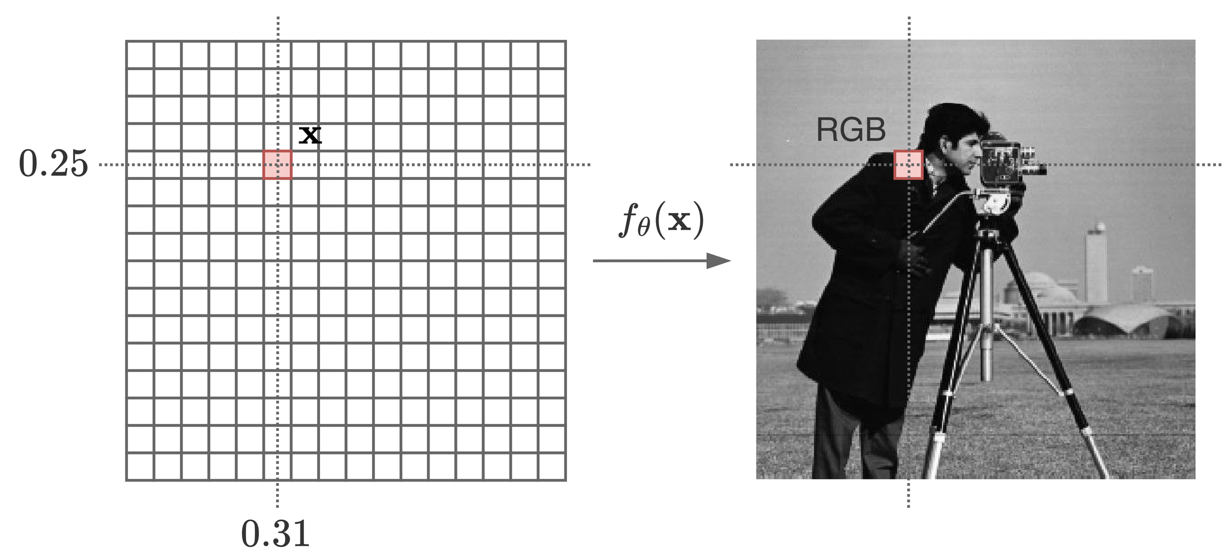
Generalised Implicit Neural Representations

