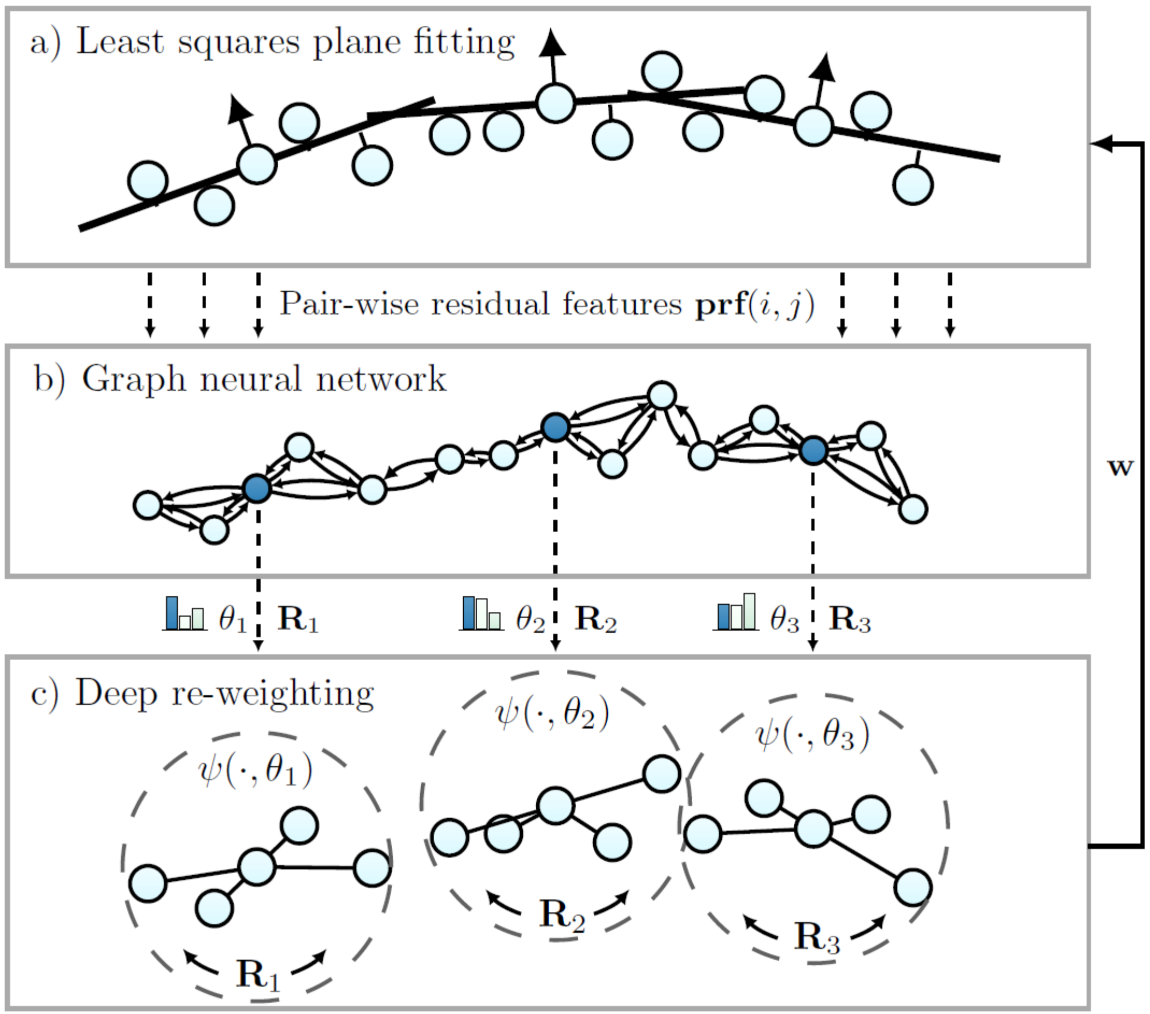
Deep Iterative Surface Normal Estimation

