GENEOs as tools to build transparent AI systems for drug discovery and computer vision

Giovanni Bocchi [1] **Diogo Lavado** [1, 2] **Alessandra Micheletti** [1]

[1] Dept. of Environmental Science and Policy, Universitá degli Studi di Milano.

[2] NOVA School of Science and Technology, NOVA University Lisbon.

ExTRAInt, Lake Como, September 8 - 11, 2025

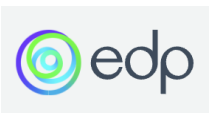
Outline

Introduction

This research originated from a collaboration between A.Micheletti and Patrizio Frosini.
Two companies were involved in the two main applications that will be illustrated today:

- Drug discovery - Giovanni Bocchi (postdoc at UniMI): Dompé Farmaceutici (Italy)

- Computer Vision - Diogo Lavado (PhD student at both UniMI and NOVA, coadvisor: Claudia Soares): EDP CNET Center for new energies technologies (Portugal)



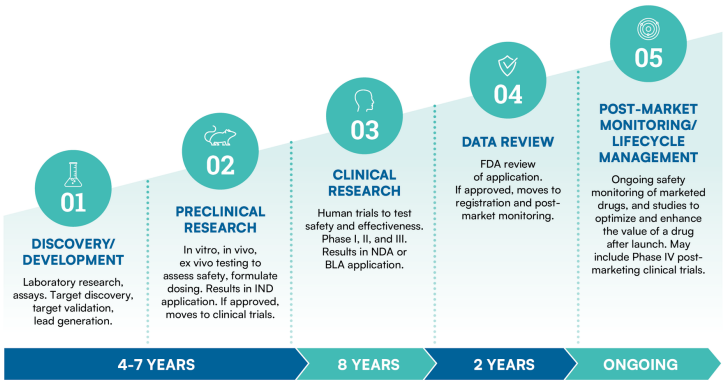
Introduction

We all collaborate with the Research Centre DESIRE - Decision Science Research Center, established in 2024 at Universitá degli Studi Milano (`https://desire.unimi.it/`)

This research is aimed to show the potential of GENEOs in building transparent, trustworthy and efficient ML models for different applications.

Drug discovery

Drug discovery (i.e., the process of developing drugs for new diseases) used to be a highly **time-consuming** (> 10 years) process.



AI for drug discovery

Nowadays, the number of applications of AI systems to many computational tasks relevant for the Drug Discovery process has literally **exploded**.

AI systems for Drug Discovery

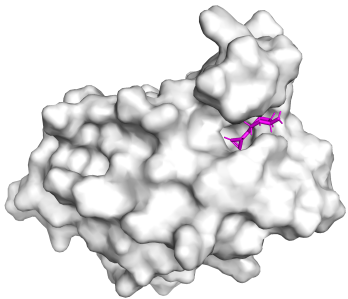
- *Protein Folding:* AlphaFold.
- *Protein Pocket Detection:* P2Rank, DeepSite, DeepPocket, CAVIAR, AF2BIND **GENEOnet**.
- *Protein Pocket Comparison:* Site2Vec, DeeplyTough, **GENEOmap (WIP)**.
- *Molecular Docking:* EQUIBIND, TankBind.

Protein pocket detection

We will focus on a specific task related to Drug Discovery: **Protein pocket detection**. Indeed, proteins are one of the most important biological targets for drugs.

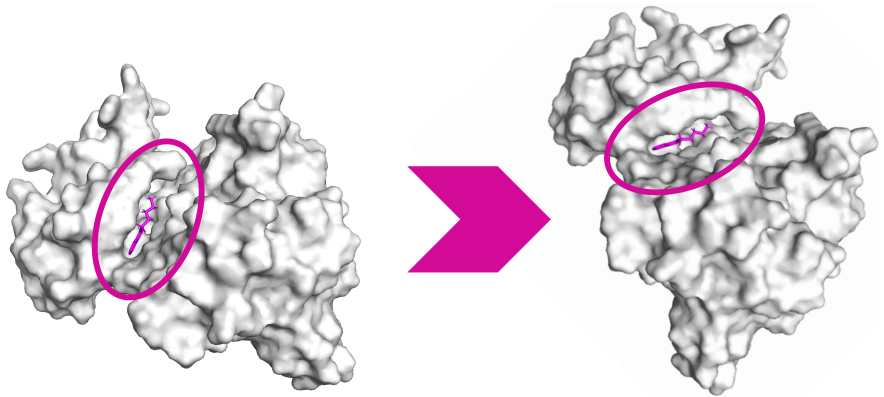
Protein data are usually stored in a **PDB** file that stores the atomic coordinates of the protein and any smaller molecules bound to it.

```
ATOM [] N [] 1.504 29.785 32.644 []
ATOM [] C [] 1.015 29.651 34.047 []
ATOM [] C [] 2.166 29.567 35.050 []
```



