Math and Data

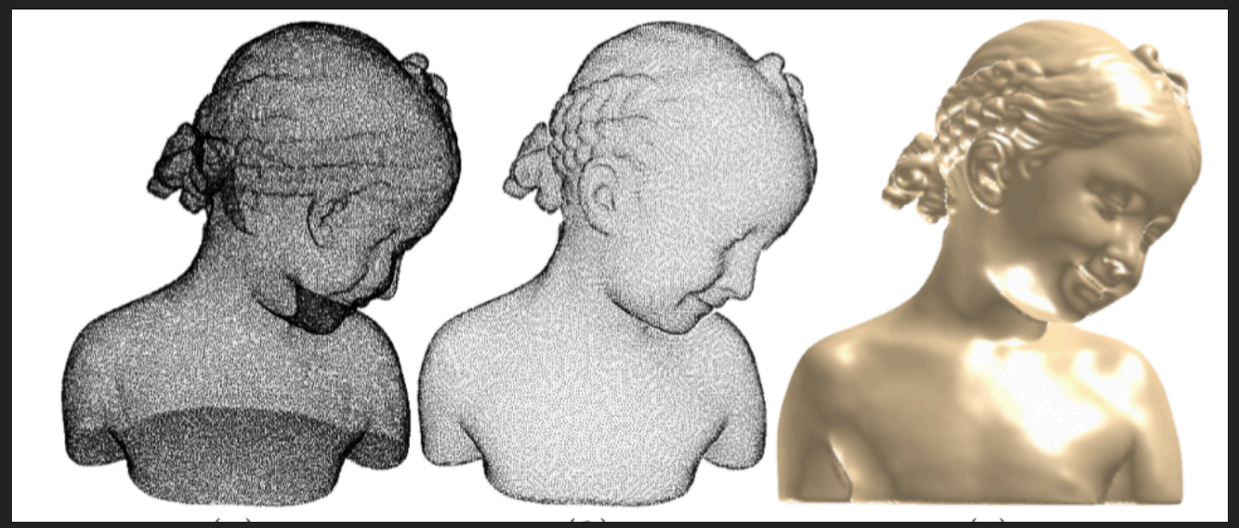
NYU COURANT INSTITUTE OF MATHEMATICAL SCIENCES

NYU CENTER FOR DATA SCIENCE

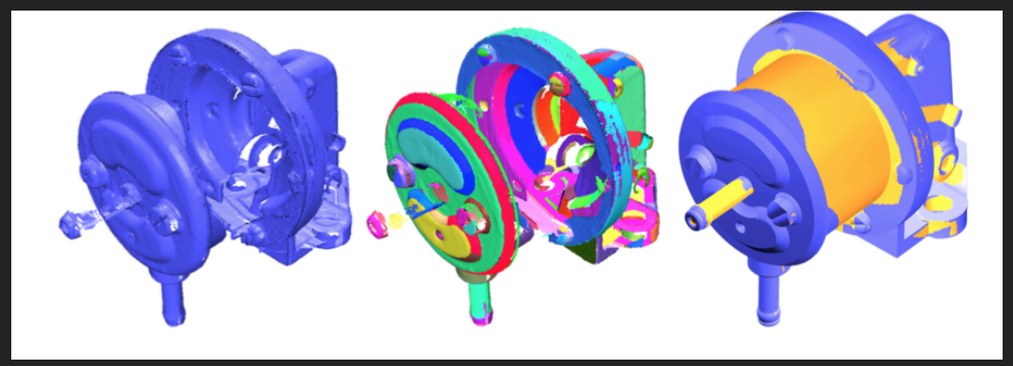
FRANCIS WILLIAMS, TESEO SCHNEIDER, CLAUDIO SILVA, MATTHEW TRAGER, DENIS ZORIN, JOAN BRUNA, DANIELE PANOZZO

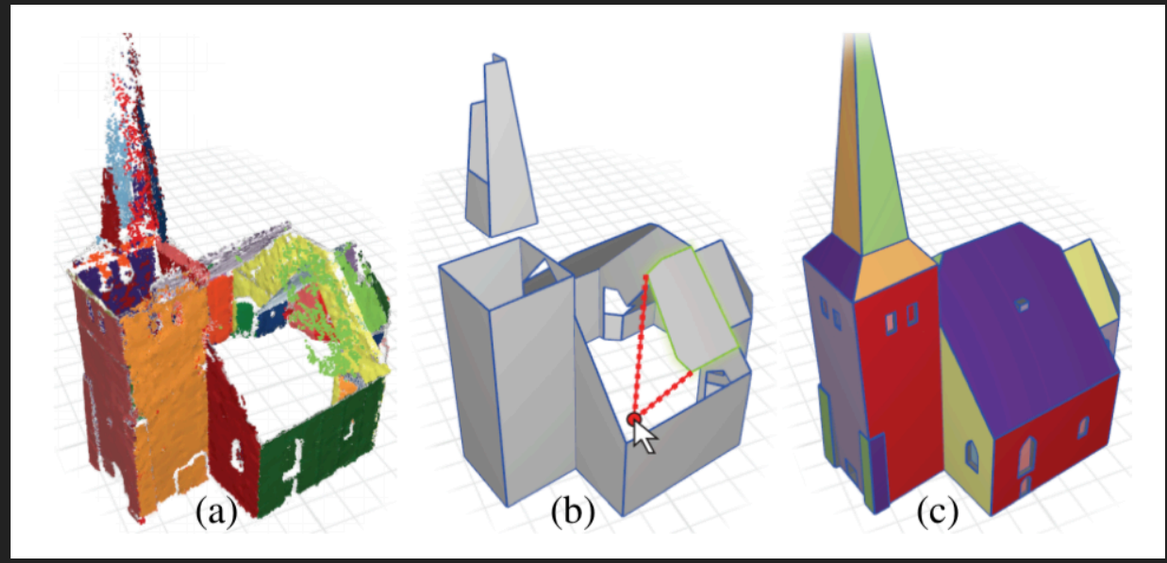
DEEP GEOMETRIC PRIOR FOR SURFACE RECONSTRUCTION

- Range Scans, LIDAR, Depth-map.



(figures from [Berger et al.])





- Geometric Inverse Problem: Reconstruct a continuous surface from a collection of noisy, irregularly sampled points (and perhaps normals).

SURFACE RECONSTRUCTION: CANONICAL APPROACHES

- Energy-based Methods
  - Poisson Reconstruction
  - Fourier/Wavelet
- Point-to-Set Methods
  - EAR
- Partitions of Unity/ Kernel methods
- Scattering Point Meshes.

- In Machine Learning, implicit models are popular generative models:

$$\min_{\theta} \mathrm{d}(Q_\theta, \hat{P})$$

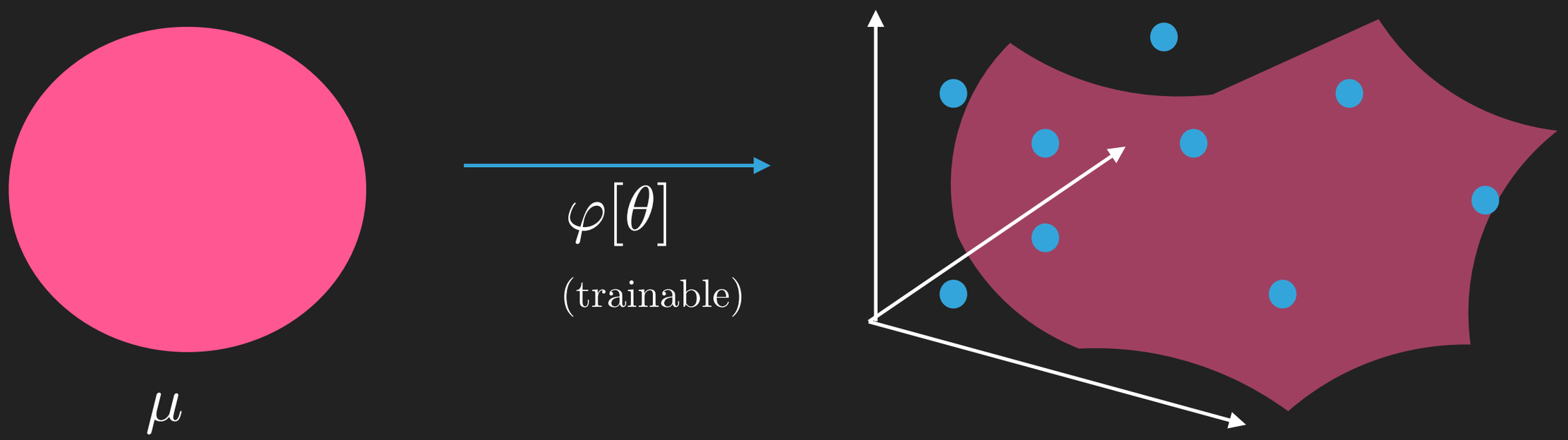
$\hat{P}$: Empirical Data distribution,
$Q_\theta = \varphi[\theta]_{\#}\mu$: Pushforward measure,
d: discrepancy/distance, e.g. MMD, W_2, KL.

- In Machine Learning, implicit models are popular generative models:

$$\min_{\theta} \mathrm{d}(Q_\theta, \hat{P})$$

$\hat{P}$: Empirical Data distribution,
$Q_\theta = \varphi[\theta]_{\#}\mu$: Pushforward measure,
d: discrepancy/distance, e.g. MMD, W_2, KL.

- If d is an Integral Probability Metric, this corresponds to GANs.



- Overparametrised Deep Neural Networks are powerful "curve-fitting" models.
- Empirical Risk Minimisation:

$$\min_{\theta} \widehat{\mathcal{R}}(\theta) := \frac{1}{L} \sum_{l} \ell(f_\theta(x_l), y_l) \ , \ (x_l, y_l) \sim_{\text{iid}} P \, .$$

- Overparametrised Deep Neural Networks are powerful "curve-fitting" models.
- Empirical Risk Minimisation:

$$\min_{\theta} \widehat{\mathcal{R}}(\theta) := \frac{1}{L} \sum_{l} \ell(f_\theta(x_l), y_l) \ , \ (x_l, y_l) \sim_{\text{iid}} P \, .$$

- Guarantees for global minimisation using SGD/GD.
- In the *interpolant* regime $(\widehat{\mathcal{R}}(\theta^*) = 0)$, what is the population risk $\mathcal{R}(\theta^*) = \mathbb{E}_P \, \ell(f_{\theta^*}(x), y)$?
- In the noisy setting, does gradient descent provide implicit regularization through so-called *early-stopping*?

THIS WORK

- We treat surface reconstruction as a “low-dimensional” implicit modeling using ReLU networks.

- Wasserstein Metric provides chart correspondences, resulting in globally consistent atlas.

- Analysis for large neural networks: gradient descent can operate in two distinct regimes: kernel or “lazy” regime, or feature selection regime.