Jan Eric Lenssen

Christian Osendorfer

Jonathan Masci

CVPR 2020 Oral Talk



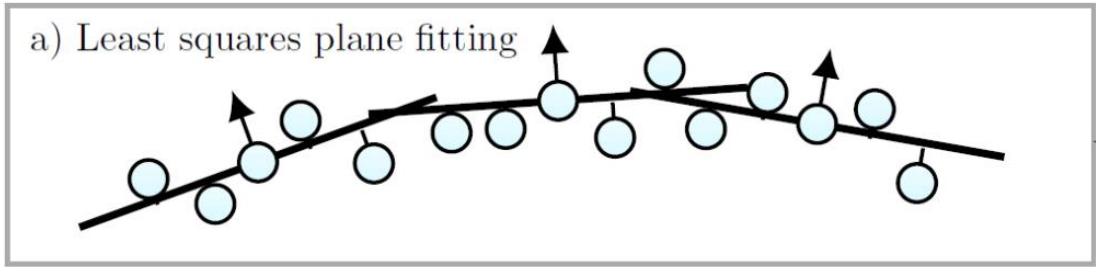
nnaisense

technische universität dortmund

Surface Normal Estimation as Least-Squares

- Unoriented normal estimation

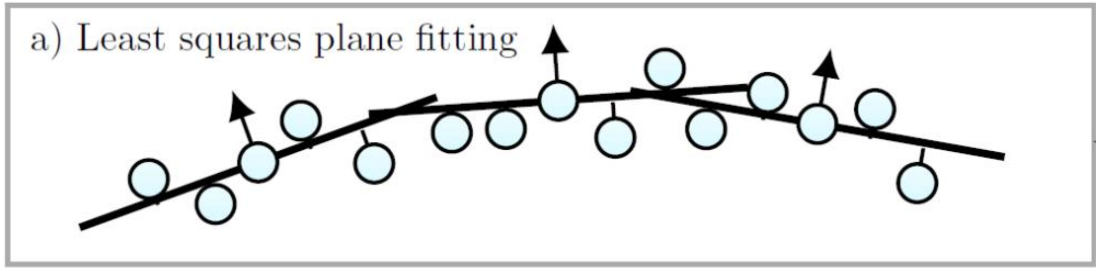
$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j\in\mathcal{N}(i)} \underbrace{||\mathcal{P}(i)_j \cdot \mathbf{n}||^2}_{\text{Squared point-plane distance}}$$



Surface Normal Estimation as Least-Squares

- Unoriented normal estimation

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j \in \mathcal{N}(i)} \underbrace{||\mathcal{P}(i)_j \cdot \mathbf{n}||^2}_{\text{Squared point-plane distance}}$$



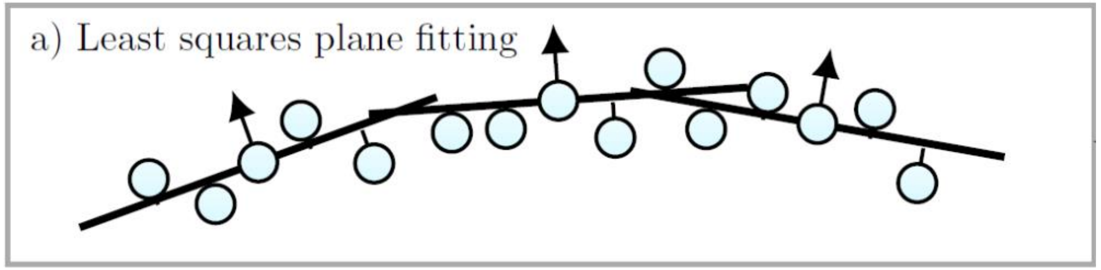
- Weighted least squares

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j \in \mathcal{N}(i)} \underbrace{w_{i,j}}_{\text{Weight for each point in each neighborhood}} ||\mathcal{P}(i)_j \cdot \mathbf{n}||^2$$

Surface Normal Estimation as Least-Squares

- Unoriented normal estimation

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j\in\mathcal{N}(i)} \underbrace{||\mathcal{P}(i)_j \cdot \mathbf{n}||^2}_{\text{Squared point-plane distance}}$$



Examples:

- Weighted least squares

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j\in\mathcal{N}(i)} \underbrace{w_{i,j}}_{\text{Weight for each point in each neighborhood}} ||\mathcal{P}(i)_j \cdot \mathbf{n}||^2$$