Equivariance

The problem of detecting pockets on the surface of proteins clearly benefits from an **equivariance** property with respect to 3D rigid motions:



Let's use GENEOs

You've seen that equivariance is one of the key properties of GENEOs; thus, we tried to build a model to predict protein pockets based on GENEOs.

Definition (GENEO)

A Group Equivariant Non-Expansive Operator F is a map between Φ and Ψ that, for a fixed homomorphism of groups T, has these two properties:

- **Equivariance:** $F(\varphi \circ g) = F(\varphi) \circ T(g)$ for all $\varphi \in \Phi$ and for all $g \in G$.
- **Non-Expansivity:** $||F(\varphi_1) - F(\varphi_2)||_\infty \leq ||\varphi_1 - \varphi_2||_\infty$ for all $\varphi_1, \varphi_2 \in \Phi$.

GENEOnet

How to define a pocket ?

The concept of a protein pocket is not universally well-defined; surely, a pocket is identified by a combination of geometrical and chemical properties.

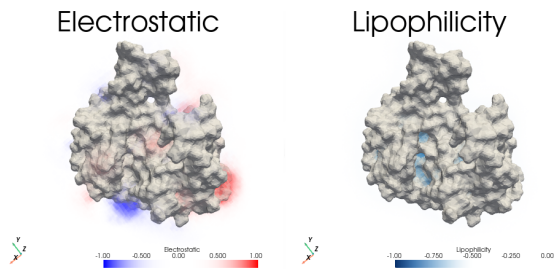
- **Geometry:** a more or less pronounced indentation in the surface of a certain volume.
- **Chemistry:** a "good" mix of chemical properties: electrical charges, hydrophilic properties, etc.

GENEOnet

The right input functions

Thus, we selected a small number of functions φ that should encode the geometrical and chemical properties of a protein.

- Distance
- Gravitational
- Electrostatic
- Lipophilic
- Hydrophilic
- Polar
- HB acceptor
- HB donor



GENEOnet

The right input functions

For example the electrostatic potentials $\varphi_3(x)$ is formally defined as

$$\varphi_3(x) = \sum_{a:\ d(x_a,x)\leq D} \frac{q_a}{d(x_a,x)}$$

where x_a, q_a are the atomic coordinate and the partial charge of atom a. Similarly, the lipophilic potential $\varphi_4(x)$ is defined as

$$\varphi_4(x) = \sum_{a:\ d(x_a,x)\leq D} \frac{l_a}{d(x_a,x)}$$

GENEOnet

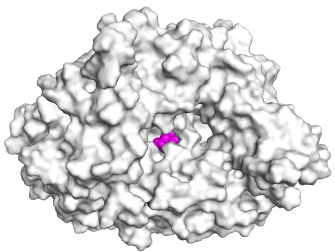
The ground truth

As already noticed, the definition of pocket is rather fuzzy, but to implement a machine learning method, it is required to have a clear definition of **ground truth**.

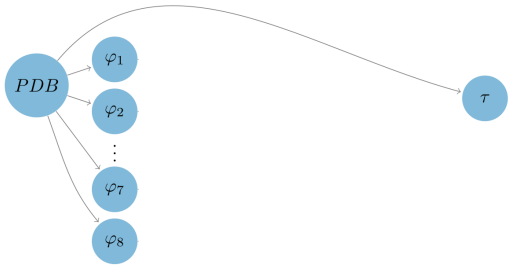
As ground truth, we consider the function
τ

$$\tau(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{otherwise} \end{cases}$$

where L it's a volume containing the ligand slightly enlarged.