D. Grattarola and P. Vandergheynst

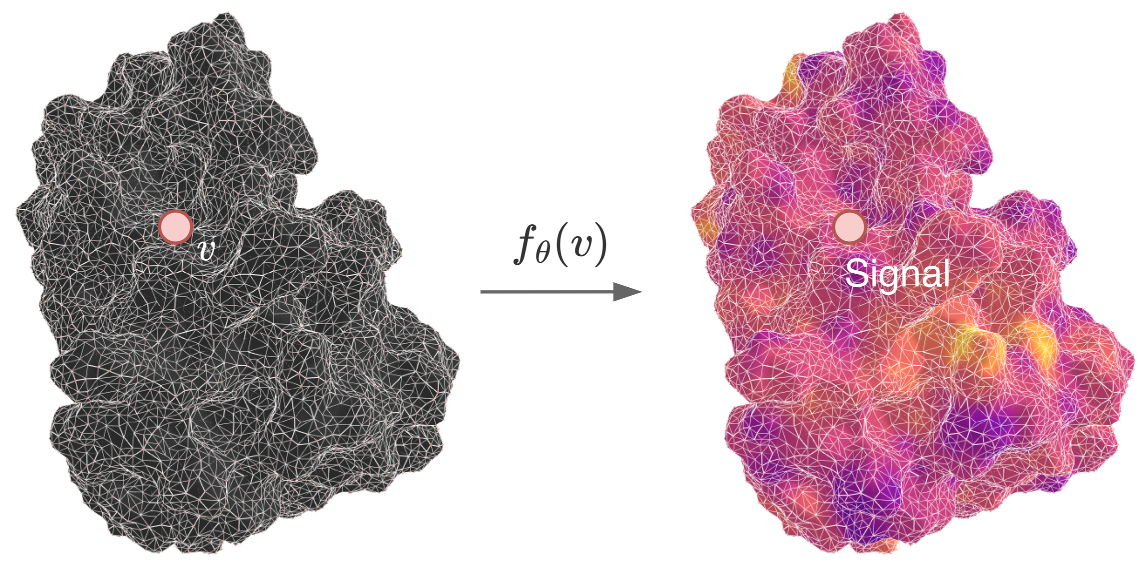
EPFL
NEURAL INFORMATION PROCESSING SYSTEMS

Implicit Neural Representations

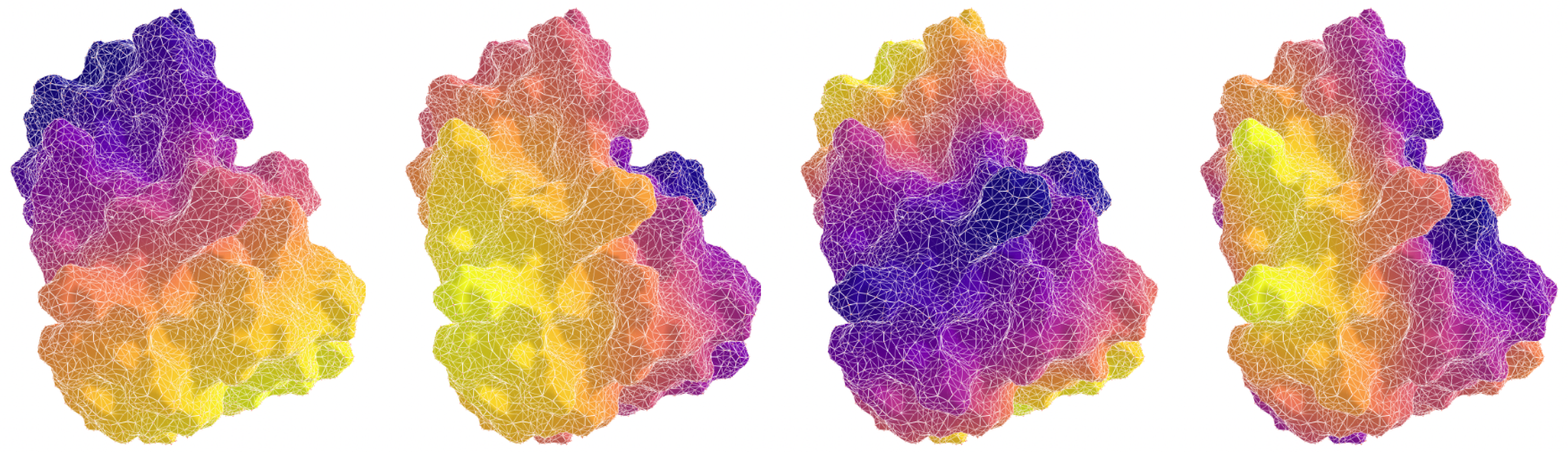


Signals are typically represented *explicitly* as a grid of observations.
An *implicit* representation for a signal is a map $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$ that maps the coordinates of points in the domain (*e.g.*, pixel coordinates) to the corresponding value of the signal (*e.g.*, RGB values).
An **implicit neural representation** (INR) approximates f with a neural network f_θ trained on observations $(\mathbf{x}_i, \mathbf{y}_i)$.

Generalised Implicit Neural Representations



In our paper, we study the setting where, instead of a signal on a grid, we observe a **signal on a graph** (a discretization of some space $\mathcal{T}$). In this case, an implicit representation is a map $f : \mathcal{T} \rightarrow \mathbb{R}^p$.