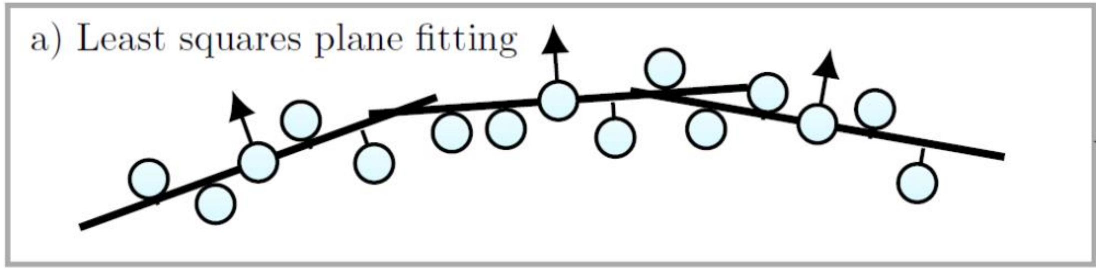
- Iterative re-weighted least squares

$$w_{i,j}^l = s(\mathbf{r}_{i,j}^{l-1})$$

Surface Normal Estimation as Least-Squares

- Unoriented normal estimation

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j \in \mathcal{N}(i)} \underbrace{||\mathcal{P}(i)_j \cdot \mathbf{n}||^2}_{\text{Squared point-plane distance}}$$



- Weighted least squares

$$\mathbf{n}_i^* = \arg\min_{\mathbf{n}:|\mathbf{n}|=1} \sum_{j \in \mathcal{N}(i)} \underbrace{w_{i,j}}_{\text{Weight for each point in each neighborhood}} ||\mathcal{P}(i)_j \cdot \mathbf{n}||^2$$

Examples:

- Iterative re-weighted least squares

$$w_{i,j}^l = s(\mathbf{r}_{i,j}^{l-1})$$

- Anisotropic Kernel

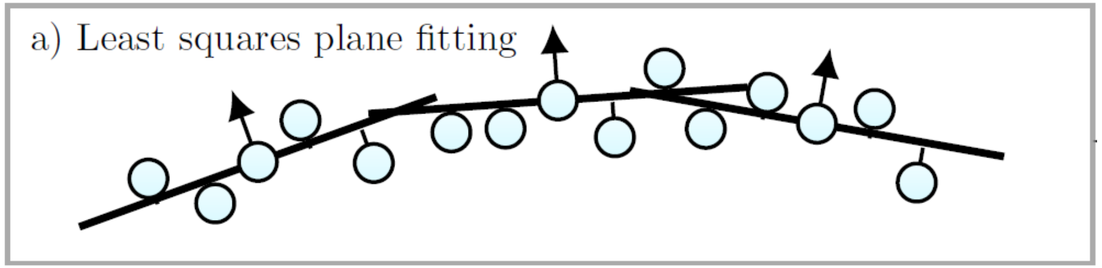
$$w_{i,j}^l = \psi(\mathbf{p}_j - \mathbf{p}_i)$$