PROBLEM SETUP

- $\mathcal{S}$: Surface in $\mathbb{R}^3$, possibly with boundary.
- $\mathcal{X} = \{x_i = y_i + w_i; i \leq n, w_i \sim \nu, y_i \sim \rho_{\mathcal{S}}\}$

$\rho_{\mathcal{S}}$: measure on $\mathcal{S}$.
ν: noise distribution.

- Goal: estimate $\mathcal{S}$ from $\mathcal{X}$.

PROBLEM SETUP

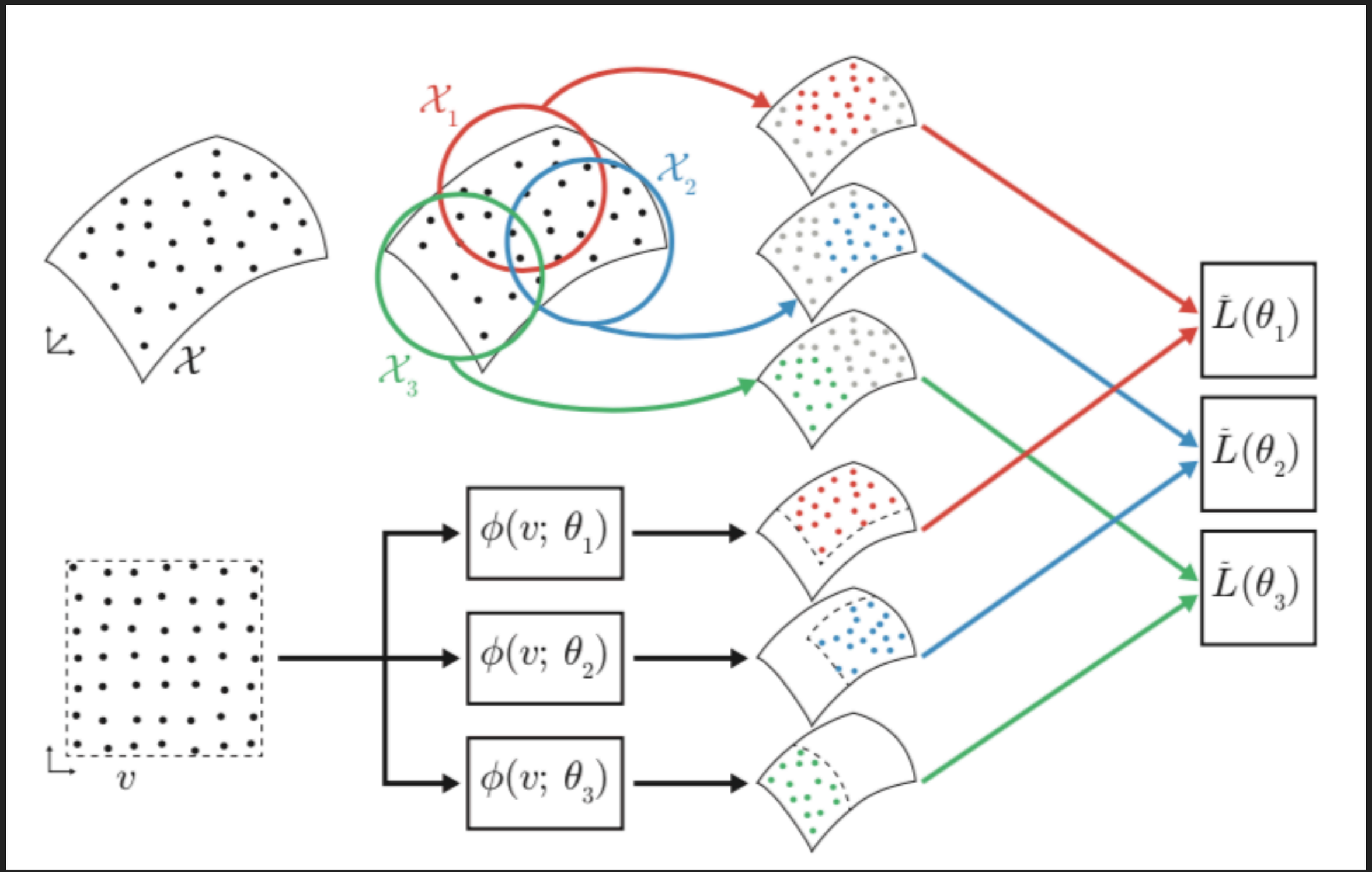
- $\mathcal{S}$: Surface in $\mathbb{R}^3$, possibly with boundary.
- $\mathcal{X} = \{x_i = y_i + w_i; i \leq n, w_i \sim \nu, y_i \sim \rho_{\mathcal{S}}\}$

$\rho_{\mathcal{S}}$: measure on $\mathcal{S}$.
ν: noise distribution.

- Goal: estimate $\mathcal{S}$ from $\mathcal{X}$.

- Challenges
  - $\mathcal{X}$ may contain holes and acquisition noise.
  - $\mathcal{S}$ may contain singularities e.g. in corners, tips. and thin regions.

OVERALL MODEL



LOCAL PARAMETRIZATION

- First we consider a local surface parametrization:
- Given $p \in \mathcal{S}$, consider small neighborhood $\mathcal{U}_p$.
- For sufficiently small radius, $\mathcal{S} \cap \mathcal{U}_p \cong V = (0,1)^2$
- Thus, exists a differentiable mapping $\varphi : V \to \mathcal{U}_p$ such that $\varphi(V) = \mathcal{S} \cap \mathcal{U}_p$.

LOCAL PARAMETRIZATION

- First we consider a local surface parametrization:
- Given $p \in \mathcal{S}$, consider small neighborhood $\mathcal{U}_p$.
- For sufficiently small radius, $\mathcal{S} \cap \mathcal{U}_p \cong V = (0,1)^2$
- Thus, exists a differentiable mapping $\varphi : V \to \mathcal{U}_p$ such that $\varphi(V) = \mathcal{S} \cap \mathcal{U}_p$.
- For each patch, we fit an implicit model $\varphi_p : V \to \mathbb{R}^3$ with

$$\min_{\theta_p} W_2^2(\varphi[\theta_p]_{\#}\mu, \mathcal{X} \cap \mathcal{U}_p) \ .$$

  - $\varphi[\theta]$ parametrized as a neural network.
  - μ is a Poisson Disk sampling in V.

- Since we are dealing with empirical distributions, the Wasserstein distance is computed as an EMD:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\pi \in \Pi_n} \sum_{i=1}^{n} \|x_{\pi(i)} - \varphi[\theta](v_i)\|^2 \ .$$

$$\mathcal{X} \cap \mathcal{U}_p = \{x_1, \ldots, x_n\} \ , \ \{v_1 \ldots v_n\} \sim \mu.$$

- Since we are dealing with empirical distributions, the Wasserstein distance is computed as an EMD:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\pi \in \Pi_n} \sum_{i=1}^{n} \|x_{\pi(i)} - \varphi[\theta](v_i)\|^2 \ .$$

$$\mathcal{X} \cap \mathcal{U}_p = \{x_1, \ldots, x_n\} \ , \ \{v_1 \ldots v_n\} \sim \mu.$$

- Linear Assignment problem: can be solved, e.g with network simplex, but expensive $O(n^3)$.

- Since we are dealing with empirical distributions, the Wasserstein distance is computed as an EMD:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\pi \in \Pi_n} \sum_{i=1}^{n} \|x_{\pi(i)} - \varphi[\theta](v_i)\|^2 \ .$$

$$\mathcal{X} \cap \mathcal{U}_p = \{x_1, \ldots, x_n\} \ , \ \{v_1 \ldots v_n\} \sim \mu.$$

- Linear Assignment problem: can be solved, e.g with network simplex, but expensive $O(n^3)$.

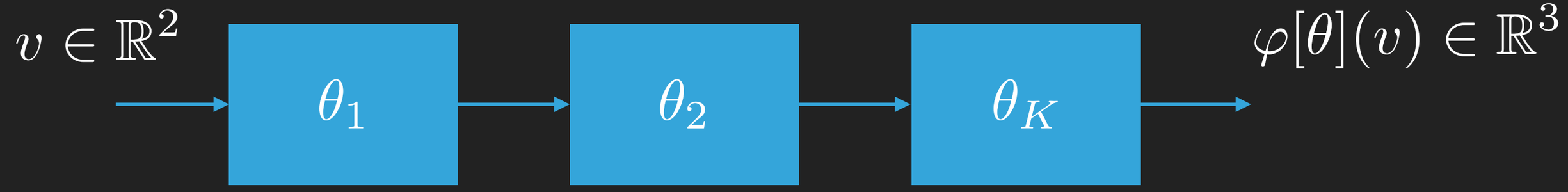
- Instead, we use Sinkhorn regularization [Cuturi,'13]:

$$\widetilde{\mathcal{L}}(\theta) = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} P_{i,j} \|x_j - \varphi[\theta](v_i)\|^2 - \lambda^{-1} H(P) \ ,$$

$\mathcal{P}_n$: bi-stochastic matrices, $H(P) = -\sum_{i,j} P_{i,j} \log P_{i,j}$.

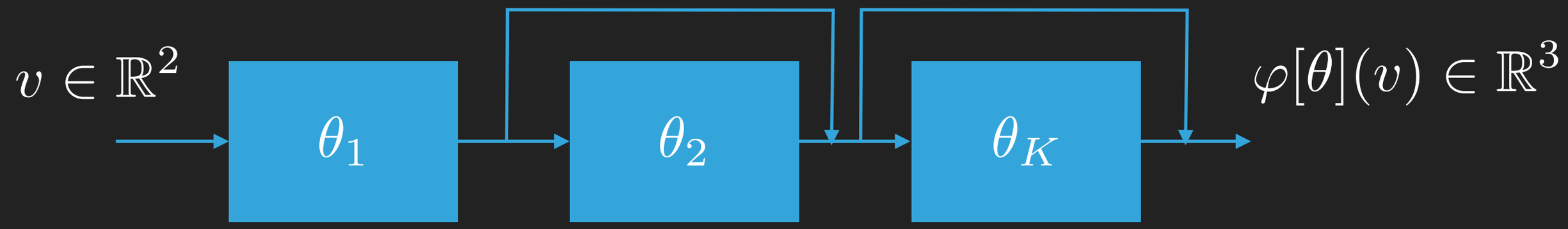
  - It defines a distance that converges to W_2 as $\lambda \to \infty$.
  - Near-linear time complexity [Altshuler et al.'17]

- We use a Multilayer Perceptron to parametrize the push forward map:



$$z^{(k+1)} = \sigma(\theta_{k,1} z^{(k)} + \theta_{k,2})$$

- Also, its residual reparametrisation:



$$z^{(k+1)} = z^{(k)} + \sigma(\theta_{k,1} z^{(k)} + \theta_{k,2}) \, .$$

- We consider $\sigma(u) = \max(0, u)$, so the resulting local surface model is piece-wise linear.
- As opposed to triangulations, in our setting we do not require $x_i \in \mathcal{X}$ to be nodes (not necessarily interpolant).