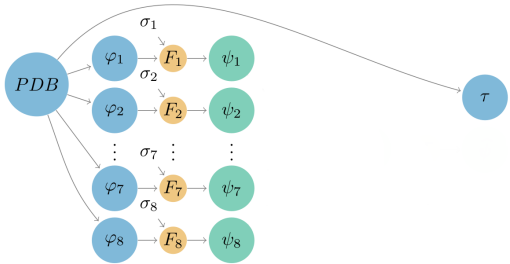


GENEOnet
The right inputs



Given an input PDB file, the model can compute the input functions φ_i and additionally the ground truth τ.

GENEOnet
The right GENEOs



Each input function is then processed by a family of parametric GENEOs F_i, which are convolutional operators.

$$F_i(\varphi_i) = \varphi_i * K_i$$

The kernels K_i are radial functions depending on the "shape" parameters σ_i:

$$K_i(x) = K_i(||x||, \sigma_i)$$

GENEOnet

The right GENEOs

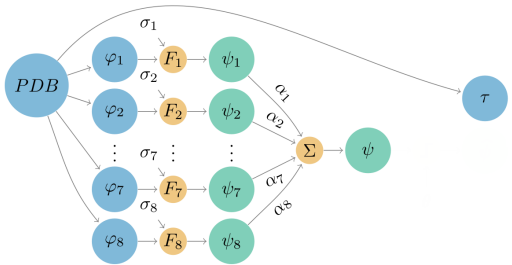
```
class GENEOUnit()
    def __init__(self, ...)
        self.sigma = sigma0
        self.act = act
    def kernel(self, ...)
        ...
        return k
    def call(self, x)
        k = self.kernel(...)
        s = conv3d(x, k)
        return self.act(s)
```

Each GENEO looks for a favorable configuration of the corresponding potential:

- F_3 looks for areas with a variation in the electrostatic potential. (Gaussian Laplacian Kernel)
- F_4 looks for areas uniformly lipophilic. (Gaussian Kernel)

GENEOnet

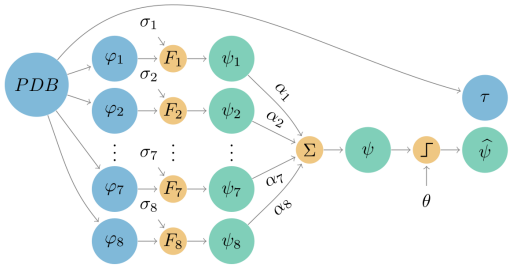
The right combination



Each first-level GENEO is subsequently aggregated into a second-level GENEO obtained via convex combination with weights α_j summing to 1:

$$F_\alpha(\cdot) = \sum_{j=1}^{d} \alpha_j F_j(\cdot)$$

GENEOnet
Prediction

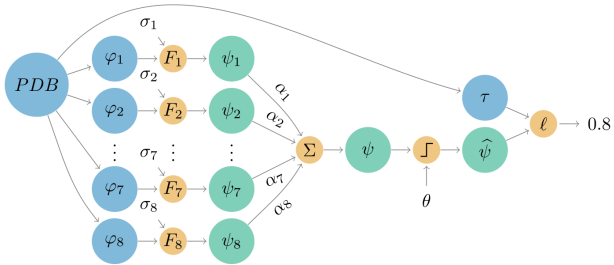


The output function $\psi(x)$ is normalized in [0,1] and it encodes the probability that each point x belongs to a pocket.

The final prediction is the binary function $\widehat{\psi}$ obtained as

$$\widehat{\psi}(x) = \begin{cases} 1 & \text{if } \psi(x) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

GENEOnet
Evaluation



Finally the prediction $\widehat{\psi}(x)$ and the ground truth $\tau(x)$ can be compared using the loss function $\mathcal{L}$ that measures the volumetric (in)coherence of the two functions.

$$\mathcal{L}(\widehat{\psi}, \tau) = 1 - \frac{|\widehat{\psi} \wedge \tau| + \kappa|(1 - \widehat{\psi}) \wedge (1 - \tau|)}{|\tau| + \kappa|1 - \tau|}$$

GENEOnet
Training

The loss function is optimized in Python using PyTorch and the Adam optimizer. From our experiments, we noticed that a training set made of 200 protein-ligand complexes was enough to estimate the values of the 17 learnable parameters of GENEOnet (σ_i, α_j, θ).

Shape parameters update

$$\sigma_i^{t+1} = \sigma_i^t - \eta u_i^t$$

Kernels are computed using updated parameters; equivalence is maintained during all the optimization.