nnaisense

technische universität
dortmund

Deep Kernel Re-Weighting

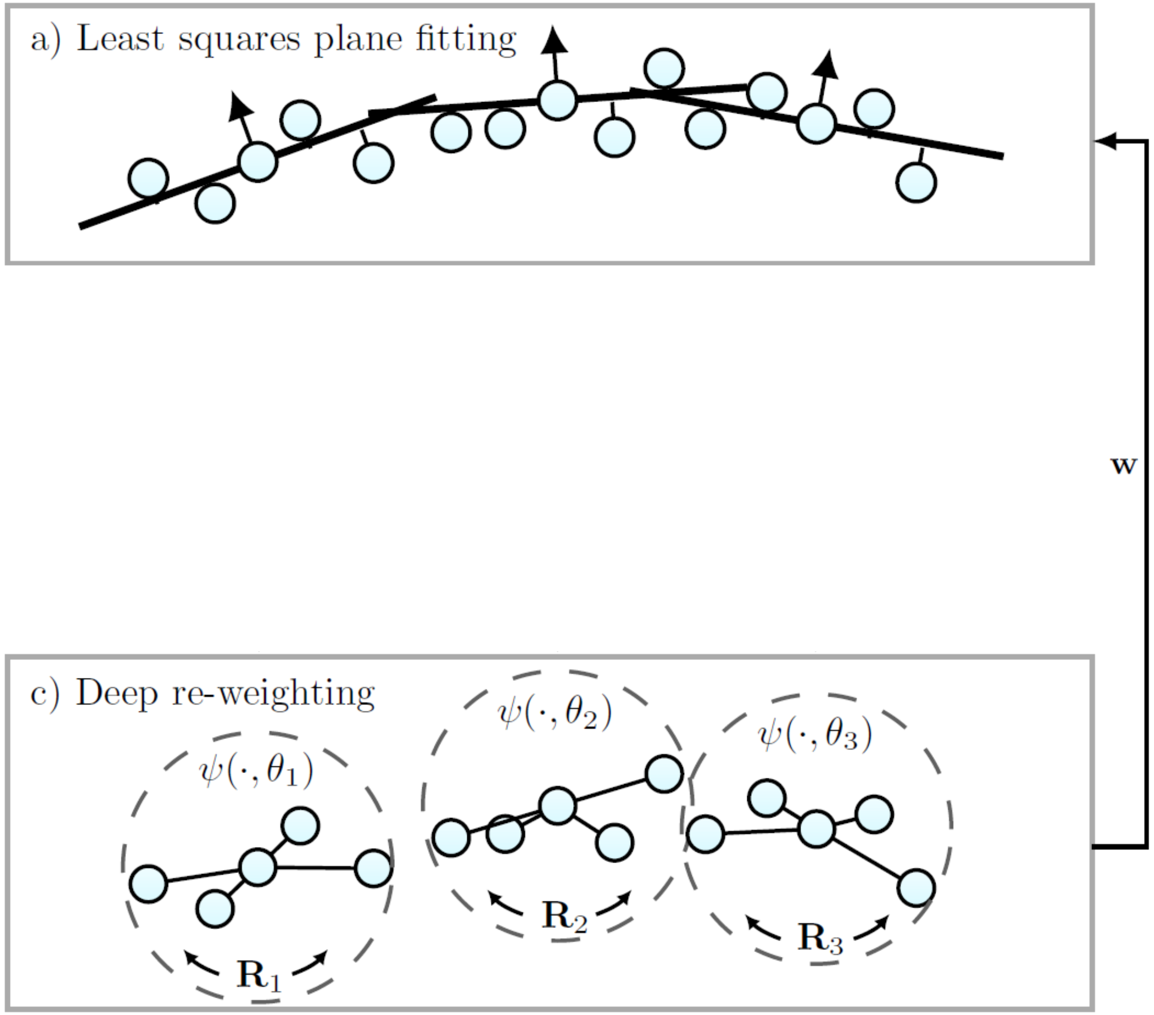
- Robustness to outliers
- Equivariance to rotation
- Recovering sharp details



Deep Kernel Re-Weighting

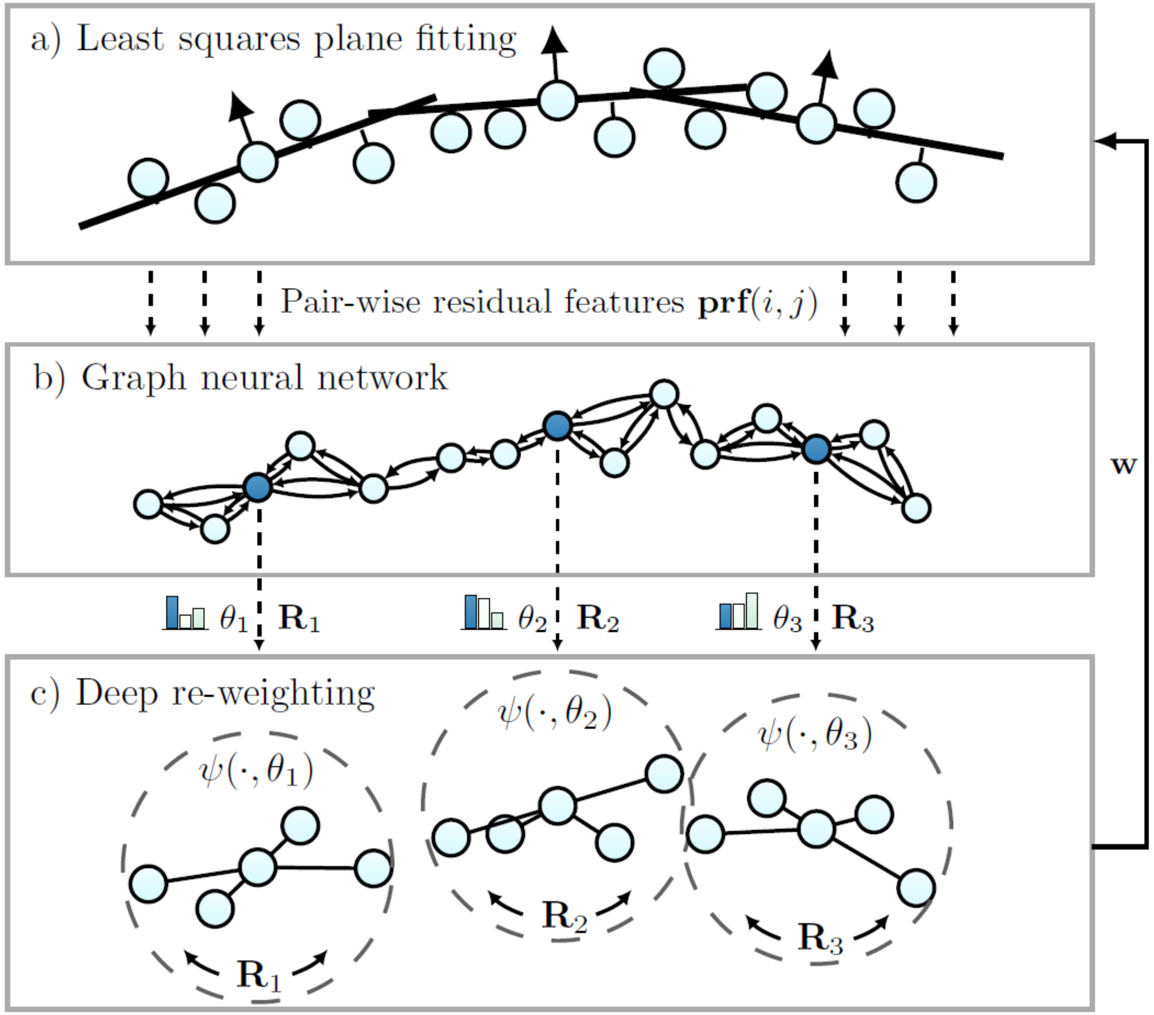
- Robustness to outliers
- Equivariance to rotation
- Recovering sharp details