LOCAL PARAMETRIZATION

$$\widetilde{\mathcal{L}}(\theta) = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} P_{i,j} \|x_j - \varphi[\theta](v_i)\|^2 - \lambda^{-1} H(P) \ ,$$

- For fixed *P*, this problem is a least-squares regression using an over-parametrized ReLU network

$$\widetilde{\mathcal{L}}(\theta) = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} P_{i,j} \|x_j - \varphi[\theta](v_i)\|^2 - \lambda^{-1} H(P) \ ,$$

- For fixed *P*, this problem is a least-squares regression using an over-parametrized ReLU network.
- Several authors [Du et al.'18,Venturi et al.'18,Oymak et al.'19] observe that gradient descent reaches global optimum in such overparametrised regime.

$$\widetilde{\mathcal{L}}(\theta) = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} P_{i,j} \|x_j - \varphi[\theta](v_i)\|^2 - \lambda^{-1} H(P) \ ,$$

- For fixed *P*, this problem is a least-squares regression using an over-parametrized ReLU network
- Several authors [Du et al.'18,Venturi et al.'18,Oymak et al.'19] observe that gradient descent reaches global optimum in such overparametrised regime.
- Since this is independent of *P*, why do we obtain meaningful parametrisations?

$$\widetilde{\mathcal{L}}(\theta) = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} P_{i,j} \|x_j - \varphi[\theta](v_i)\|^2 - \lambda^{-1} H(P) \ ,$$

- For fixed *P*, this problem is a least-squares regression using an over-parametrized ReLU network
- Several authors [Du et al.'18,Venturi et al.'18,Oymak et al.'19] observe that gradient descent reaches global optimum in such overparametrised regime.
- Since this is independent of *P*, why do we obtain meaningful parametrisations?
  - $\varphi[\theta]$ is a Lipschitz map, with $\mathrm{Lip}(\varphi[\theta]) \lesssim \|\theta\|$
    - Early Sinkhorn iterations quickly lock onto correspondence with smallest distortion.

BUILDING A GLOBAL ATLAS

- So far, we have considered the problem of finding local charts $(V_p, \varphi_p = \varphi[\theta_p^*]), p \in \mathcal{S}$.
- In order to build an atlas $\{(V_q, \varphi_q), q \in Q\}, Q : \text{anchor points.}$, we need to ensure consistent transitions between overlapping charts.

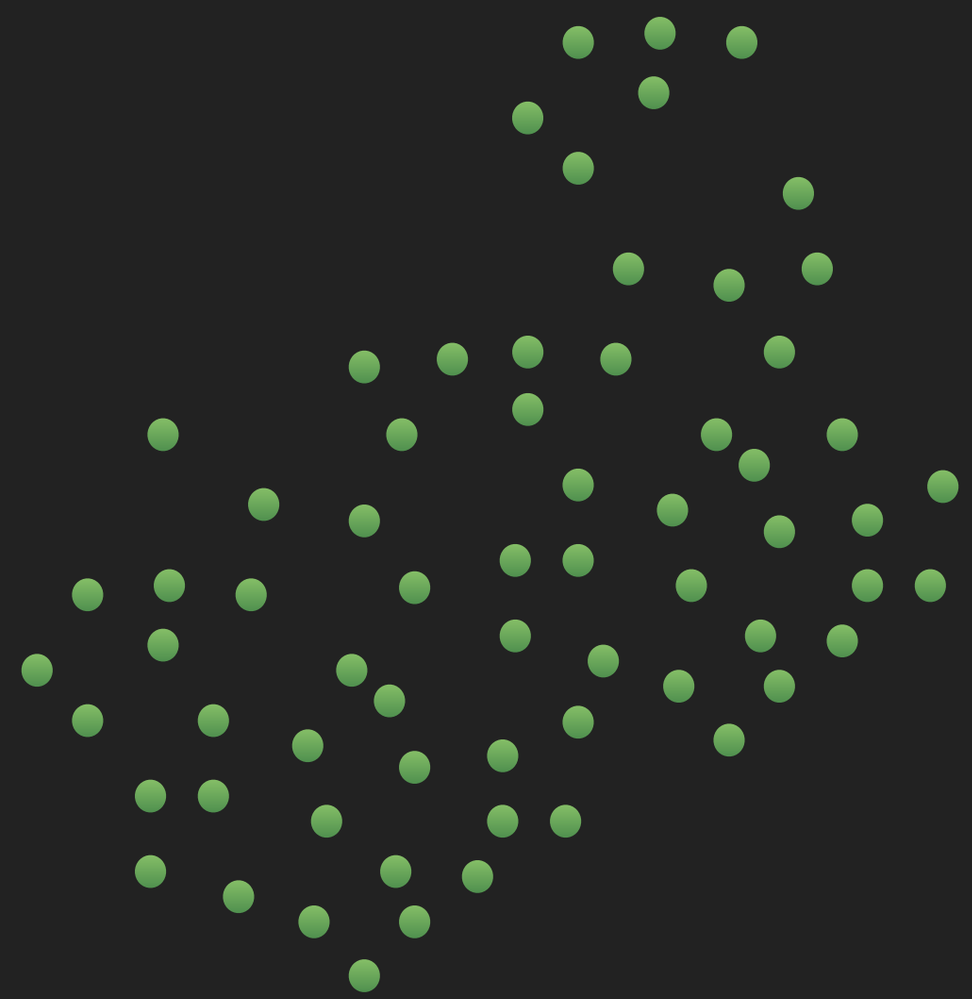
BUILDING A GLOBAL ATLAS

- So far, we have considered the problem of finding local charts $(V_p, \varphi_p = \varphi[\theta_p^*]), p \in \mathcal{S}$.
- In order to build an atlas $\{(V_q, \varphi_q), q \in Q\}, Q : \text{anchor points.}$, we need to ensure consistent transitions between overlapping charts.
- Two charts centered at p, q overlap if $\mathcal{X}_{p,q} = \mathcal{X}_p \cap \mathcal{X}_q \neq \emptyset$.
- Two overlapping charts are consistent if

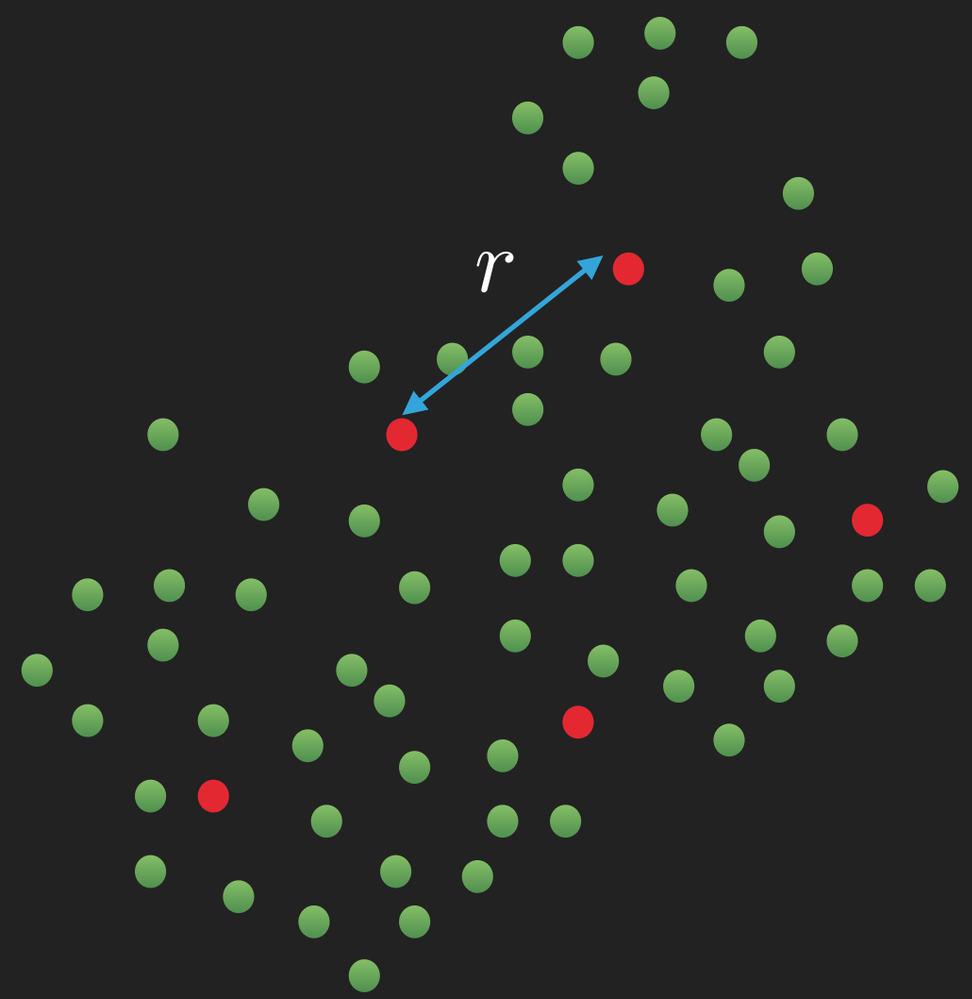
$$\varphi_q(v_{q,\pi_q^{-1}(i)}) = \varphi_p(v_{p,\pi_p^{-1}(i)}) \, \forall x_i \in \mathcal{X}_{p,q} \, .$$

  - well defined since each chart has its associated permutation.
  - consistency is guaranteed if the charts are *interpolating*.