Combination parameters update

$$\alpha_j^{t+1} = \alpha_j^t - \eta w_j^t$$

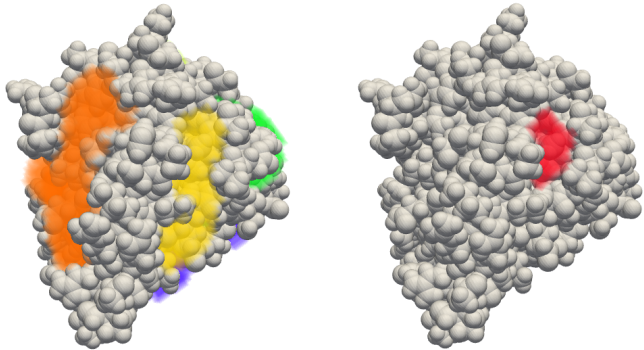
They are projected to keep the sum equal to one; convexity is maintained during all the optimization.

GENEOnet

Pockets and scoring

Given the prediction $\widehat{\psi}(x)$, predicted pockets can be segmented as connected components.

To order predicted pockets, a pocket **score** can be derived as a weighted mean of ψ restricted to the corresponding connected component.

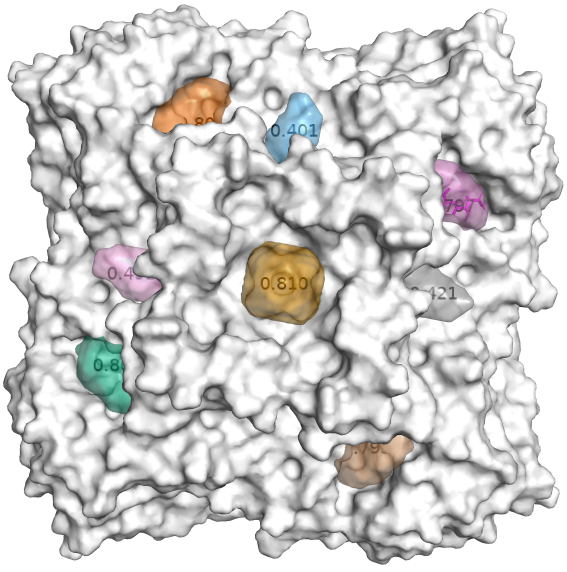


GENEOnet
Prediction example

Protein 2QWE

Such a protein is made by four identical units arranged in a symmetrical conformation. The true pocket is repeated **four times**!

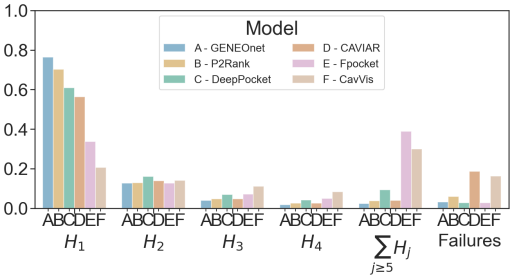
Equivariance and non-expansivity combined guarantee the result of four similar predicted pockets.

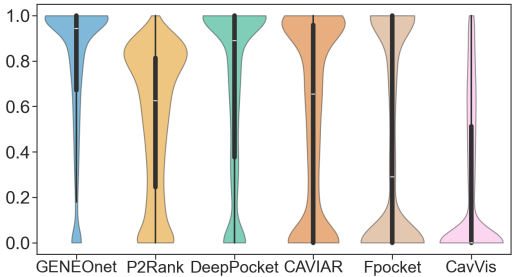


GENEOnet

Performances

Once trained, GENEOnet performances were tested using many different metrics:





GENEOnet

Transparency

Most importantly, GENEOnet has some features that make it an **explainable-by-design**, **transparent**, and **robust** AI method:

GENEOnet features

1. It allows to incorporate prior knowledge.
2. It has a few learnable parameters and requires less training data.
3. The computational costs are reduced.
4. Its parameters are easy to interpret.
5. It is robust to minor alterations of the input data.

GENEOnet

Parameters interpretation

Since GENEOnet has only 17 parameters, we can assign each of them a **clear** and **human-understandable** meaning:

1. The shape parameters σ_i influence the **amplitude** of each kernel, affecting the extension of the volume of interest.
2. The combination parameters α_j can be interpreted as **feature importance** values, denoting the input functions that are more important to derive the prediction.
3. The threshold θ regulates the extension and number of predicted pockets.

