
  - Works well in a large data regime (image, text, audio).
  - Scientific and engineering data is scarce and expensive to obtain.

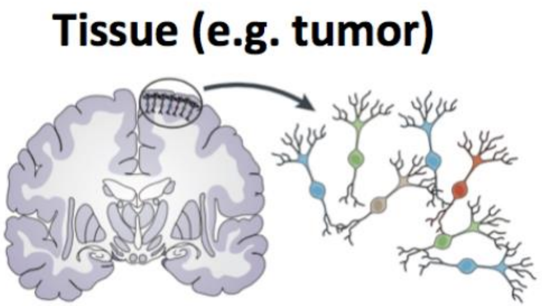


Generated with ChatGPT

Cell Classification

- **Motivating application:** inspect the concentration of cancer cells before and after treatment.

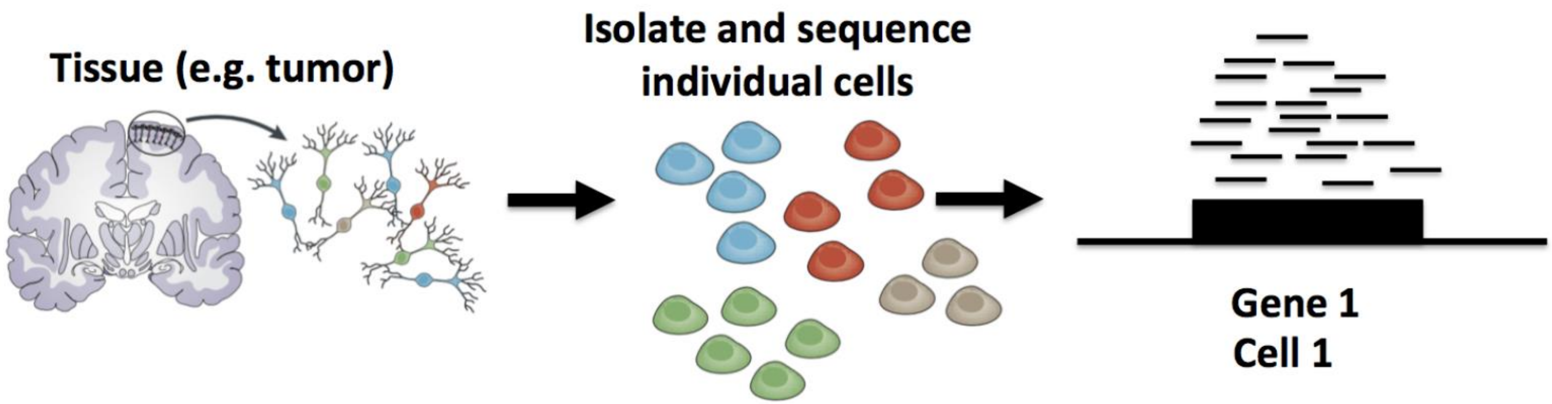




https://www.thoughtco.com/types-of-cells-in-the-body-373388
https://learn.gencore.bio.nyu.edu/single-cell-rnaseq/

Single Cell RNA Sequencing

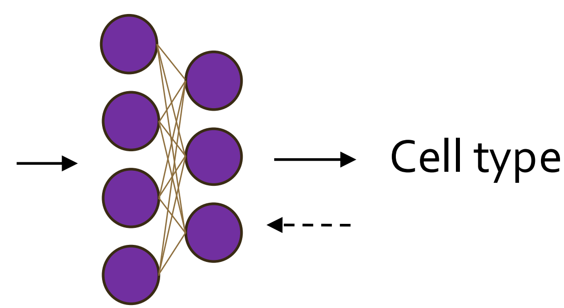
- scRNA-seq: experimental methods to measure the number of messenger RNA in every single cell.
- Gene expression is inferred by measuring the number of messenger RNA molecules corresponding to each gene.



Read Counts

	Cell 1	Cell 2	...
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
...			

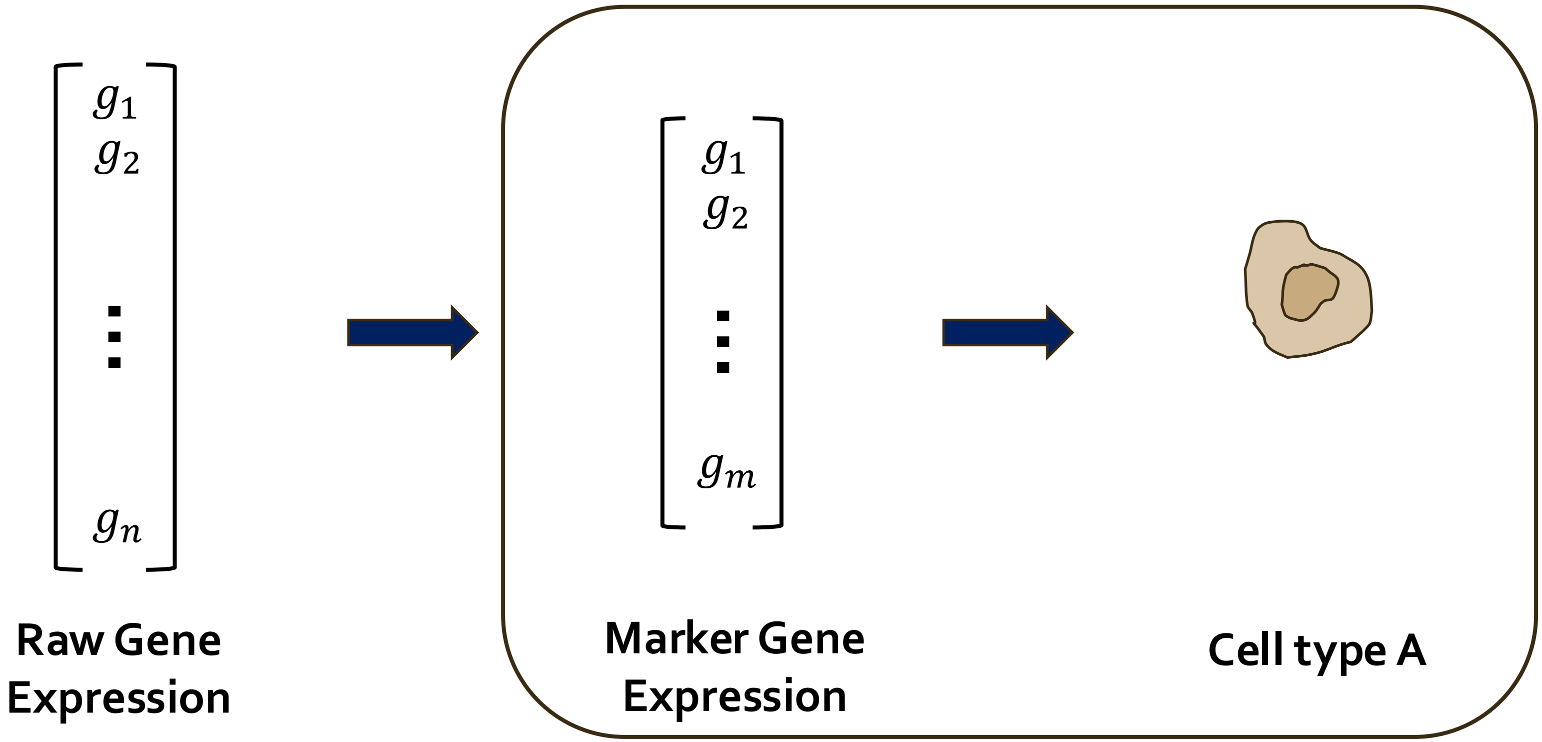
$$\begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$