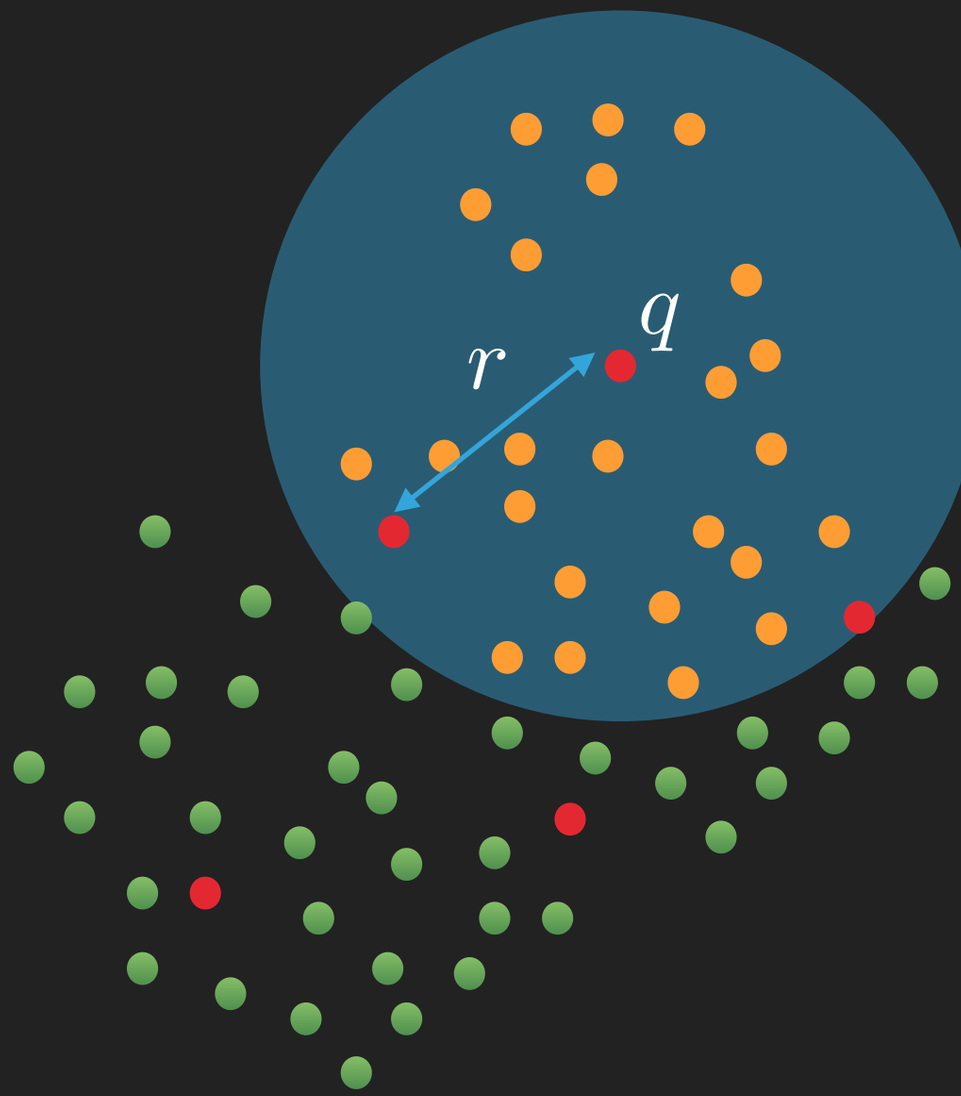
- We use Poisson Disk Sampling on full point cloud $\mathcal{X}$ with radius r.



- We use Poisson Disk Sampling on full point cloud $\mathcal{X}$ with radius *r*.

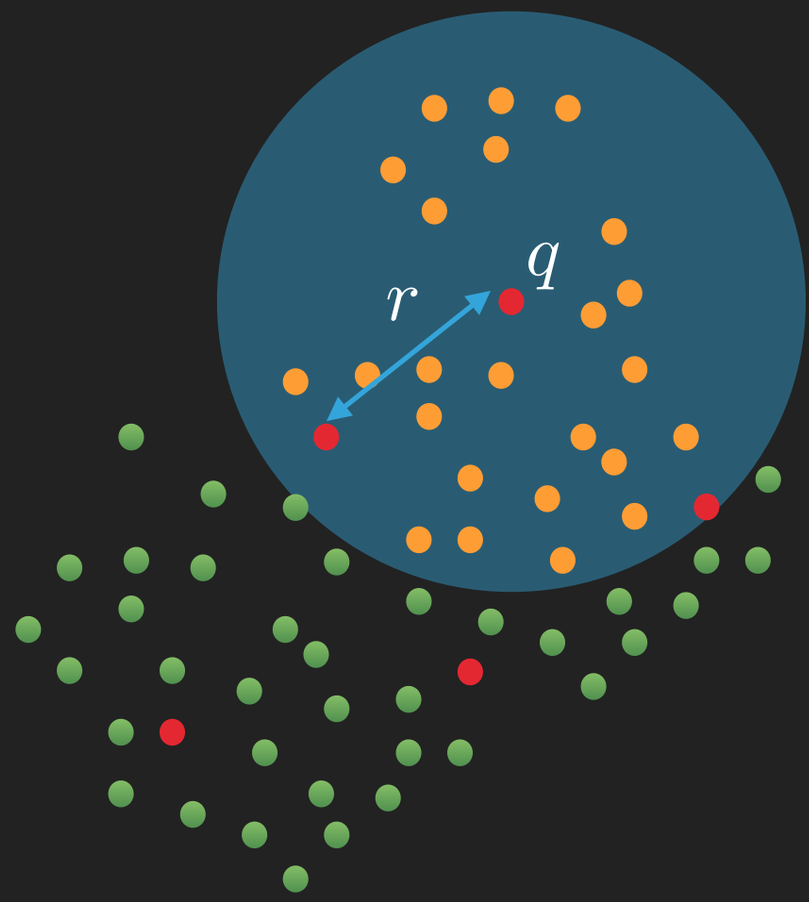


- We use Poisson Disk Sampling on full point cloud $\mathcal{X}$ with radius *r*.
- We intersect $\mathcal{X}$ with balls $B(q, cr), c > 1.$ to obtain $\mathcal{X}_q$.



BUILDING A GLOBAL ATLAS

- We use Poisson Disk Sampling on full point cloud $\mathcal{X}$ with radius *r*.
- We intersect $\mathcal{X}$ with balls $B(q, cr), c > 1.$ to obtain $\mathcal{X}_q$.
- We use heuristic using normals to filter out points belonging to different sheets.



- Full model is fit by minimizing

$$\min_{\theta_1,\ldots,\theta_Q,\pi_1,\ldots\pi_q} \sum_q \mathcal{L}_q(\theta_q,\pi_q) + \sum_{q,q';\mathcal{X}_{q,q'}\neq\emptyset} \mathcal{L}_{q,q'}(\theta_q,\theta'_q,\pi_q,\pi_{q'}), \text{with}$$

$$\mathcal{L}_q(\theta,\pi) = \sum_{i=1}^{|\mathcal{X}_q|} \|\varphi[\theta](v_i) - x_{\pi(i)}\|^2,$$

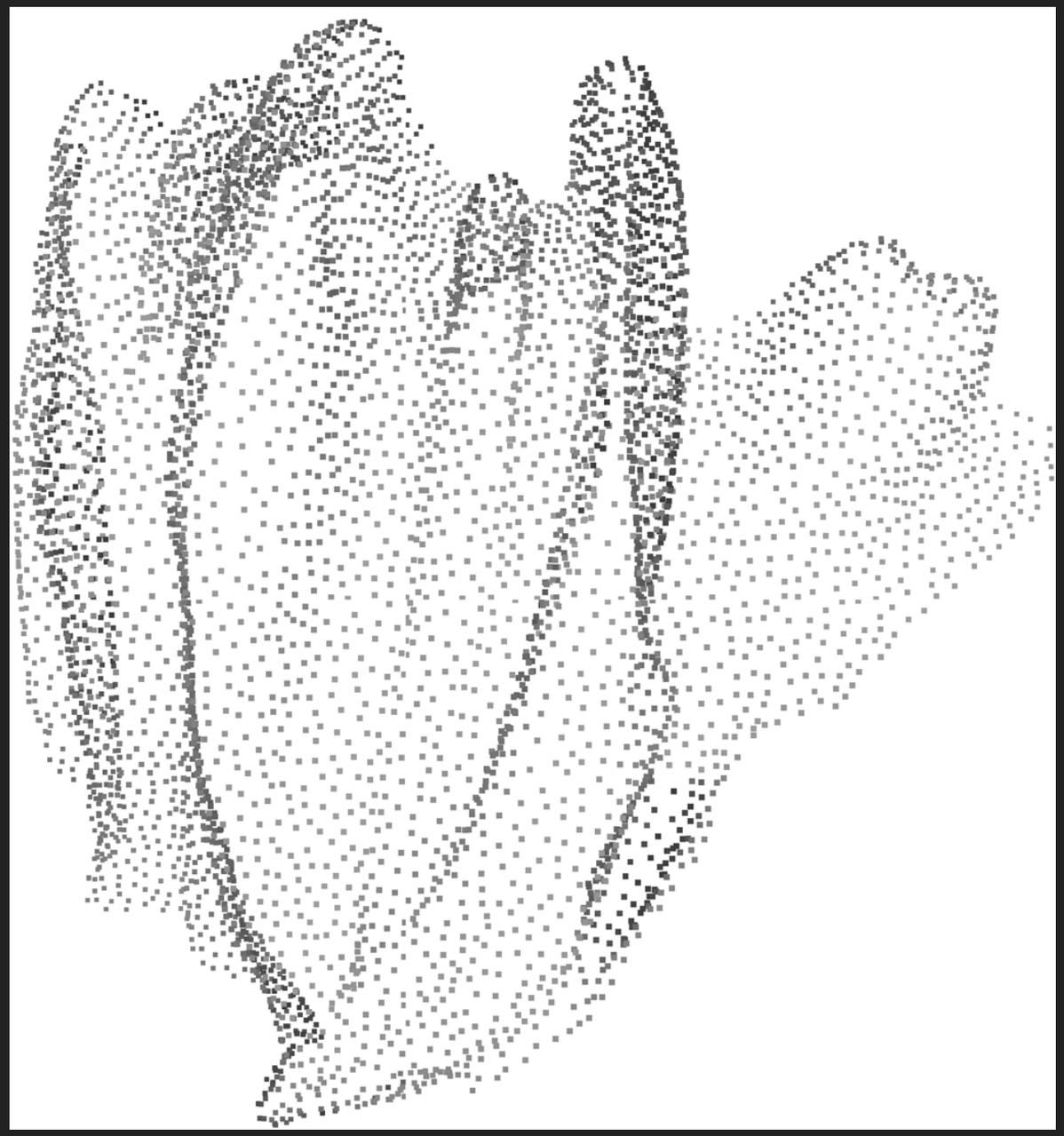
$$\mathcal{L}_{q,q'}(\theta,\theta',\pi,\pi') = \sum_{i=1}^{|\mathcal{X}_{q,q'}|} \|\varphi[\theta](v_i) - \varphi[\theta'](v_{\pi'\circ\pi^{-1}(i)})\|^2.$$

- Easy to parallelize
- In practice, initial fit using only unary terms, fine-tuning using consistency terms.

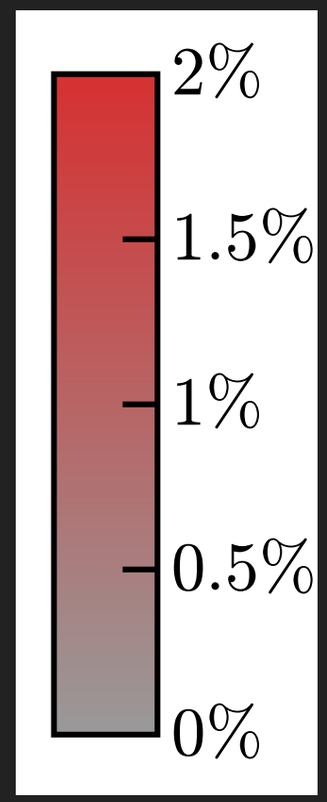
RELATED WORK

- PU Net [Yu et al.CVPR'18]: Network trained to upsample 3d point clouds. No surface parametrisation.
- [Basri & Jacobs,ICLR'17]: Study ability of Neural Networks to represent low-dimensional manifolds. No training dynamics.
- AtlasNet [Groueix et al.'18]: Shape auto encoder with applications to surface reconstruction. No consistent transitions.