$$w_{i,j} = \psi(\underbrace{\mathbf{R}_i}_{\text{Rotation matrix}}(\mathbf{p}_j - \mathbf{p}_i), \underbrace{\theta_i}_{\text{Latent kernel parameterization}})$$



GNN and Training

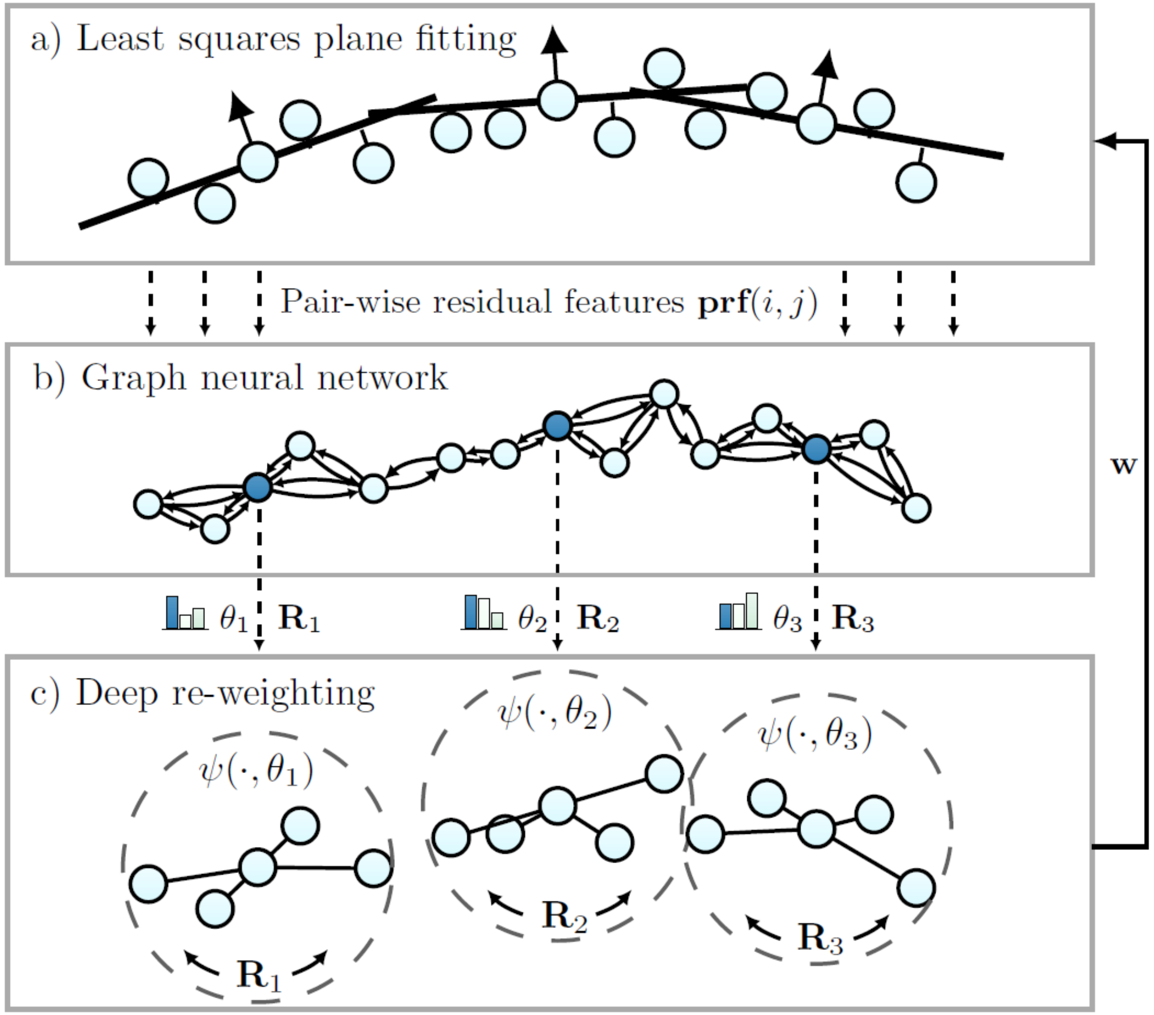
$$(\underbrace{\mathbf{p}_j - \mathbf{p}_i}_{\text{Relative point positions}}, \underbrace{\mathbf{prf}(i,j)}_{\text{Residual features}}) \longrightarrow \boxed{\text{GNN}} \longrightarrow (\mathbf{R}_i, \theta_i)$$