https://learn.gencore.bio.nyu.edu/single-cell-rnaseq/

Cell Annotation

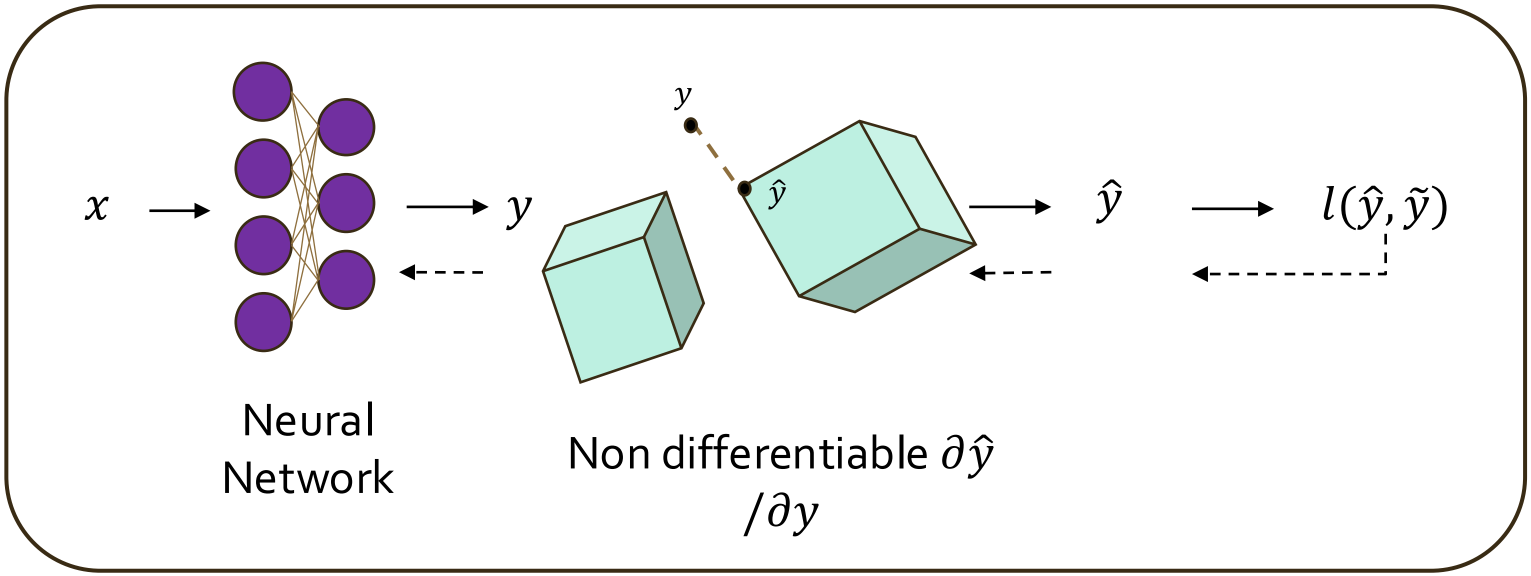
- Data are expensive to collect: each sample costs around $3,000.
- Prior knowledge: marker genes are the highly expressed genes for a given cell type



$$G_r(u) \geq \tau_r \;\Rightarrow\; \bigvee_{c \in \mathcal{C}_r} y_c \geq \rho_r,$$

Challenges in Projection to Disjunction

- Feasible region is **nonconvex and discontinuous**
- Result in **non-differentiable** projection



Goal of the Proposed Method

- Embed input-dependent hard mixed-integer linear constraints.

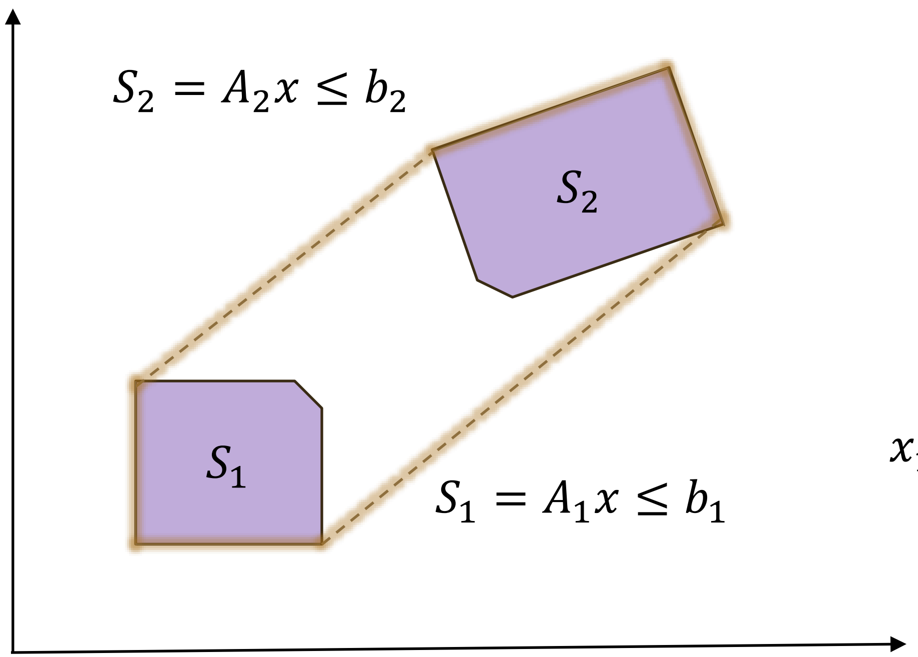
$$\mathbb{I}[x \in \mathcal{A}_r] \Rightarrow \mathbb{I}[y \in \mathcal{C}_r(x)], \qquad r = 1, \ldots, R,$$

$$\mathcal{C}_r(x) = \bigcup_{j=1}^{m_r} \mathcal{S}_{rj}(x), \quad \mathcal{S}_{rj}(x) = \{y : A_{rj}(x)y \leq b_{rj}(x)\}.$$