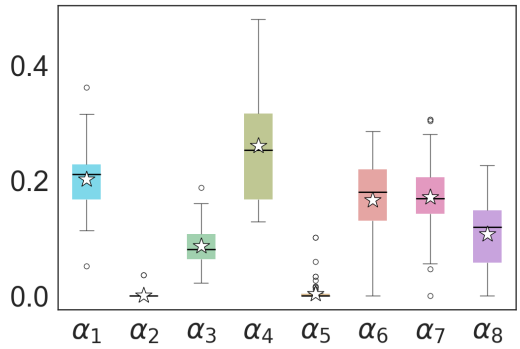
Channel	σ_i	α_j	θ
Distance	3.110	0.362	0.756
Gravitational	5.197	0.002	
Electrostatic	2.561	0.054	
Lipophilic	4.678	0.338	
Hydrophilic	3.545	0.001	
Polar	6.166	0.185	
HB Acceptor	4.186	0.056	
HB Donor	3.908	0.001	

GENEOnet

Parameters robustness

We also performed a **sensitivity analysis** of the parameters: 200 models were trained using different training sets of the same size (200).

This allows us to further evaluate the importance of the coefficients also from a statistical point of view.

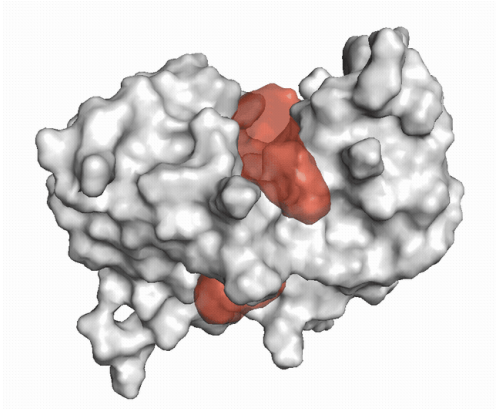


GENEOnet

Robustness

Non-expansivity provides a theoretical guarantee that GENEOnet should be robust to minor perturbations in the input data.

We tested this fact using molecular dynamics data (data regarding the evolution of a protein immersed in a solvent over time).



GENEOnet

Webservice: test it!

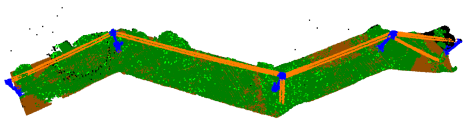
The chosen GENEOnet model was made available through a webservice at the URL https://geneonet.exscalate.eu/.

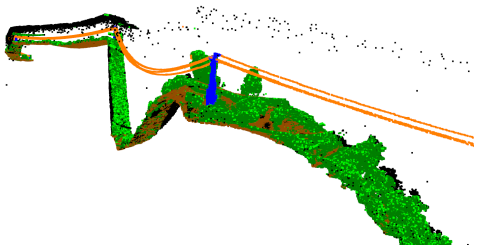
TEST IT!

GENEOnet references

Power Grid Inspections

Transmission system operators, such as EDP, have the mission to inspect the power grids in order to prevent defects, power outages, and even forest fires.





Our Goal: Detect Power Line Towers!

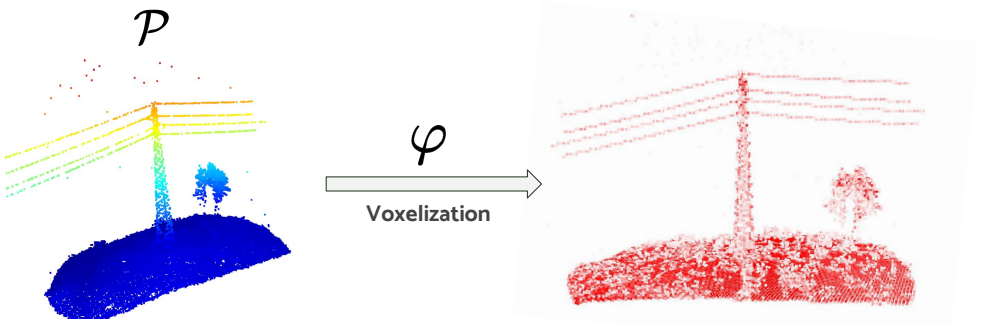
The Key to Automated Grid Inspection

Why Towers?

- **Towers are the backbone of the power grid.**
- Their detection is crucial for inspection, maintenance, and safety.
- Knowing tower positions allows us to infer the location of power lines and monitor the entire grid.