technische universität dortmund

GNN and Training

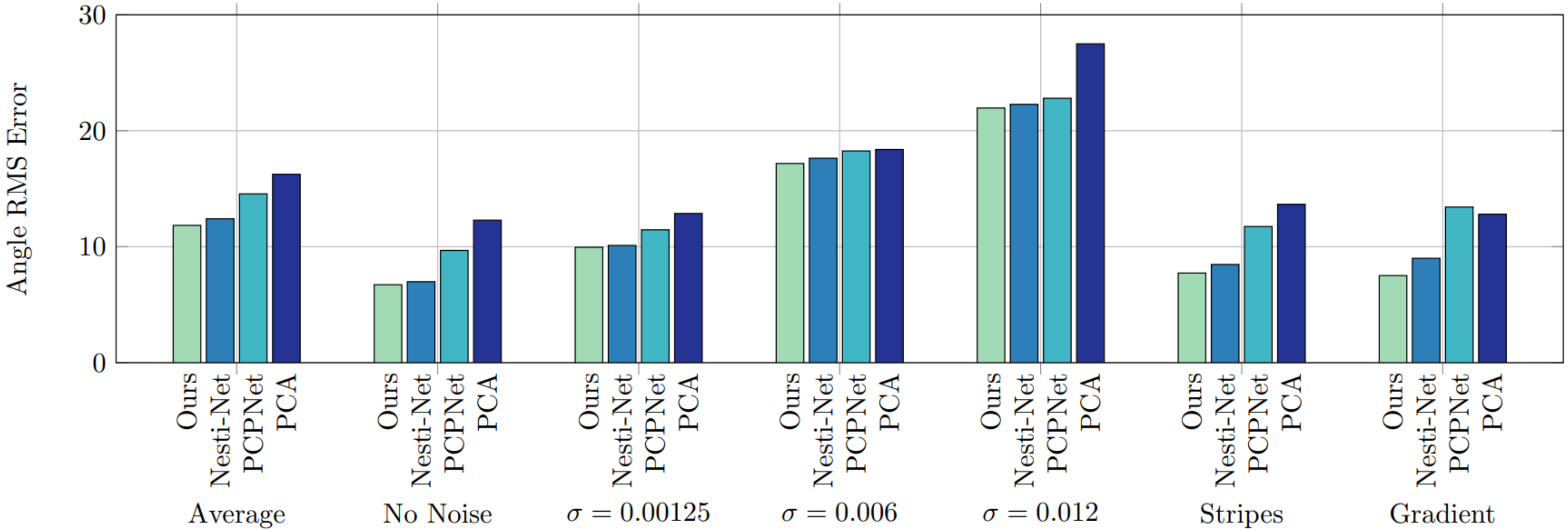
$$\Big(\underbrace{\mathbf{p}_j - \mathbf{p}_i}_{\text{Relative point positions}},\ \underbrace{\mathbf{prf}(i,j)}_{\text{Residual features}}\Big) \longrightarrow \boxed{\text{GNN}} \longrightarrow (\mathbf{R}_i, \theta_i)$$