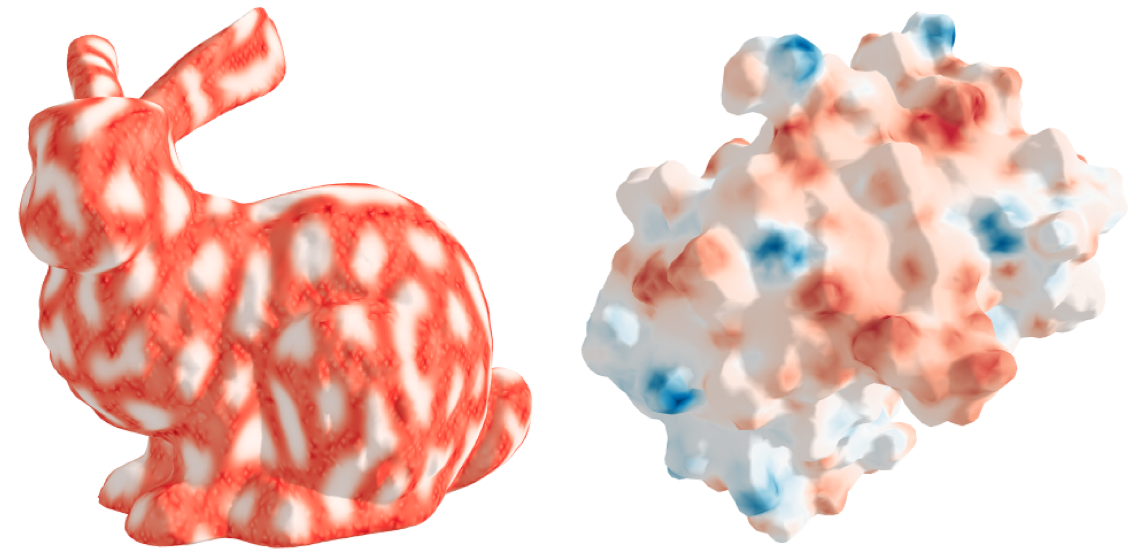


To train a **generalised INR**, we use node embeddings based on the first k eigenvectors of the graph Laplacian as a coordinate system:

$$\mathbf{e}_i = \sqrt{n}\ \underbrace{[\mathbf{u}_{1,i}, \dots, \mathbf{u}_{k,i}]^\top}_{\text{Laplacian eigenvectors}} \in \mathbb{R}^k$$

This is a well-defined coordinate system, since the embeddings converge to the continuous Laplace-Beltrami eigenfunctions of $\mathcal{T}$ for $n \rightarrow \infty$.

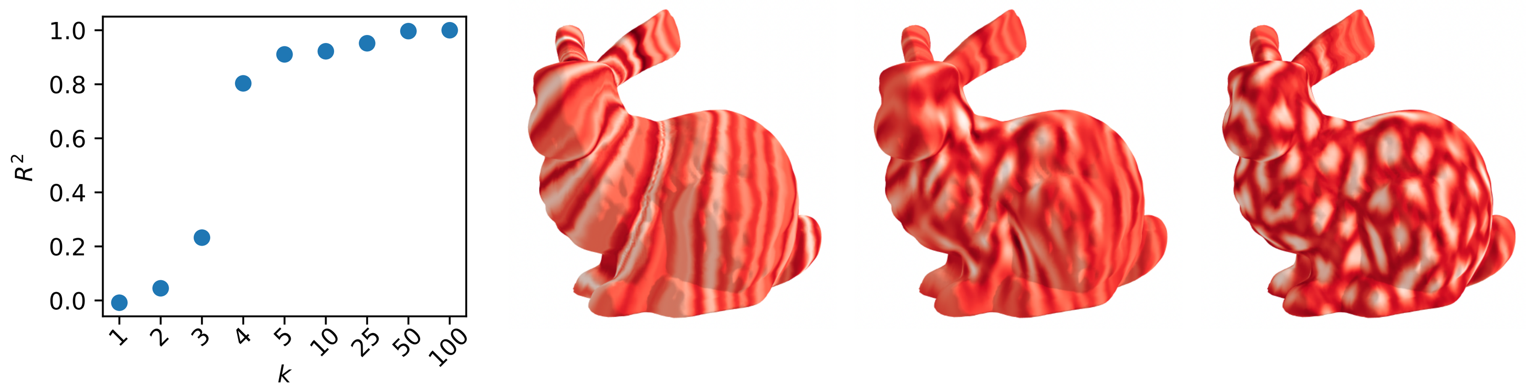
Learning Generalised INRs



	Bunny	Protein	US Election
R^2	1.000	1.000	0.999
MSE	$9.14 \cdot 10^{-8}$	$1.17 \cdot 10^{-10}$	$1.45 \cdot 10^{-3}$