- local parametrization (a single chart)

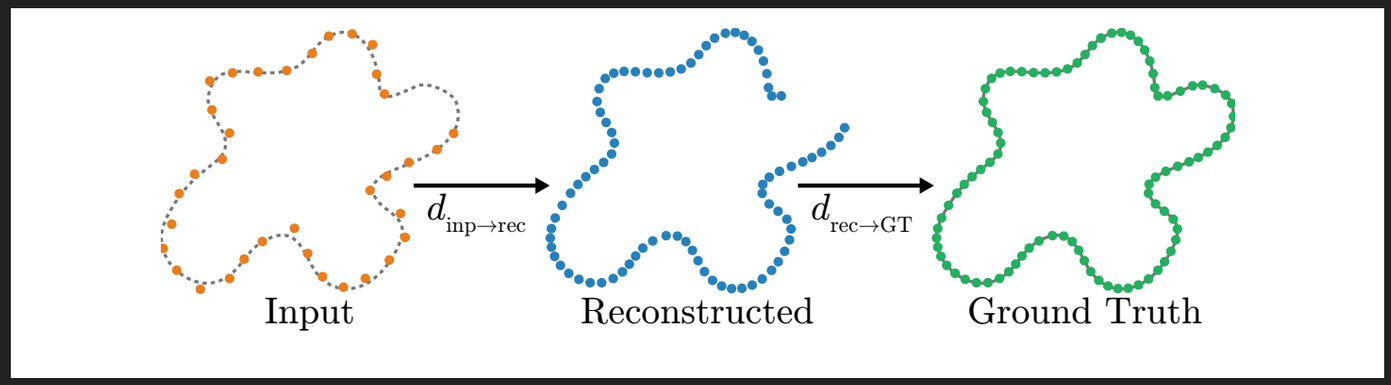


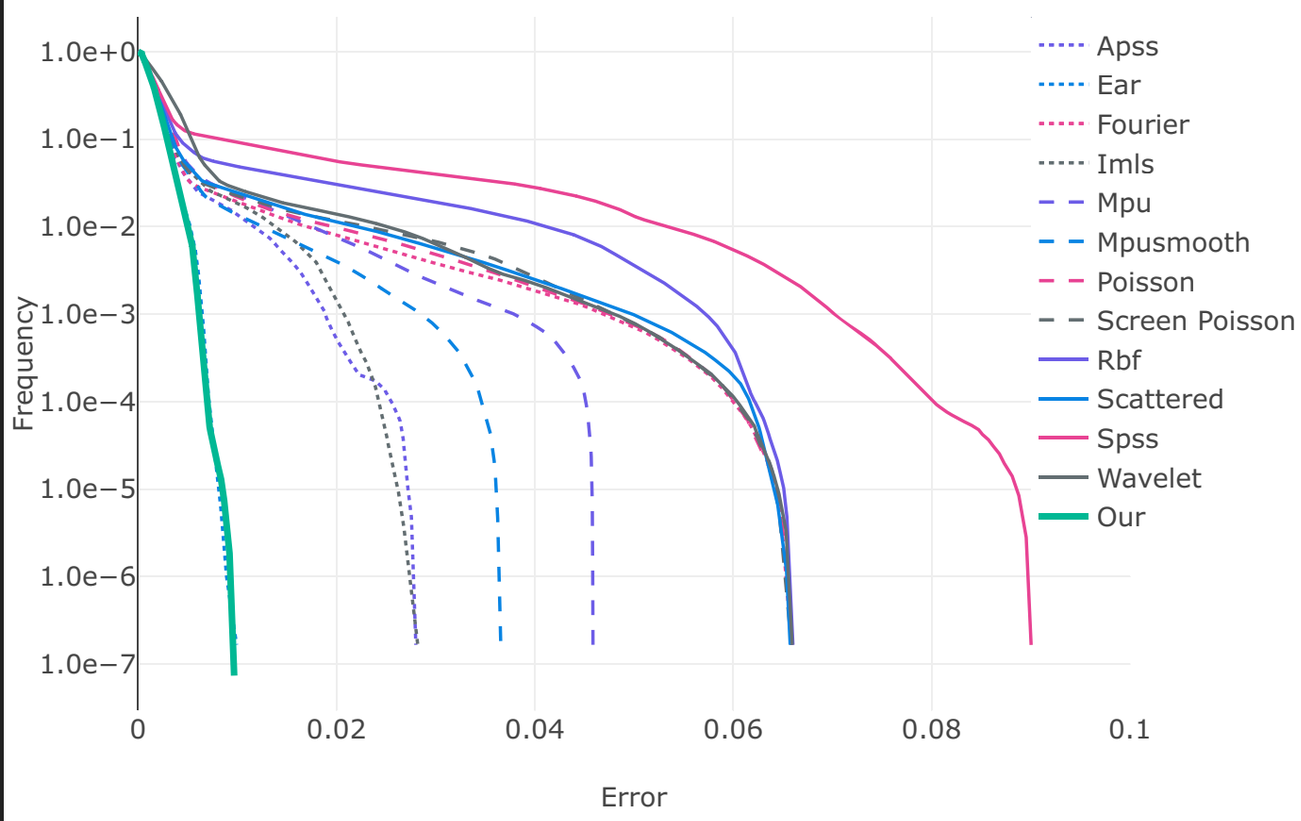
NUMERICAL EXPERIMENTS: GLOBAL ATLAS

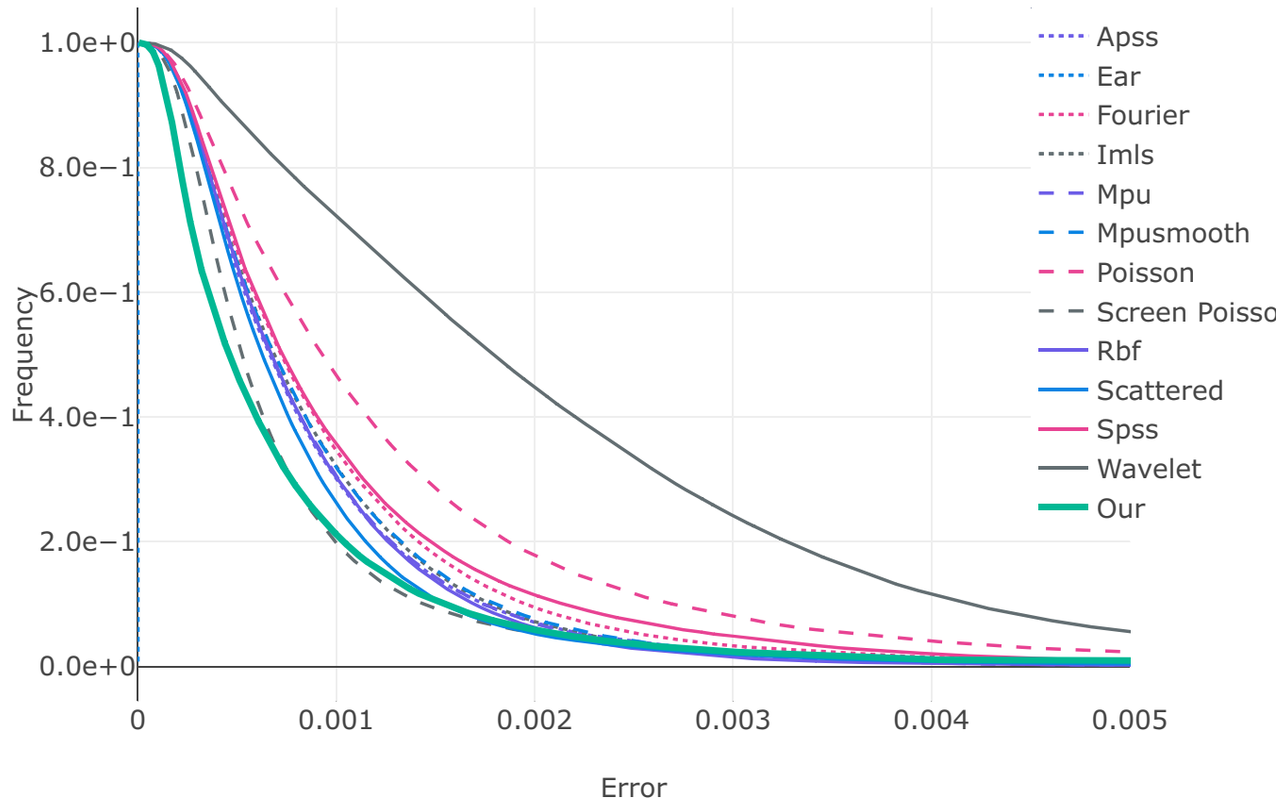


QUANTITATIVE COMPARISONS