GNN Properties:

- Permutation invariance
- Handling of varying neighborhood size
- Locality

Accuracy and Speed

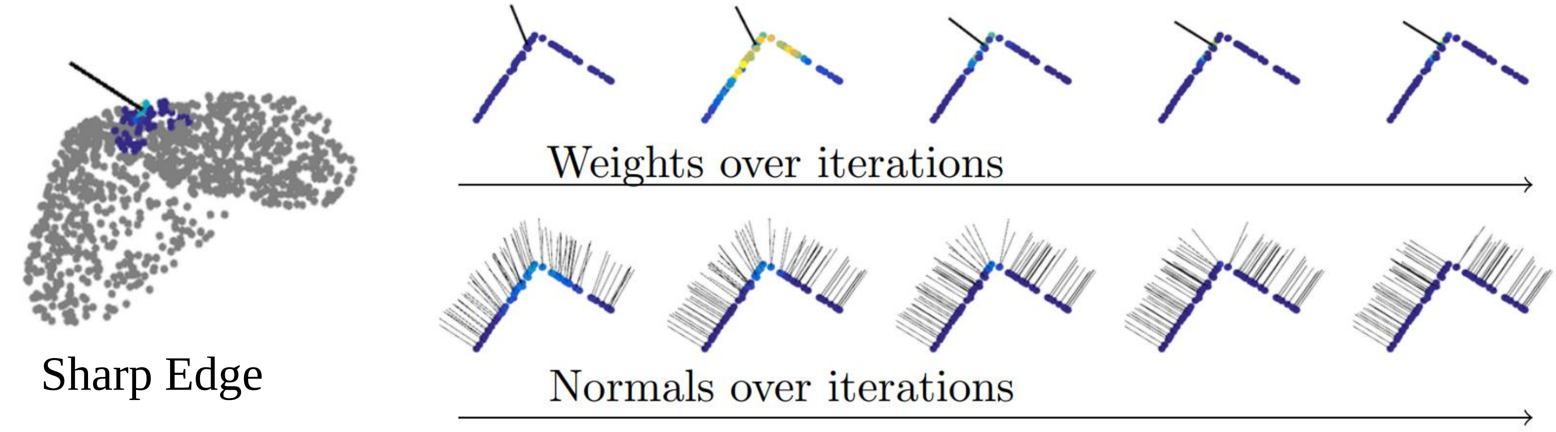


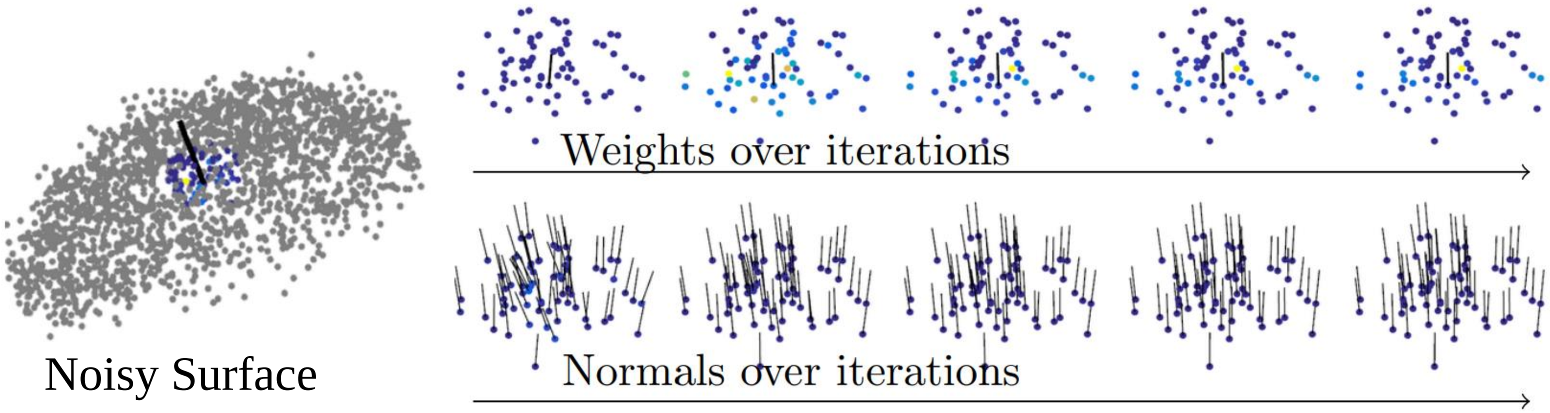
	Ours	Nesti-Net	PCPNet
Number of network parameters	**7981**	179M	22M
Execution time for 100k points	**3.57 s**	1350 s	470 s
Relative execution time	1×	**378**×	**131**×