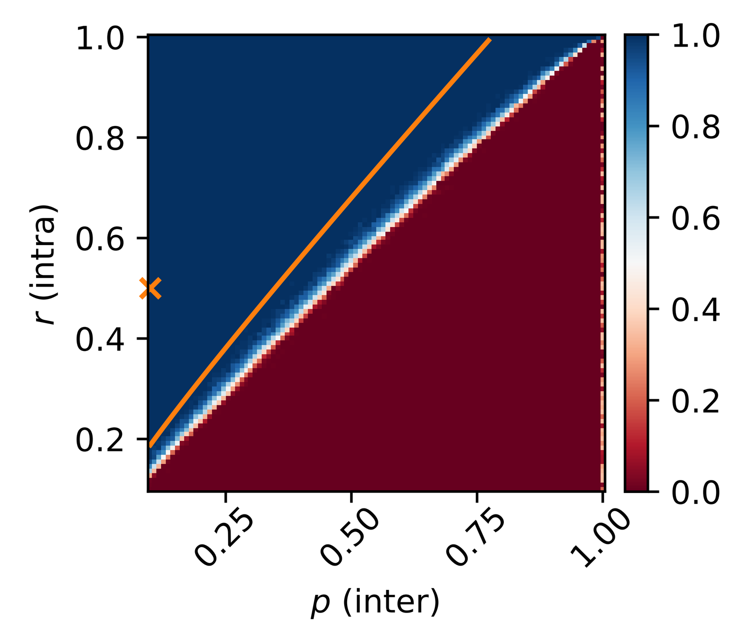
Approximation error

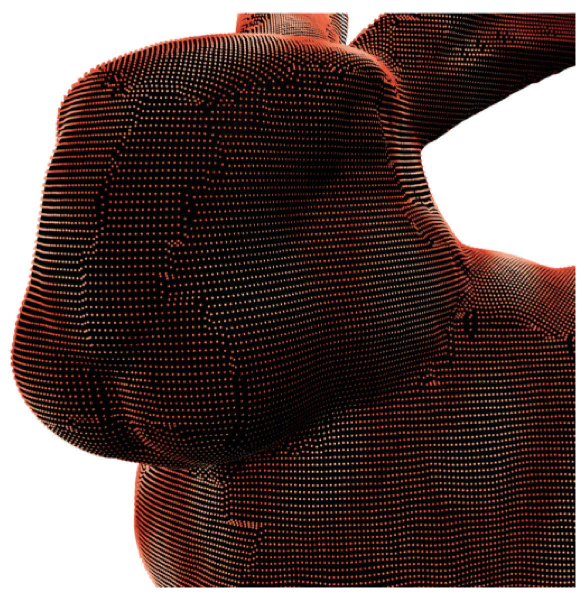
We begin by testing whether our method can indeed **learn some graph signals**: a reaction-diffusion pattern on the Stanford bunny, the electrostatic potential of a protein surface, and the political opinions in a social network.



We also study the effect of changing the **size of the embeddings**, showing that small values of k are enough to get a good reconstruction.