Method	Logic	Linear	Mix	Hard	Input dep
OptNet (Amos & Kolter, 2017)	×	✓	×	✓	✓
CVXPYlayers (Agrawal et al., 2019)	×	✓	×	✓	✓
DC3 (Donti et al., 2021)	×	✓	×	×	✓
Primal dual (Park & Van Hentenryck, 2023)	×	✓	×	×	✓
KKTHardNet (Iftakher et al., 2025)	×	✓	×	✓	✓
DecisionRuleNet (Constante-Flores et al., 2025)	×	✓	×	✓	✓
Homeomorphic projection (Liang et al., 2024)	×	✓	×	✓	✓
FSNet (Nguyen & Donti, 2025)	×	✓	×	✓	✓
Semantic Loss (Xu et al., 2018)	✓	✓	×	×	×
DL2 (Fischer et al., 2019)	✓	✓	✓	×	✓
LTN (Badreddine et al., 2022)	✓	×	✓	×	✓
DeepProbLog (Manhaeve et al., 2018)	✓	✓	×	×	✓
MultiplexNet (Hoernle et al., 2022)	✓	×	✓	✓	×
Straight through estimator (Sahoo et al., 2023)	✓	✓	✓	✓	×
SATNet (Wang et al., 2019)	✓	×	×	×	×
LP relaxation of ILP (Wilder et al., 2019)	✓	✓	✓	×	×
MIPaaL (Ferber et al., 2020)	✓	✓	✓	×	×
DYS-Net (McKenzie et al., 2024)	✓	✓	✓	×	×
Perturbed optimizer (Berthet et al., 2020)	✓	✓	✓	✓	×
CombOptNet (Paulus et al., 2021)	✓	✓	✓	×	×
ILP inference (Roth & Yih, 2005)	✓	✓	✓	×	✓
DRL (Stoian & Giunchiglia, 2025)	✓	×	✓	✓	×
Ours	✓	✓	✓	✓	✓

Convex Hull Relaxation

- **Convex hull relaxation**
  - Offers a tight relaxation.



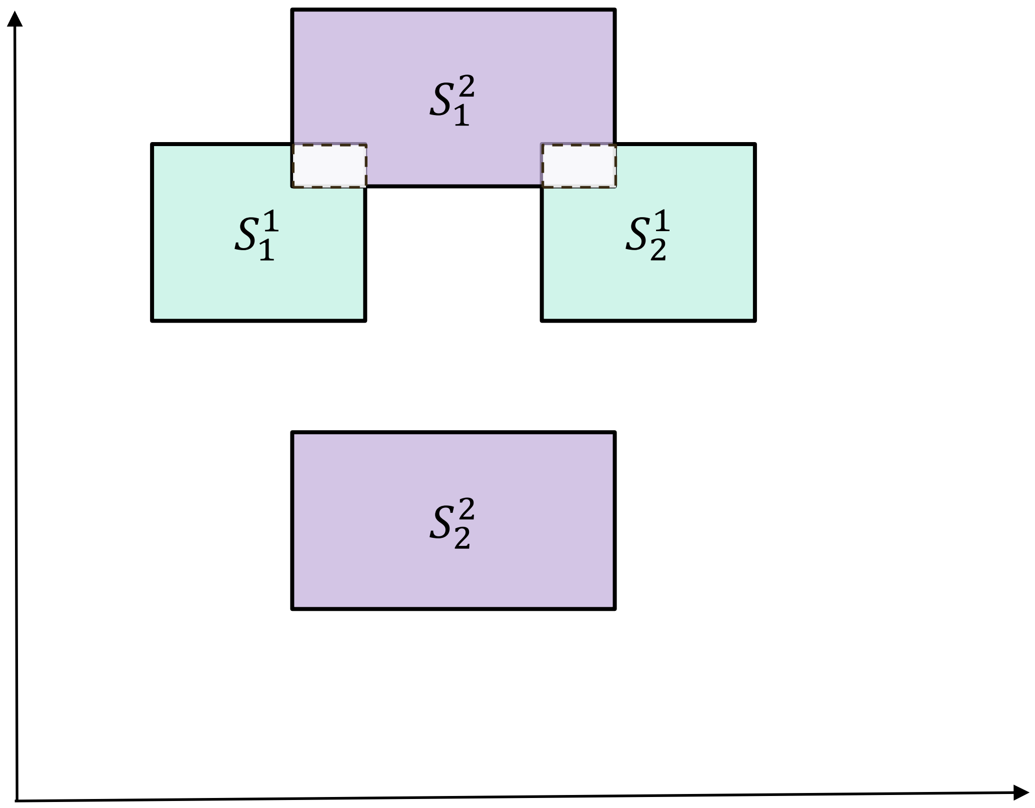
Disjunctive Form : $S_1 \vee S_2$

Convex Hull$(S_1 \cup S_2)$:

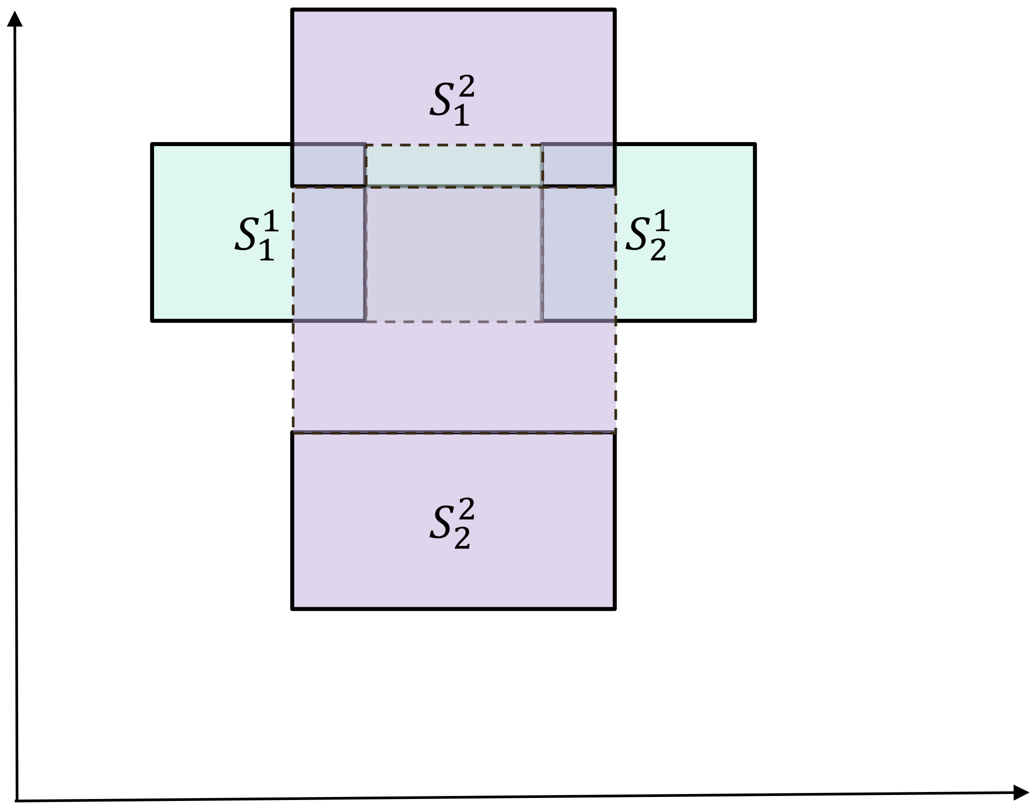
$$S = \{x \mid x = \lambda_1 x_1 + \lambda_2 x_2,$$
$$x_1 \in S_1, x_2 \in S_2, \lambda_1 + \lambda_2 = 1, \lambda_1, \lambda_2 \geq 0$$