SceneNet: Architecture Overview

Step 1: Voxelization



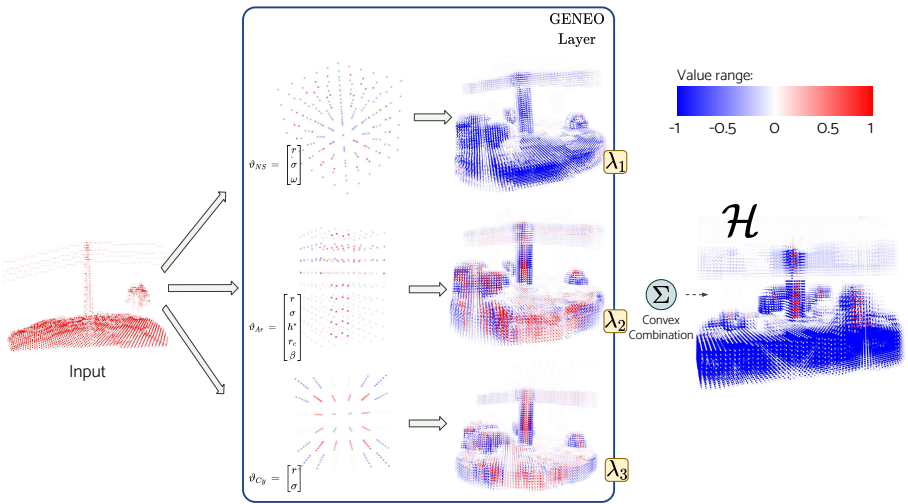
Point cloud format: Set of N 3D points

Voxel-grid format: 3D grid with 128^3 voxels

Voxels are colored according to their density values, red symbolizes voxels with high density, whereas those with density close to zero are white.

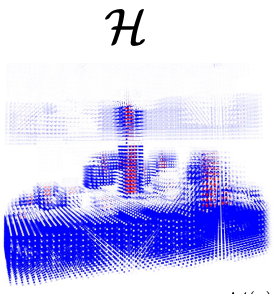
SceneNet: Architecture Overview

Step 2: GENEO Layer and Convex Combination



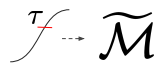
SceneNet: Architecture Overview

Step 3: Tower Probability Map



$\mathcal{H}$

$\mathcal{M}$



$$\mathcal{M}(x) = \left(\tanh\left(\mathcal{H}(x)\right)\right)_+,$$

with $(t)_+ = \max\{0, t\}$ (the ReLU)

$$\widetilde{\mathcal{M}}(x, \tau) = \left\{\mathcal{M}(x)\right\} \geq \tau$$

SceneNet: Optimization Setup

Step 4: GENEO Loss

$$\underset{\lambda,\vartheta}{\text{minimize}} \quad \underset{X,y,\alpha,\epsilon}{\mathbb{E}} \left\{ \mathcal{L}_{seg}(\lambda, \vartheta) \right\}$$
$$\text{s.t.} \quad \vartheta \geq 0, \quad \boxed{\lambda^T \mathbf{1} = 1,} \quad \lambda \geq 0$$

$$\lambda_n = 1 - \sum_{i=1}^{N-1} \lambda_i$$

$$\mathcal{L}_{seg}(\lambda, \vartheta) = \sum f_w(\alpha, \epsilon, y) \left(\underset{\lambda, \vartheta}{\mathcal{M}}(X) - y \right)^2$$

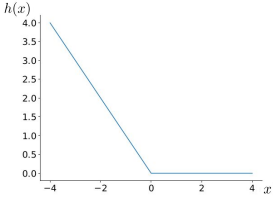
SceneNet: Optimization Setup

Step 5: GENEO Loss

$$h\colon \mathbb{R} \to \mathbb{R}_0^+$$
$$h(x) = \big(-x\big)_+$$

$$\underset{\lambda,\vartheta}{\text{minimize}} \quad \mathop{\mathbb{E}}_{X,y,\alpha,\epsilon}\left\{\mathcal{L}_{seg}(\lambda,\vartheta)\right\} + \rho_l\Big(\sum_i^K h(\lambda_i)\Big) + \rho_t\Big(\sum_i^K\sum_j^{T_i} h(\vartheta_{ij})\Big),$$