Transferability

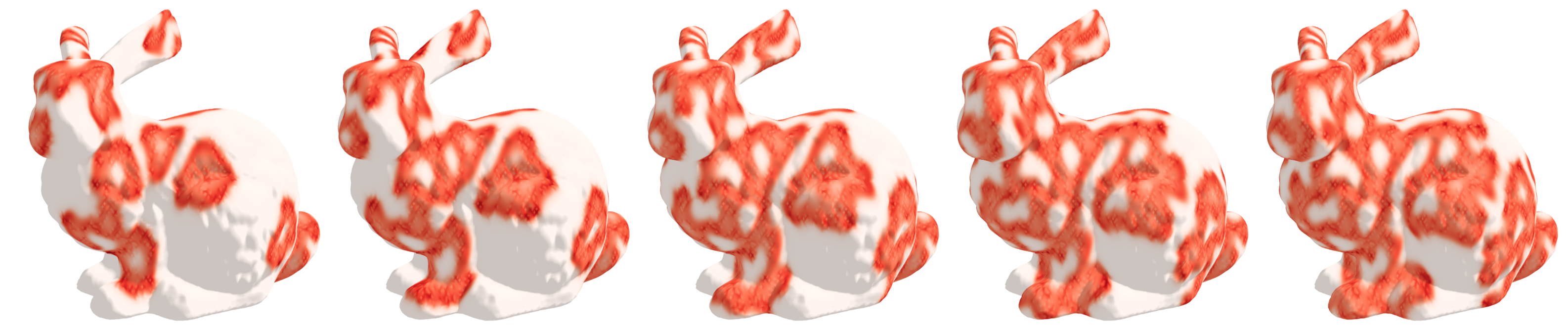




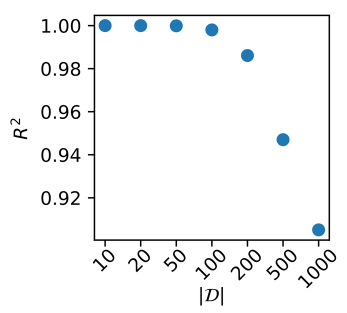
We study the transferability of GINRs by training them on one graph and then predicting the signal on **different realisations of the same domain**.

For this, we consider a random stochastic block model and a super-resolved version of the bunny mesh.

Conditional GINRs

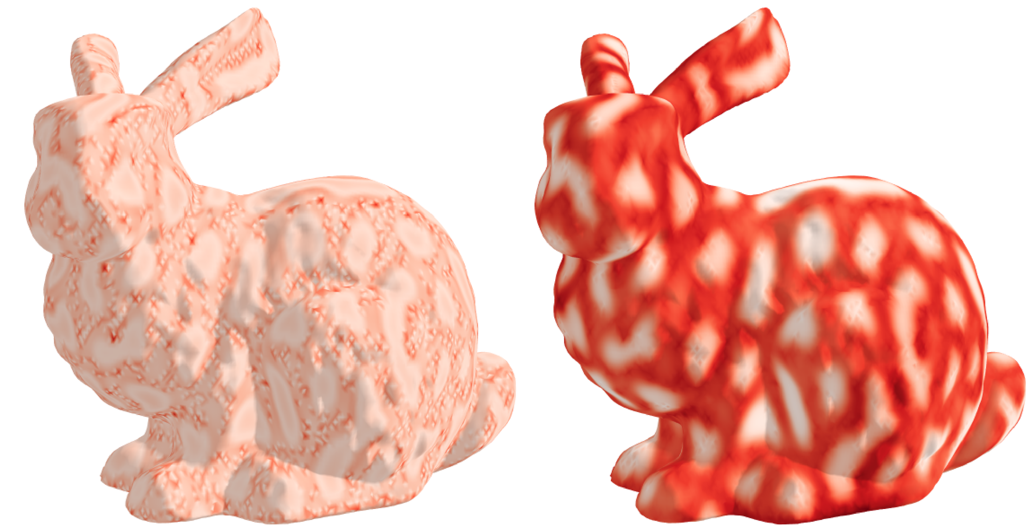


We can also **condition the output** of the GINR on an additional *global* input.
This allows us to represent spatio-temporal signals ($f_\theta(\mathbf{e}_i, t)$) or to store multiple signals on different domains ($f_\theta(\mathbf{e}_i, \mathbf{z}_d)$ for the dth signal), using a single neural network.