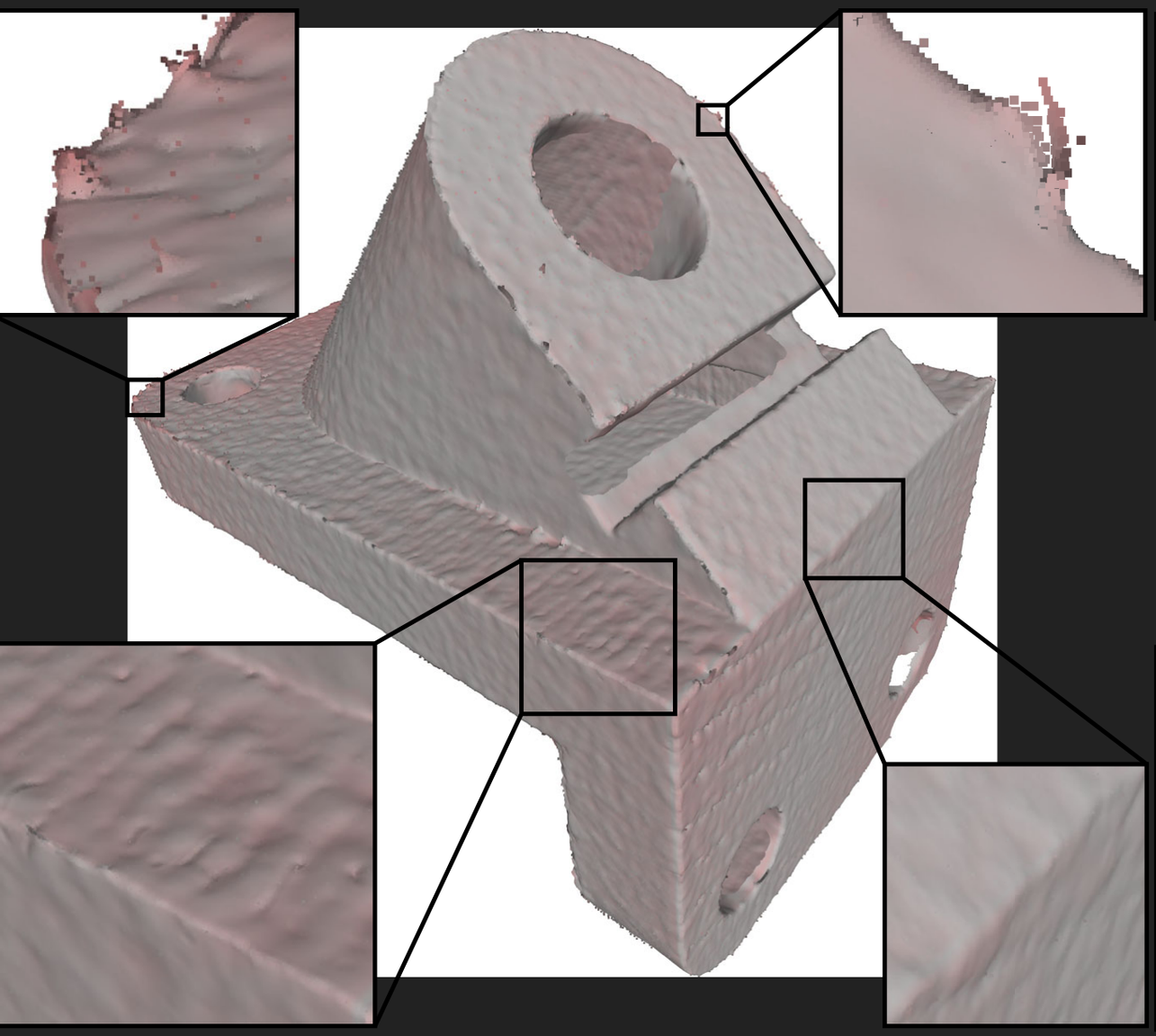
- We measure the percentage of fitted vertices to reach a given error:

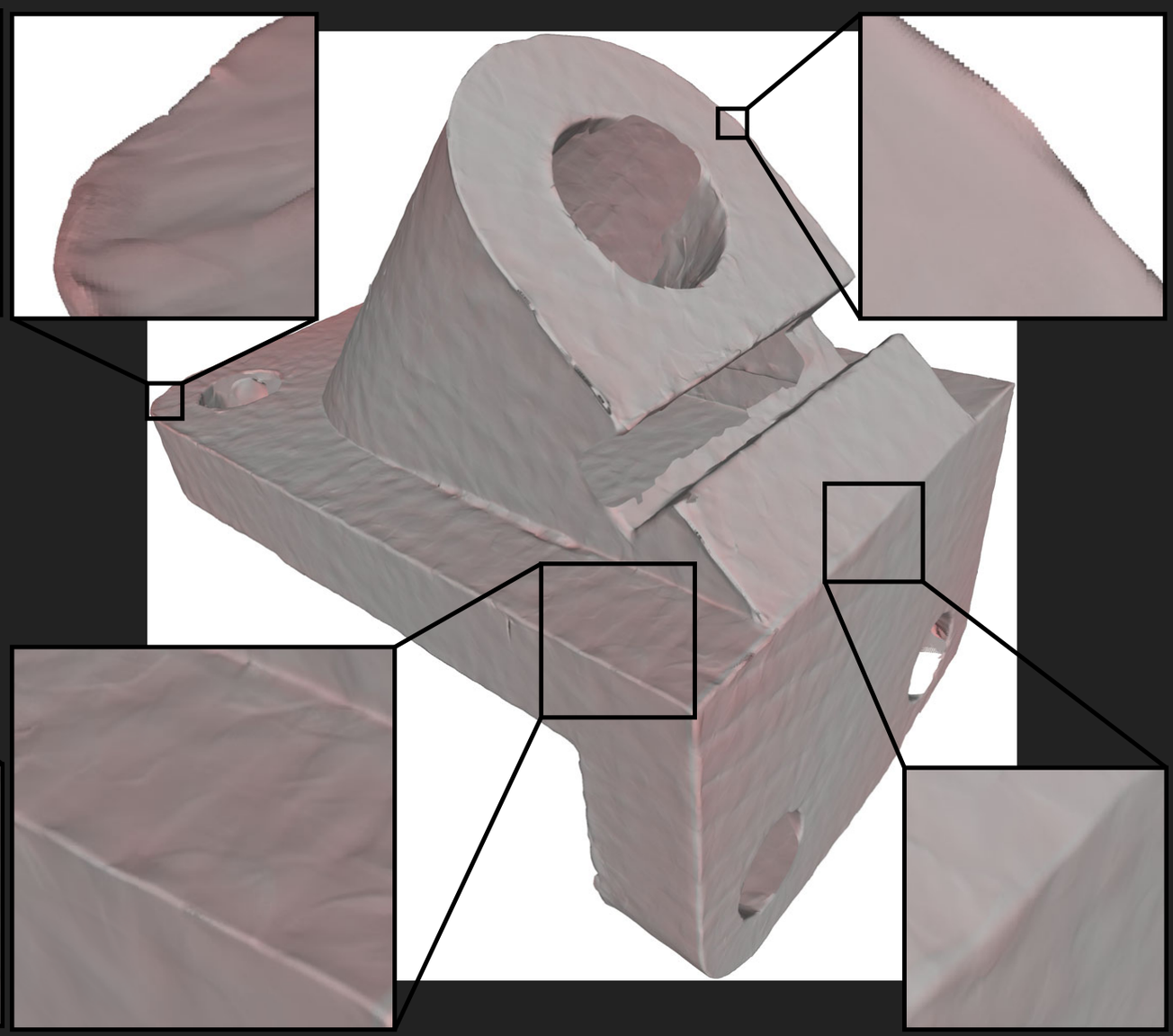




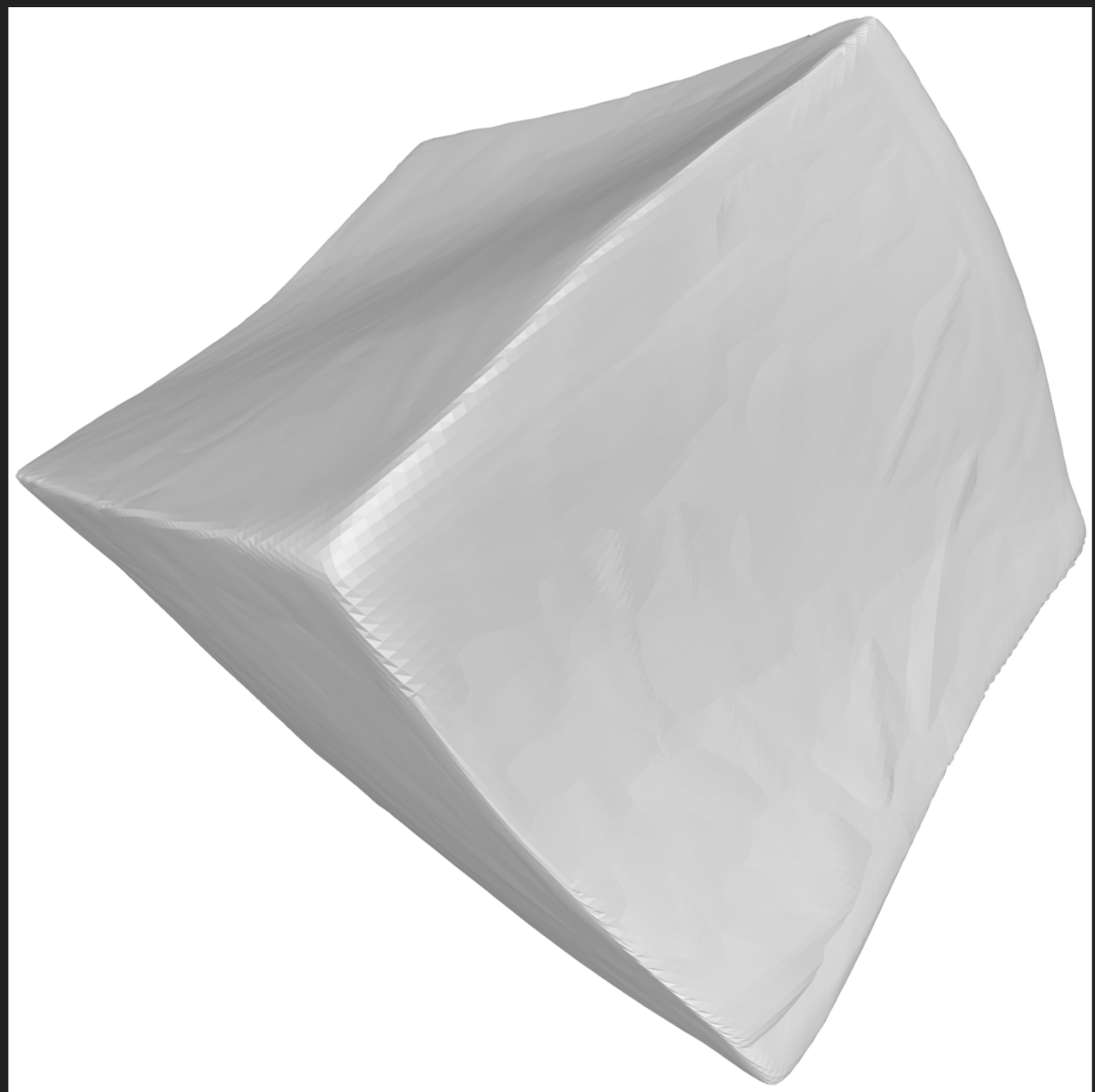


- comparisons with EAR+poisson:

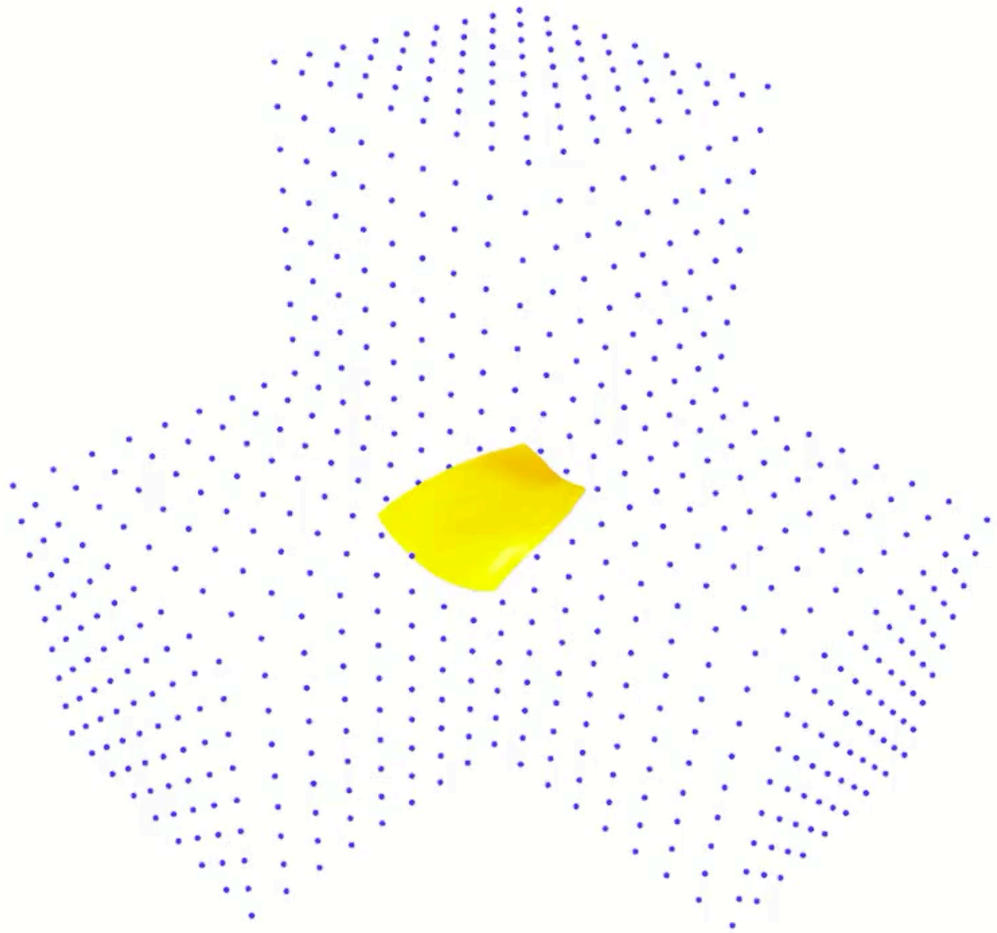




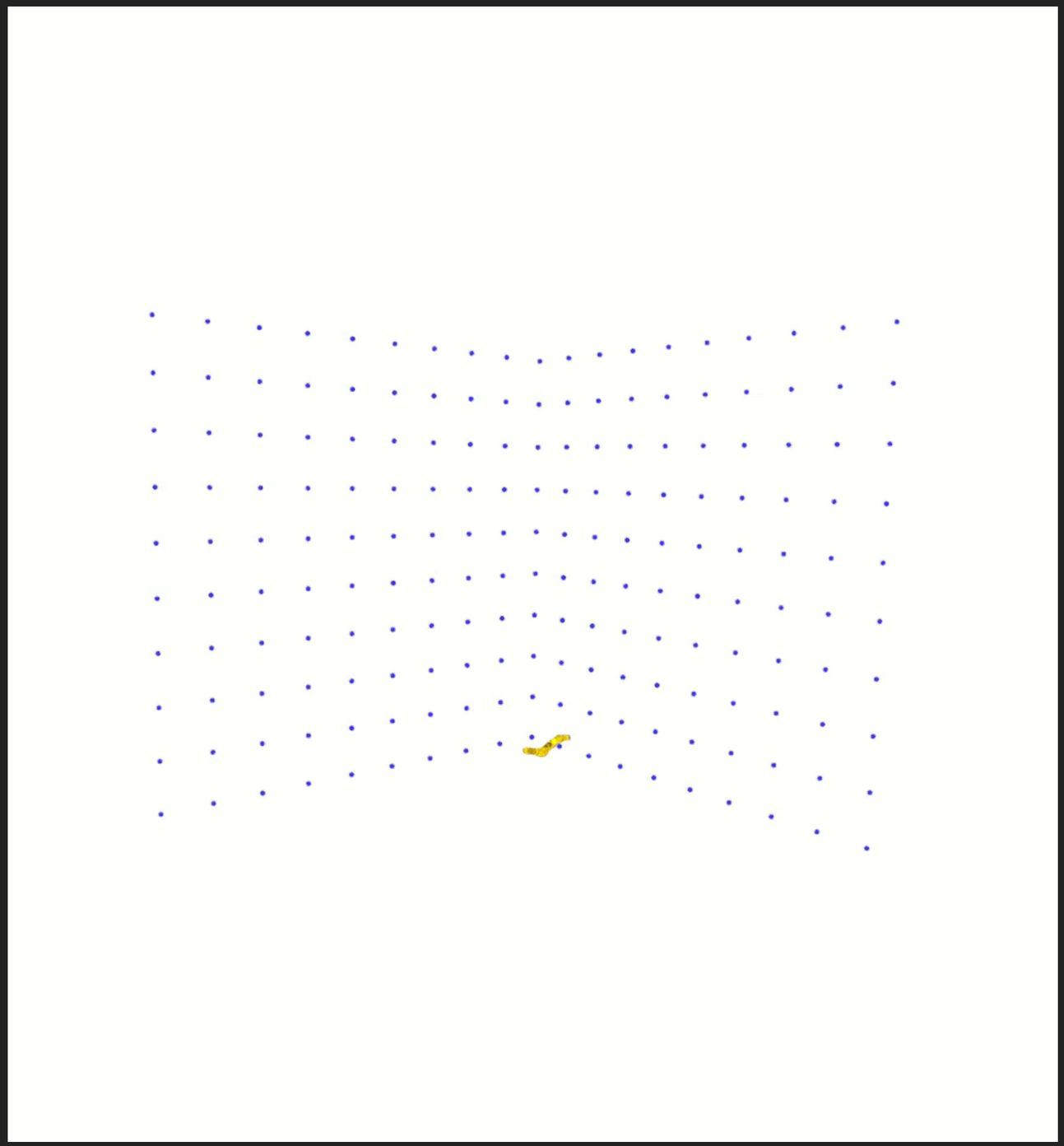
- Comparisons with AtlasNet

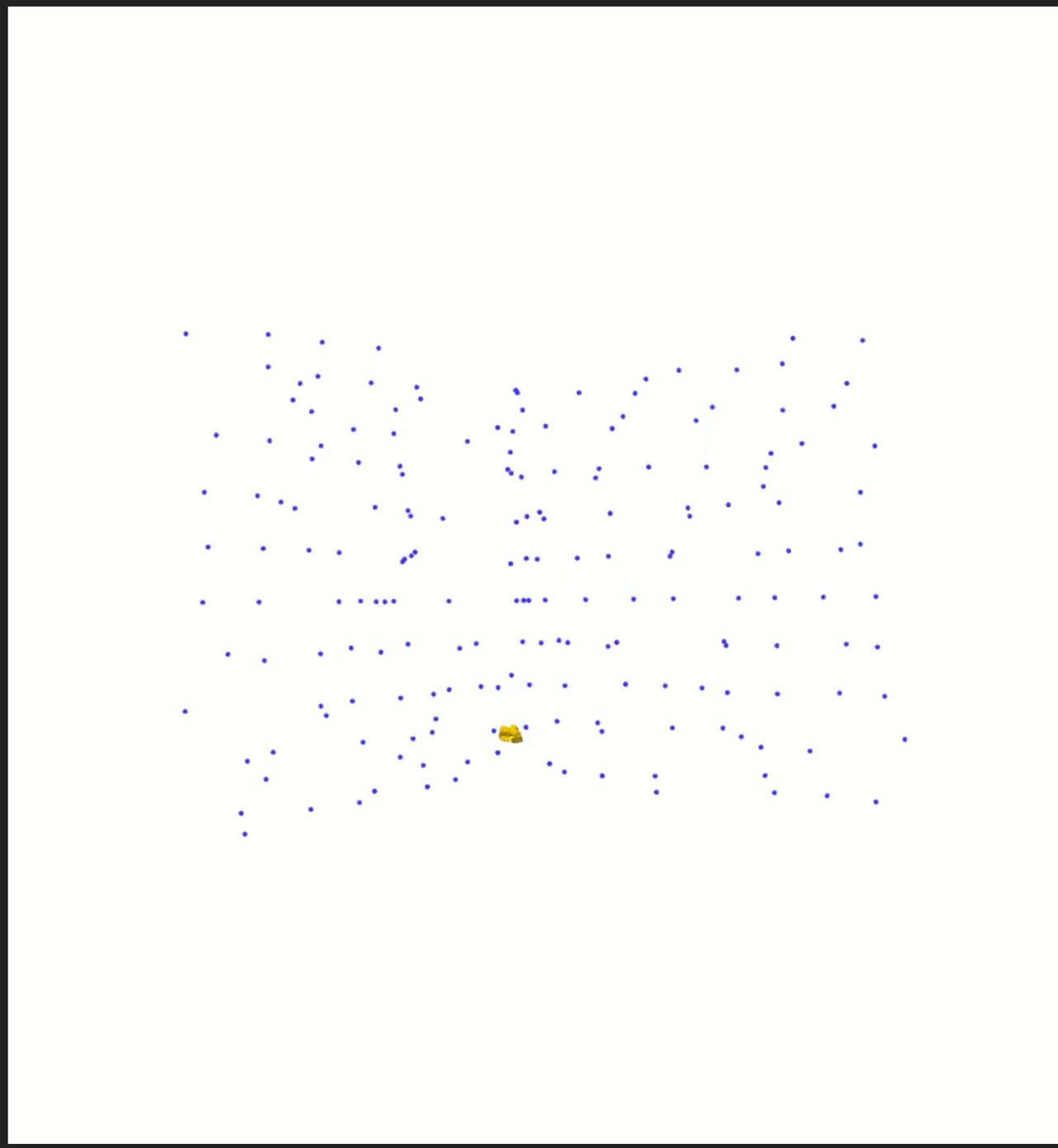


TRAINING DYNAMICS



IMPLICIT REGULARIZATION





ANALYSIS

- Analysis of the model in the *overparametrised* regime, considering a simple one hidden-layer architecture.
- Relationship with kernel methods? Advantages?
- How should the geometry and topology of S affect the network architecture?

- A neural network is simply a parametrized function $f(x;\theta)$

$$\begin{aligned}\varphi : \Theta &\to \mathcal{F} \\ \theta &\mapsto f_\theta = f(\cdot, \theta)\end{aligned}$$

- Consider a canonical regression problem of the form

$$\min_\theta \mathcal{R}(f_\theta, f^*)$$

- A neural network is simply a parametrized function $f(x;\theta)$

$$\varphi : \Theta \to \mathcal{F}$$
$$\theta \mapsto f_\theta = f(\cdot, \theta)$$

- Consider a canonical regression problem of the form

$$\min_\theta \mathcal{R}(f_\theta, f^*)$$

- Gradient-based learning (in continuous time) becomes

$$\dot{\theta}(t) = -\nabla\varphi(\theta(t)) \cdot \mathcal{R}'(f_{\theta(t)}, f^*)\,, \text{thus}$$

$$\dot{f}(t) = -\underbrace{\nabla\varphi^\top(t)\nabla\varphi(t)}_{\mathcal{K}(t)} \cdot \mathcal{R}'(f(t), f^*)\,.$$

  - $\mathcal{K}(t)$ is the so-called *Neural Tangent Kernel* [Jacot et al.'18]

KERNEL/LAZY REGIME

- For wide neural networks initialized such that covariance is preserved, NTK $\mathcal{K}(t)$ does not move during gradient descent in the infinite-width limit [Jacot et al.'18].