Balas 1979
Raman and Grossmann 1994

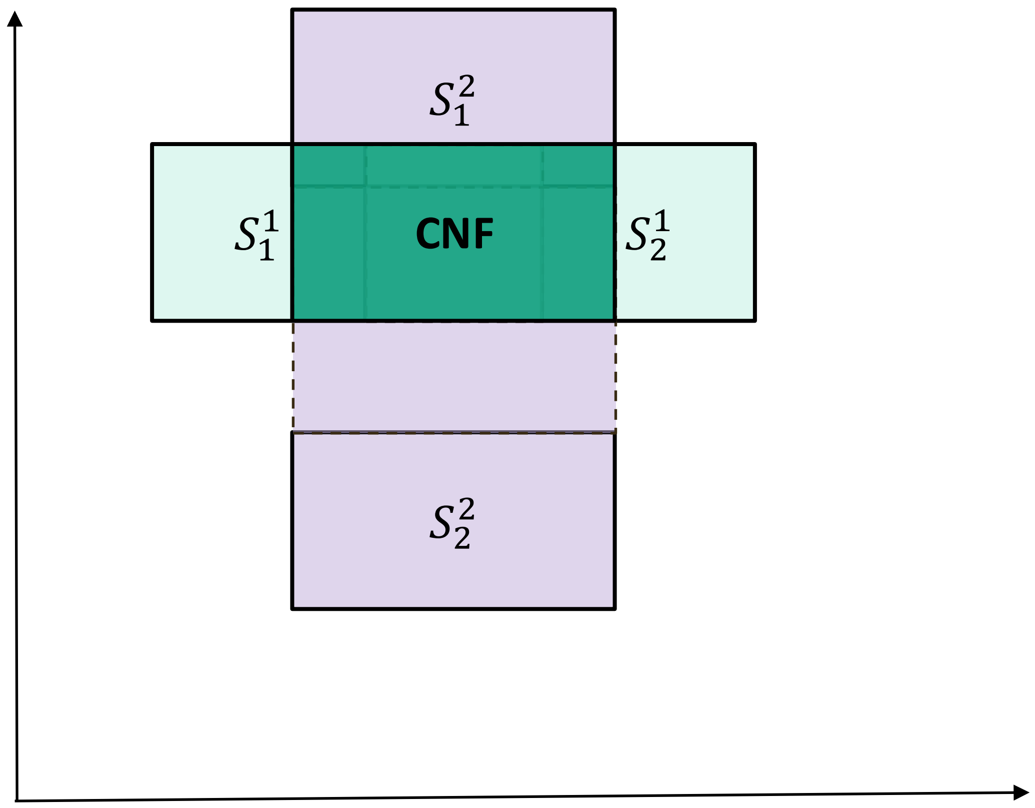
Multiple Rules



Conjunctive Normal Form (CNF)

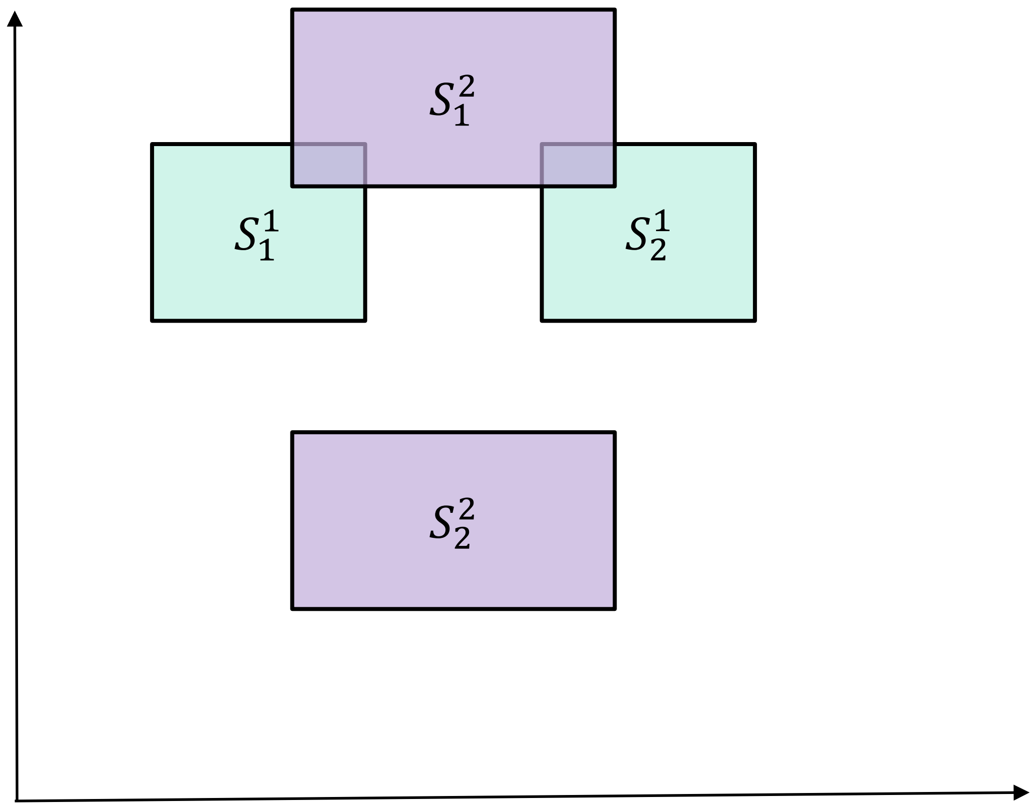


Conjunctive Normal Form (CNF)