Hyper parameters $\rho_l,\ \rho_t > 0$



Evaluating SCENE-Net

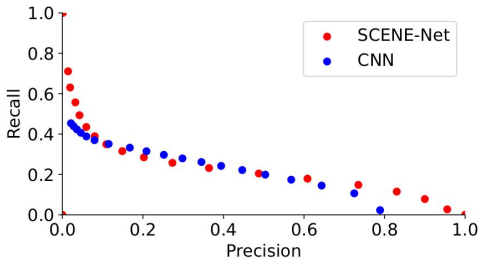
Baseline Comparison

We use a standard CNN as a baseline with a single convolutional layer, which shares the same overall pipeline as SCENE-Net but differs fundamentally in kernel definition and parameterization.

Property	**CNN Baseline**	**SCENE-Net**
Number of parameters	2,190	11
Layer depth	1 (single convolutional layer)	1 (GENEO-layer)
Number of kernels	3	3
Kernel size	$9 \times 5 \times 5$	$9 \times 5 \times 5$ (discretized from continuous functions)
Kernel definition	Unconstrained weights	Geometric priors (GENEOs)
Interpretability	$\times$	✓
Resolution Agnostic	$\times$	✓
Scalable design	$\times$	✓

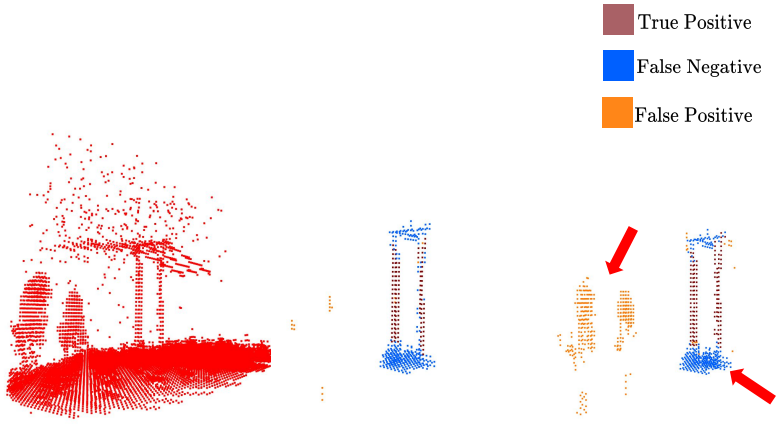
Evaluating SCENE-Net
Quantitative Results

Method	Precision	Recall	IoU
CNN	0.44 ($\pm$ 0.07)	**0.26** ($\pm$ 0.02)	0.53
SCENE-Net	**0.82** ($\pm$ 0.08)	0.13 ($\pm$ 0.05)	**0.58**