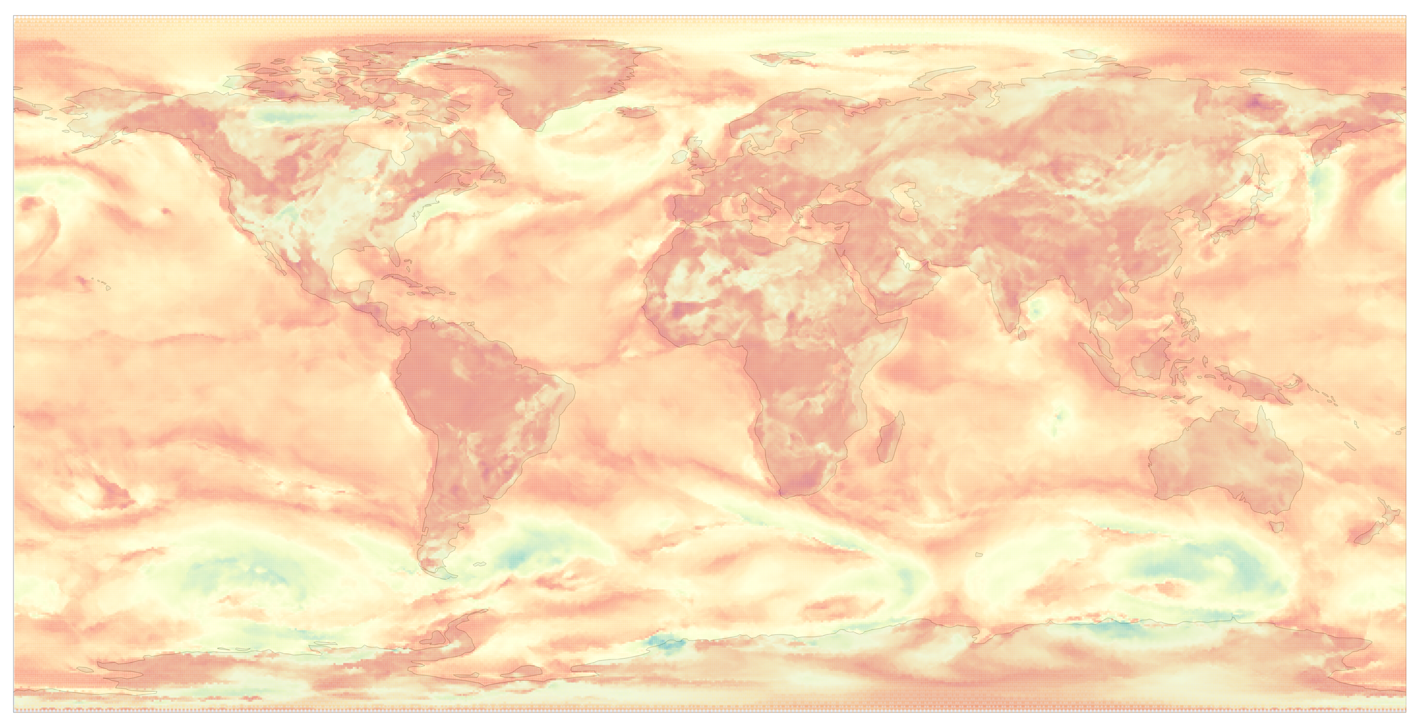
R^2 *vs.* n. of signals.

Solving differential equations



Similar to previous work on INRs, we can supervise our GINR using **derivatives of the target signal**. Here, we train the GINR using the Laplacian of the signal and recover the true f almost perfectly.

Weather modelling



As a final proof of concept, we apply all techniques studies so far to model and super-resolve a spatio-temporal signal on the surface of the Earth. We consider three different **meteorological signals** (wind, temperature, clouds) and use a GINR to represent them.

See the QR code in the top-right corner for a high-resolution video.