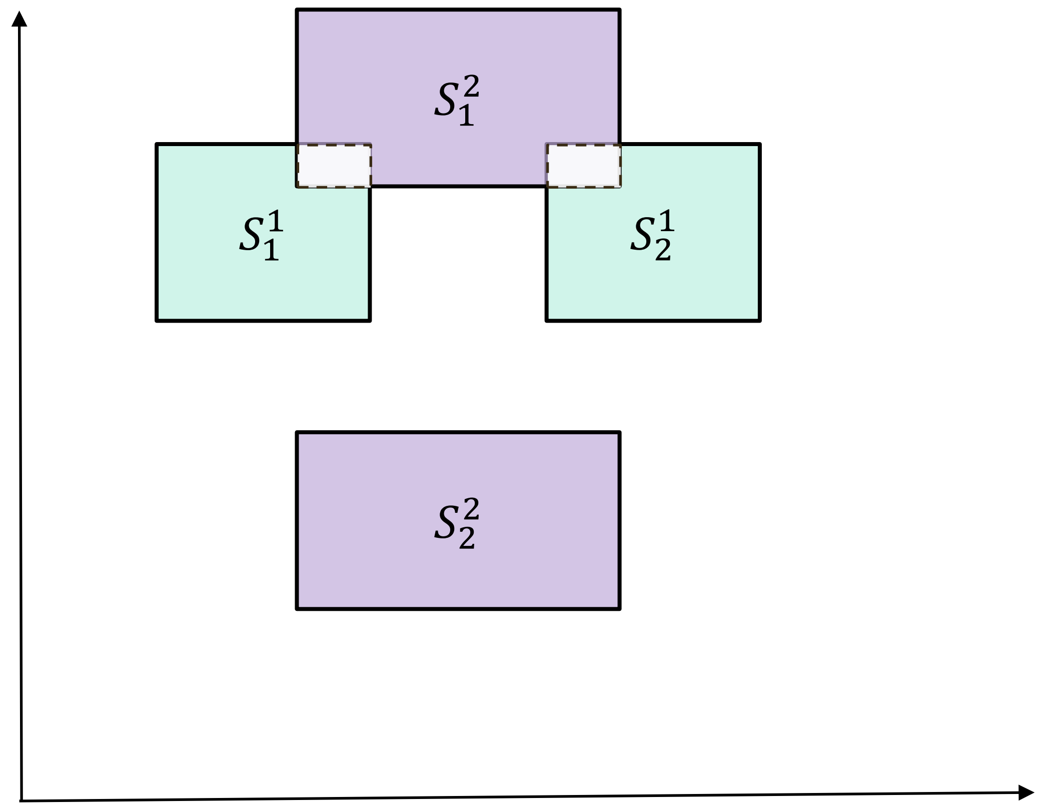


$$\text{conv}(\text{CNF}) = \bigcap_r \text{conv}(S_1^r \cup S_2^r)$$

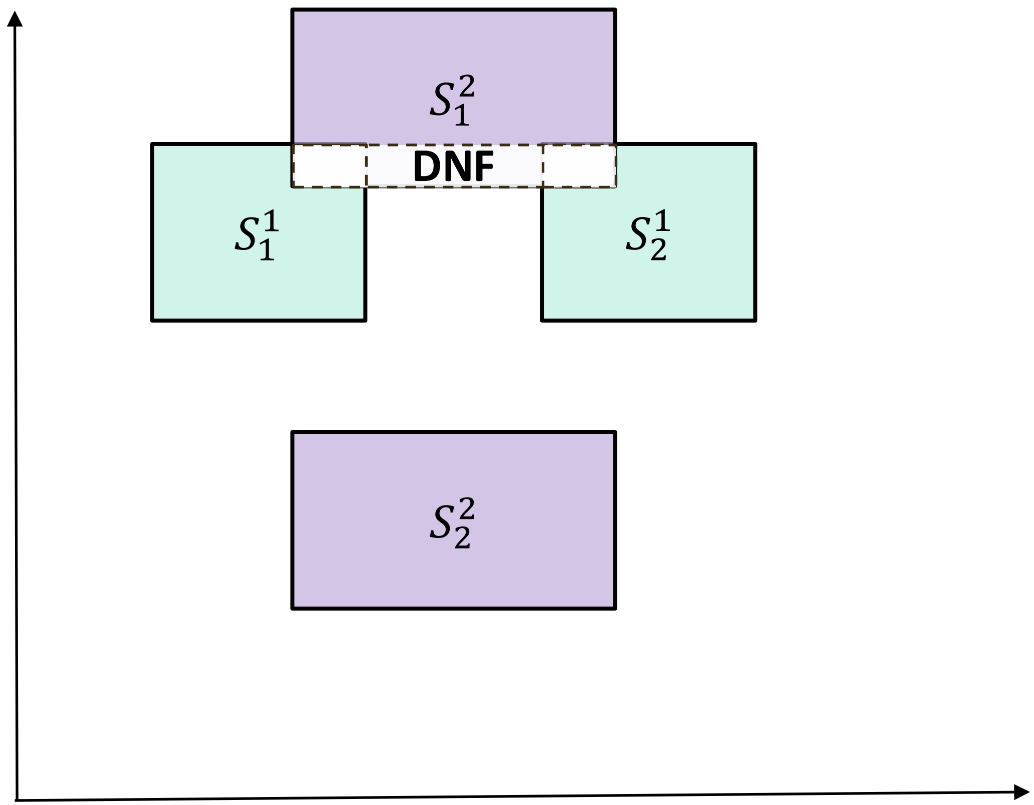
Multiple Rules



Disjunctive Normal Form (DNF)



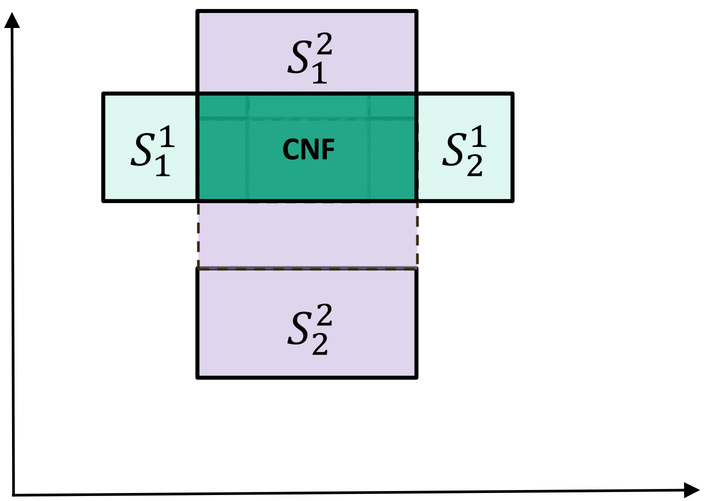
Disjunctive Normal Form (DNF)