- For wide neural networks initialized such that <u>covariance is preserved</u>, NTK $\mathcal{K}(t)$ does not move during gradient descent in the infinite-width limit[Jacot et al.'18].
  - In that regime, in the limit of infinite width, Neural Networks *are* Kernel machines:

$$\hat{f}(x) = \sum_{j=1}^{m} \beta_i \mathcal{K}(0)(x, x_j), \text{with} \quad \beta = \arg\min \|\beta\|^2 \; s.t. \hat{f}(x_j) = y_j.$$

  - Also called "lazy" training [Chizat, Bach,'18], since neurons do not move.

KERNEL/LAZY REGIME

- For wide neural networks initialized such that <u>covariance is preserved</u>, NTK $\mathcal{K}(t)$ does not move during gradient descent in the infinite-width limit[Jacot et al.'18].
  - In that regime, in the limit of infinite width, Neural Networks are Kernel machines:

$$\hat{f}(x) = \sum_{j=1}^{m} \beta_i \mathcal{K}(0)(x, x_j), \text{with} \quad \beta = \arg\min \|\beta\|^2 \; s.t. \hat{f}(x_j) = y_j.$$

  - Also called "lazy" training [Chizat, Bach,'18], since neurons do not move.
  - In the case of single hidden-layer, kernel becomes

$$f(x;\theta) = \frac{1}{\sqrt{n}} \sum_{i \leq n} \phi(x;\theta_i) \; , \; \theta_i \sim \rho \qquad \phi(x;\theta) = c\sigma(\langle x, a \rangle + b), \theta = (a, b, c).$$

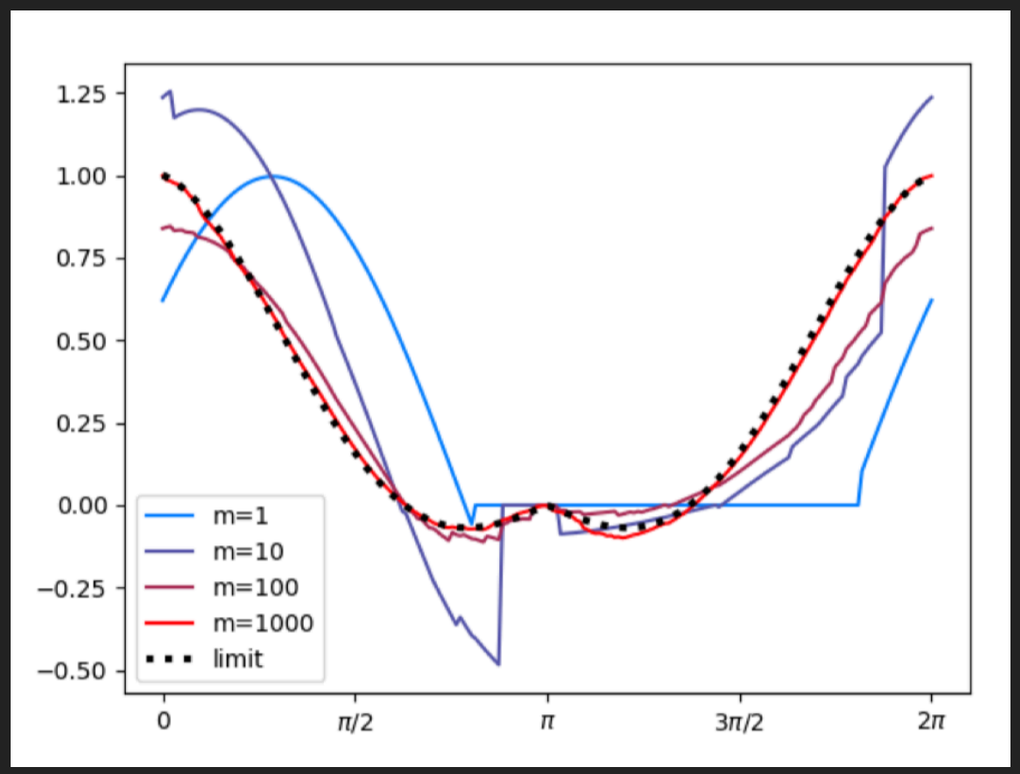
$$\mathcal{K}(x, x') = \frac{1}{n} \sum_i \nabla\phi(x, \theta_i)^\top \nabla\phi(x', \theta_i) \to \mathbb{E}_\rho[\nabla\phi(x, \theta)^\top \nabla\phi(x', \theta)]$$

KERNEL/LAZY REGIME

- What is the Kernel associated to our architecture?

- What is the Kernel associated to our architecture?
- In the single hidden-layer case, the kernel converges to

$$\mathcal{K}(x, x') = \langle x, x' \rangle (\pi - \alpha) + \|x\| \|x'\| ((\pi - \alpha) \cos \alpha + \sin \alpha) \ , \ \ \alpha = \angle(x, x') \ .$$



[Chizat & Bach, '19]

- In the one-dimensional case, this corresponds to cubic spline interpolation.

ACTIVE REGIME

- For appropriate scaling ($1/n$ as opposed to $1/\sqrt{n}$), the model does NOT behave as a kernel.

- Mean field measure in the single hidden-layer case:

$$\mu_t = \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i(t)}$$

- For appropriate scaling ($1/n$ as opposed to $1/\sqrt{n}$), the model does NOT behave as a kernel.
- Mean field measure in the single hidden-layer case:

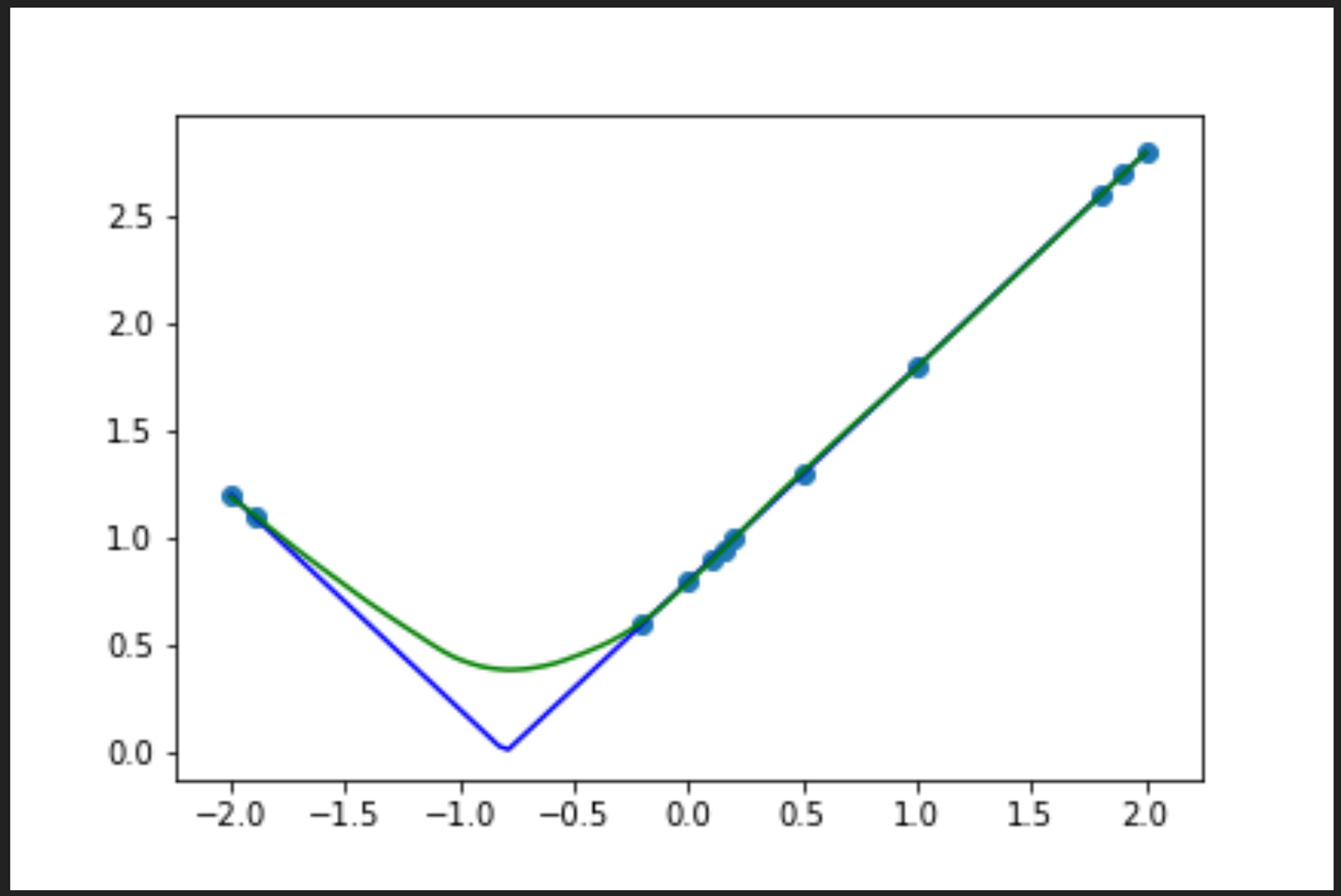
$$\mu_t = \frac{1}{n}\sum_{i=1}^{n} \delta_{\theta_i(t)}$$

- Gradient Dynamics are approximated in the mean-field limit by a PDE given by the continuity equation:

$$\partial_t \mu_t = \operatorname{div}(\nabla V \mu_t) \ , \qquad V(\theta, \mu) = F(\theta) + \int K(\theta, \theta')\mu(d\theta').$$

  - Studied in [Mei, Montanari, Nguyen], [Chizat, Bach], [Rotskoff, Vanden-Eijnden], [Rotskoff, Jelassi, B. EVE].
  - Global convergence under appropriate settings.

- Illustration in 1d curve fitting:



- Adaptivity to local regularity: feature selection.
- Adaptivity to irregular sampling

CONCLUSIONS

- Surface reconstruction as implicit generative modeling.
- Neural networks fit local charts, made consistent using Wasserstein distances.
- No training data required. Expensive inference, high quality results.
- Analysis: low-dimensional counterpart of neural network training *mystery*.
- Role of surface geometry/topology in architecture?
- Extend the analysis to deep architectures.
  - depth = non-local priors?

Thanks!

Reference:

https://arxiv.org/abs/1811.10943