nnaisense

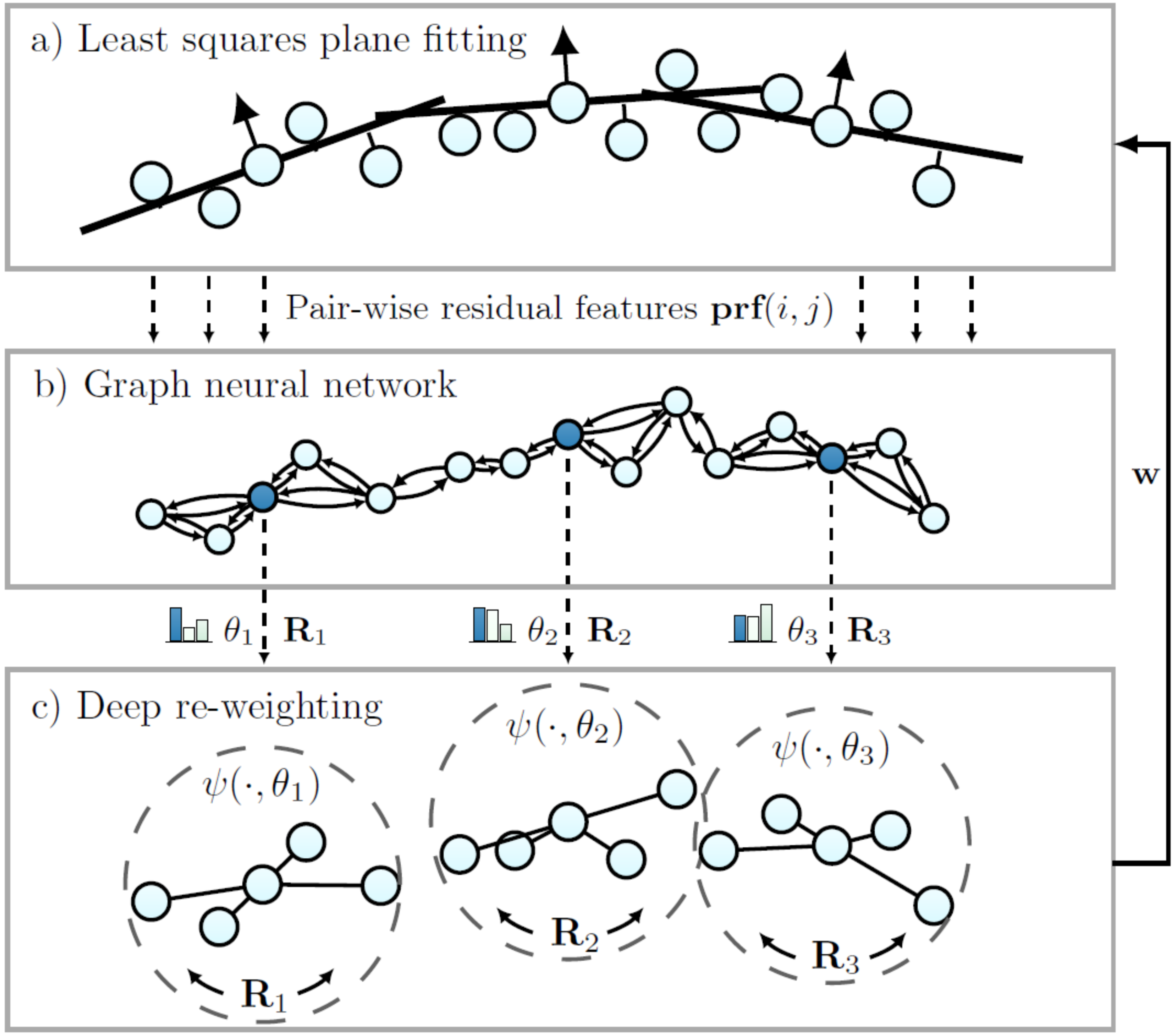
technische universität
dortmund

Interpreting the Results





Thank you and enjoy CVPR ☺



nnaisense

technische universität dortmund