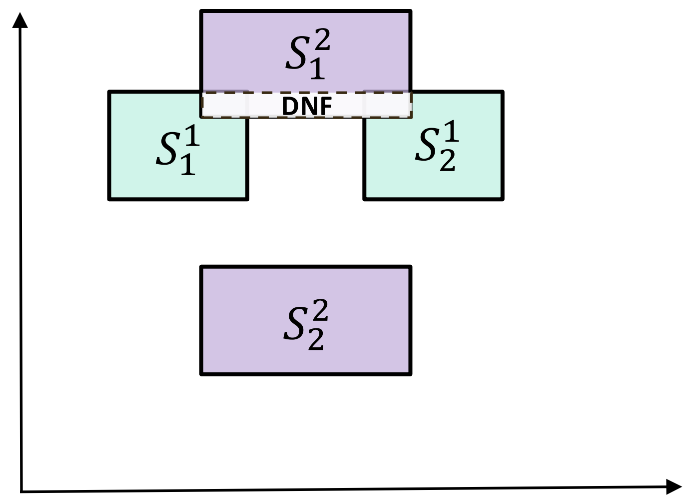


$$\text{conv}(\text{DNF}) = \text{conv}\Big(\bigcup_{k\in\{1,2\}^R} \bigcap_{r=1} S_{k_r}^r\Big)$$

CNF vs DNF

Conjunctive Normal Form	Disjunctive Normal Form
• Weaker relaxation • Variables grow linearly with no of rules. • Faster to solve.	• Tighter relaxation • Variables grow exponentially with no of rules. • Slower to solve





Both forms are linear and differentiable

Balas 1979
Raman and Grossmann 1994

Exact Rule Satisfaction

- The projection problem can be formulated as a differentiable linear program, implemented using DiffOpt.jl

$$
\begin{aligned}
(y^\star(x,\hat{y}), \eta^\star(x,\hat{y})) \; \in \; & \arg\min_{(y,\eta)} \; \mathbf{1}^\top \eta \\
\text{s.t.} \quad & (y,\eta) \in \widehat{\mathcal{F}}(x;\hat{y}),
\end{aligned}
$$

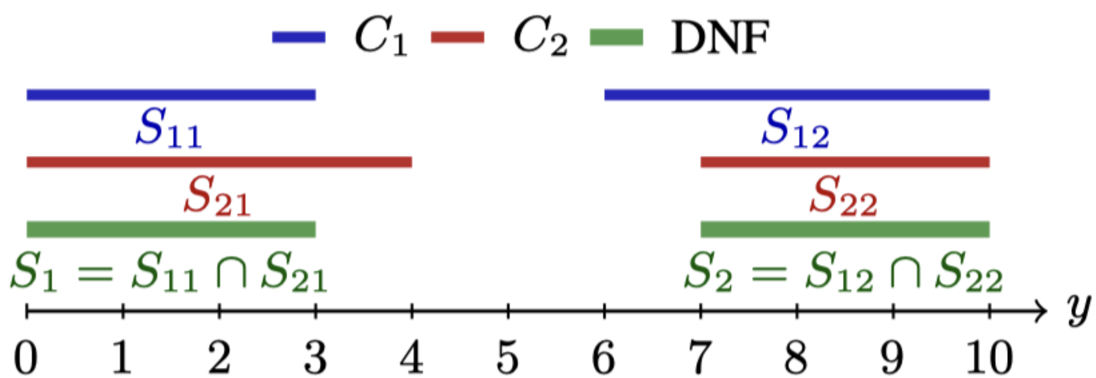
$$
\widetilde{\mathcal{F}}_{\mathrm{DNF}}(x;\hat{y}) = \mathrm{conv}\left(\bigcup_{k \in \Pi(x)} \; \bigcap_{r \in \mathcal{R}(x)} \widehat{\mathcal{S}}_{r,k_r}(x;\hat{y}) \right)
$$

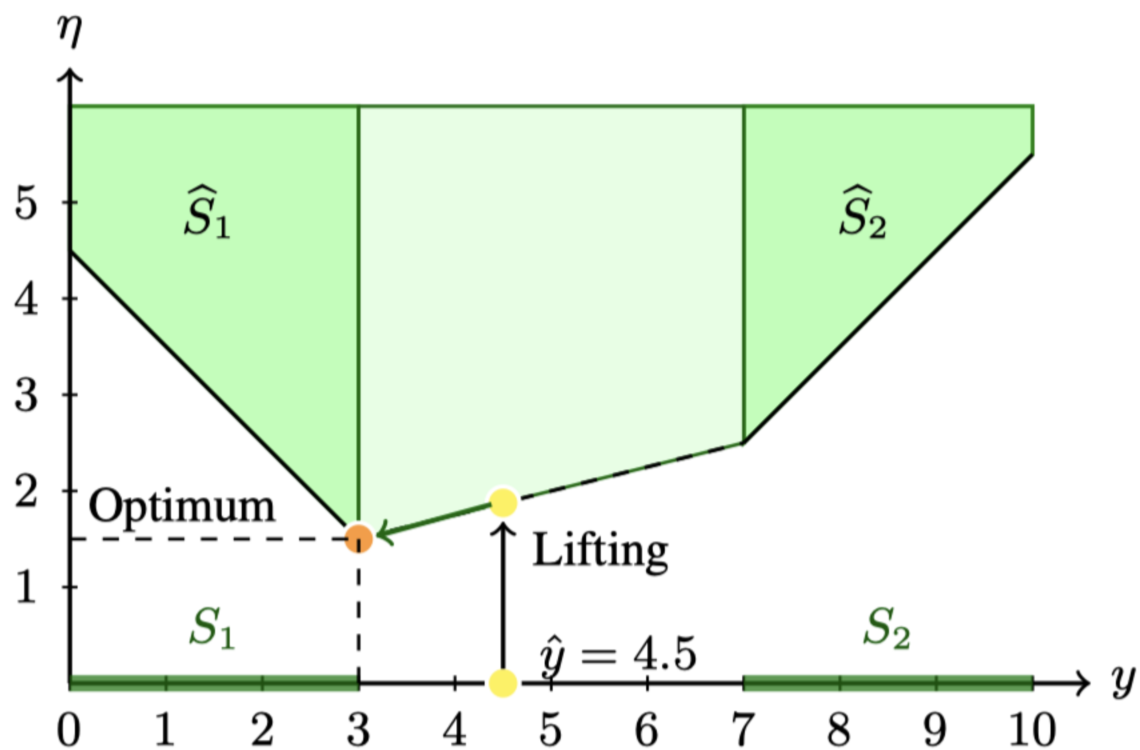
Besançon, M., Dias Garcia, J., Legat, B., & Sharma, A. (2024) DiffOpt.jl
Pirnay, H., López-Negrete, R., & Biegler, L. T. (2012) sIPOPT

- **Theorem (informal):** When solving projection problem with DNF using the extended LP formulation, the extreme point solution to the LP satisfies the constraints of the original formulation.