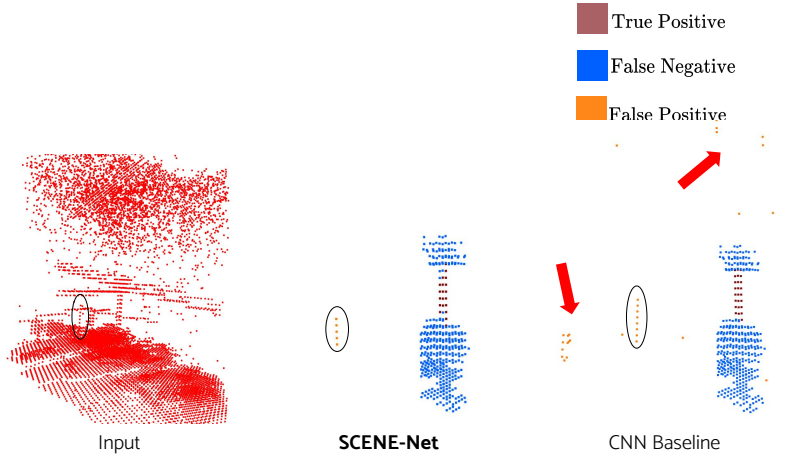
With changing thresholds τ

Evaluating SCENE-Net
Qualitative Results



Evaluating SCENE-Net

Evidences of Robustness



25

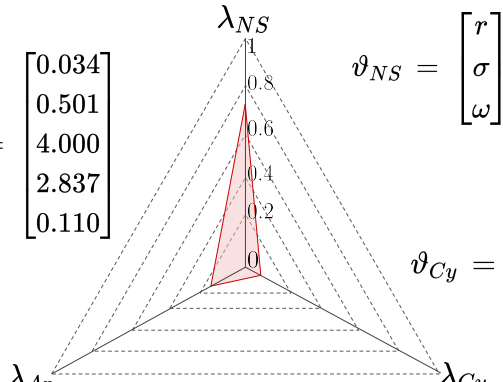
Interpretability of SCENE-Net

All Parameters are Meaningful

$$\vartheta_{Ar} = \begin{bmatrix} r \\ \sigma \\ h^* \\ r_c \\ \beta \end{bmatrix} = \begin{bmatrix} 0.034 \\ 0.501 \\ 4.000 \\ 2.837 \\ 0.110 \end{bmatrix}$$

$$\vartheta_{NS} = \begin{bmatrix} r \\ \sigma \\ \omega \end{bmatrix} = \begin{bmatrix} 3.842 \\ 0.262 \\ 0.013 \end{bmatrix}$$

$$\vartheta_{Cy} = \begin{bmatrix} r \\ \sigma \end{bmatrix} = \begin{bmatrix} 0.683 \\ 1.773 \end{bmatrix}$$

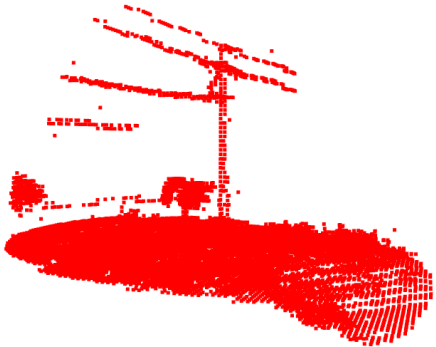


$$\lambda = \begin{bmatrix} \lambda^*_{NS} \\ \lambda_{Ar} \\ \lambda_{Cy} \end{bmatrix} = \begin{bmatrix} 0.7634 \\ 0.1616 \\ 0.0750 \end{bmatrix}$$

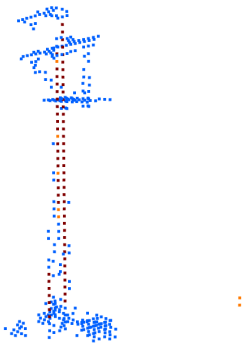
Resolution-agnostic inference of SCENE-Net

Generalization to higher voxel resolutions

The model trained at 64^3 with kernel size $(9, 5, 5)$ generalizes to 128^3 using kernel $(12, 5, 5)$ without retraining, retaining accurate tower localization.



Sample discretized in a 128^3 grid



SCENE-Net prediction

Limitations of SCENE-Net

Limitations

1. **Object specificity:** The current model is tailored for pole-like objects (e.g., power line towers) and may not generalize to other object types without significant adaptation.
2. **Voxelization dependency:** The reliance on voxelization introduces challenges for scaling to large-scale or high-resolution datasets, as memory and computational requirements grow rapidly.
3. **GENEO curation:** For each new application, a suitable set of GENEOs must be manually designed or curated, which limits automation and broad applicability.

Our Approaches
Future Directions

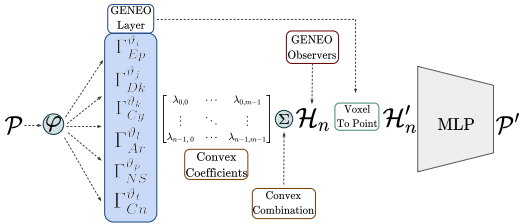


Figure: **SCENE-NetV2:** A grey-box model for computer vision. Geometric features extracted by GENEOs are provided as input to black-box models (e.g., MLPs), combining interpretability with flexibility.

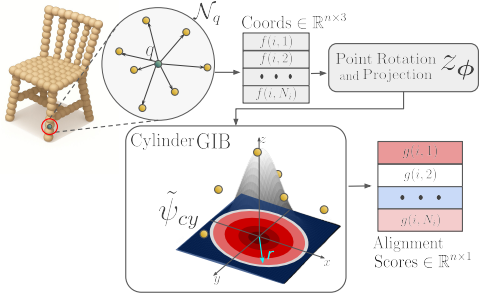


Figure: **GIBLy:**This architecture eliminates the need for voxelization by applying GENEOs directly to raw point clouds in continuous domains, enabling the first continuous parametric convolutions in 3D.

SCENE-Net references

Thank you for the attention!