Balas 1979
Raman and Grossmann 1994

Geometric Intuition





Modeling using JuMP

- Rules are modeled as MILP or disjunctive constraints.

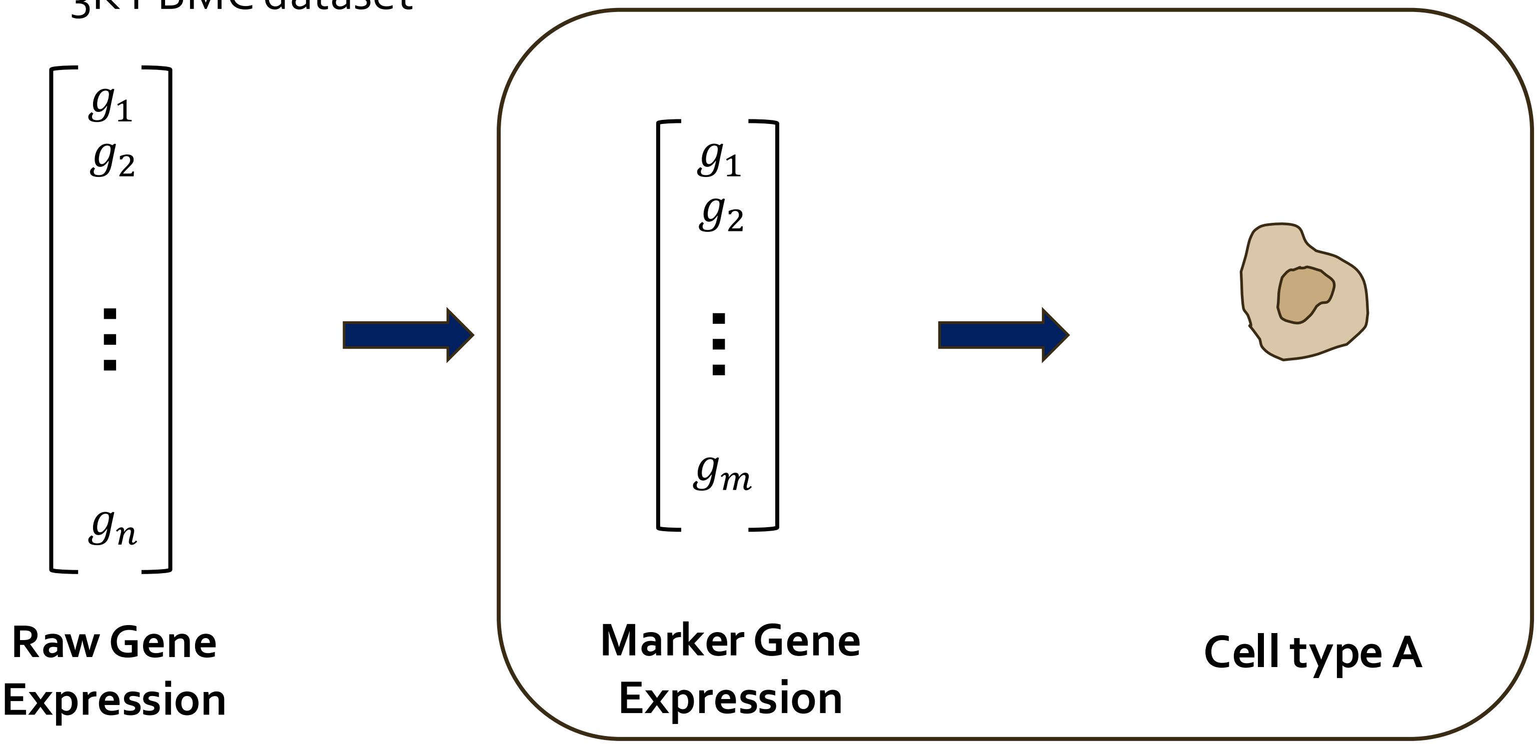
$$R_1: g_A \geq 1.3 \Rightarrow y_1 \geq 0.7 \vee y_2 \geq 0.7 \qquad R_2: g_B \geq 0.8 \Rightarrow y_2 \geq 0.6 \vee y_3 \geq 0.6$$

- Converted into modeling constraints in JuMP.
- Embed into neural network models in Flux.jl using DiffOpt.jl.
- Input-dependent rules are easy to handle: loop over the active rules and add only the relevant constraints.

```
@variable(model, y[1:3] >= 0)
# Rule R1
if g_A >= 1.3
    @variable(model, z1, Bin)
    @constraint(model, y[1] >= 0.7 * z1)
    @constraint(model, y[2] >= 0.7 * (1 - z1))
end
# Rule R2
if g_B >= 0.8
    @variable(model, z2, Bin)
    @constraint(model, y[2] >= 0.6 * z2)
    @constraint(model, y[3] >= 0.6 * (1 - z2))
end
```

Case Study: Overview

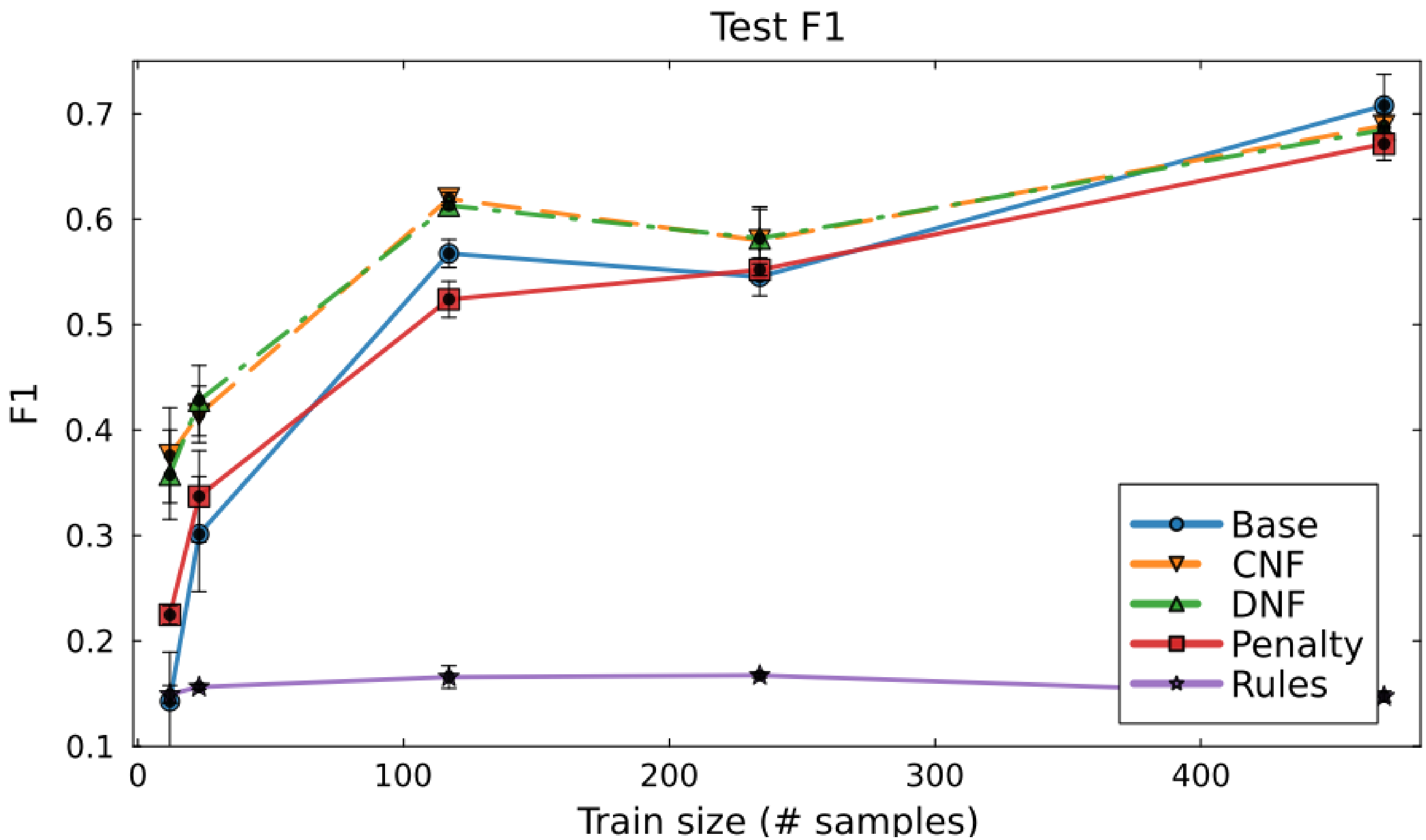
- **Single-cell RNA sequence classification problem (scRNA-seq)**
  - RNA-seq of cells
  - Gene counts of *marker genes*
  - 3K PBMC dataset



$$G_r(u) \geq \tau_r \;\Rightarrow\; \bigvee_{c \in \mathcal{C}_r} y_c \geq \rho_r,$$

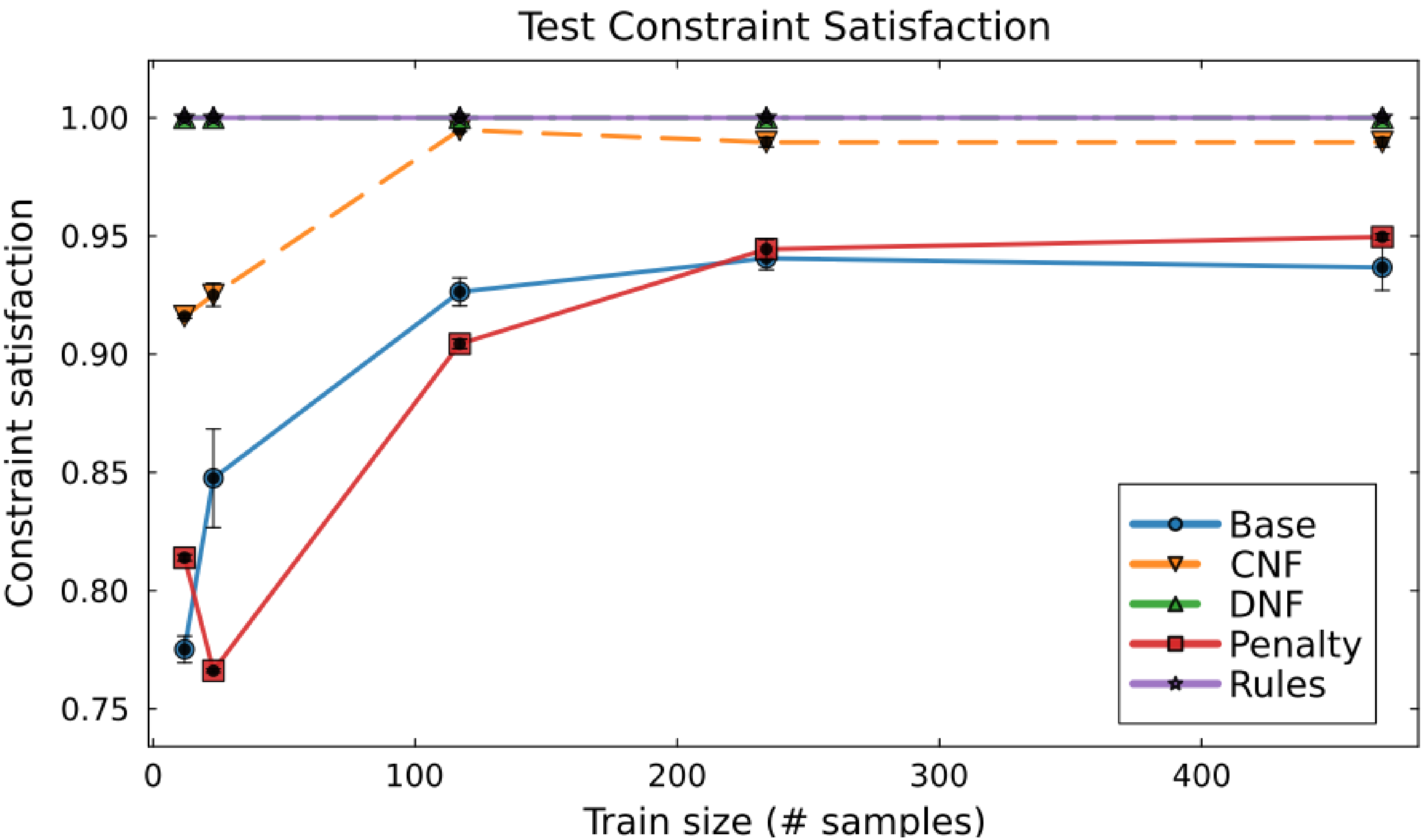
Test Performance

- Enforcing rules via projection improves performance in low-data regimes.
- As the dataset size increases, the unconstrained model also improves.



Constraint Satisfaction

- Tighter DNF-based projections can trade a small amount of F1 for stronger rule adherence.
- The unconstrained model can gradually learn the rules as the dataset size increases.



References and Acknowledgement

- Pal, Shraman and Can Li (2026). *DisjunctiveNet: Neural Symbolic Learning via Differentiable Convexified Optimization Layers.* Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)
- Code: https://github.com/li-group/DisjunctiveNet.jl

Paper

Code

Shraman Pal

Funding source