JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

清華大學 Tsinghua University

Code available

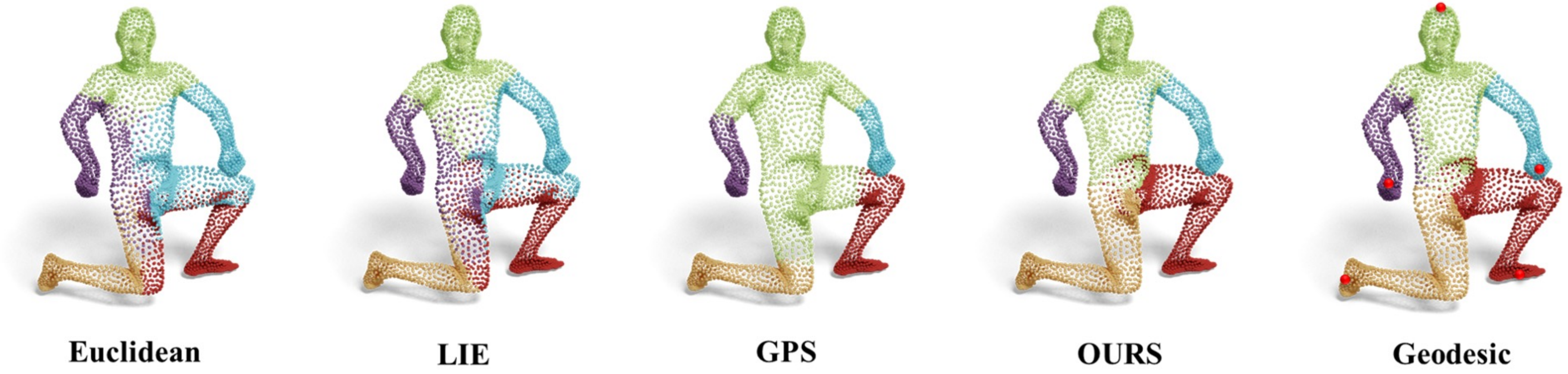
Neural Intrinsic Embedding for Non-Rigid Point Cloud Matching

Puhua Jiang[1,2], Mingze Sun[1], Ruqi Huang[1]

[1]Tsinghua Shenzhen International Graduate School [2]Peng Cheng Laboratory

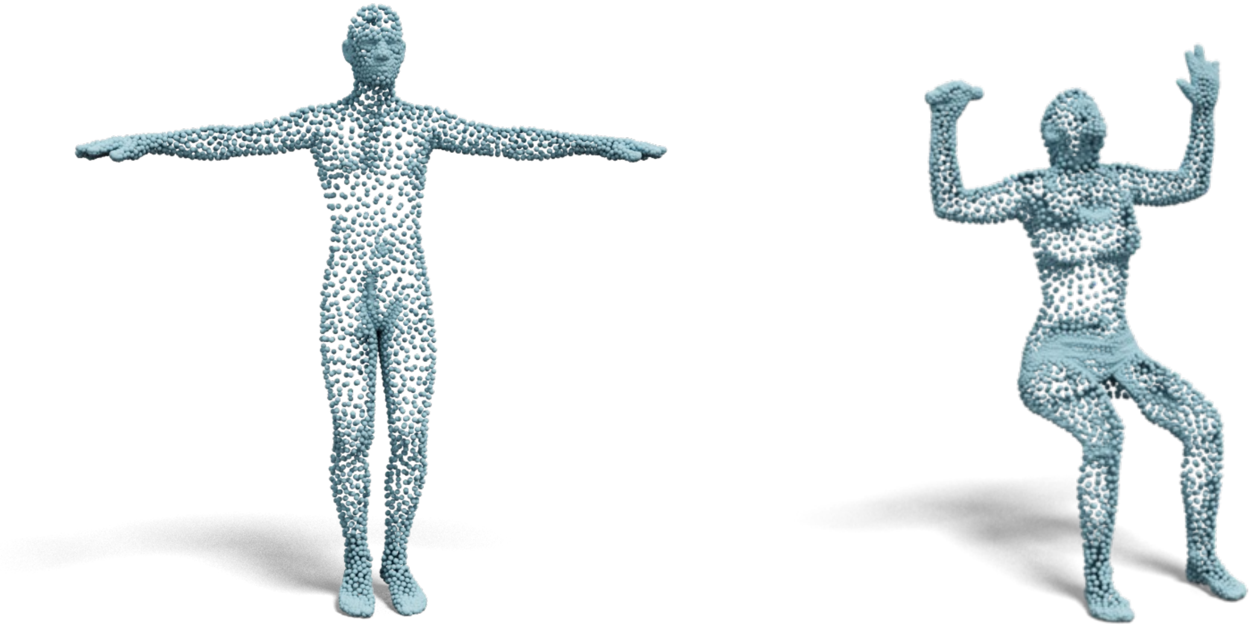
{jph21, smz22}@mails.tsinghua.edu.cn ruqihuang@sz.tsinghua.edu.cn

Poster: THU-PM-117



Overall Goal： Non-rigid point cloud matching

Given two **raw** point clouds undergoing **non-rigid** deformation



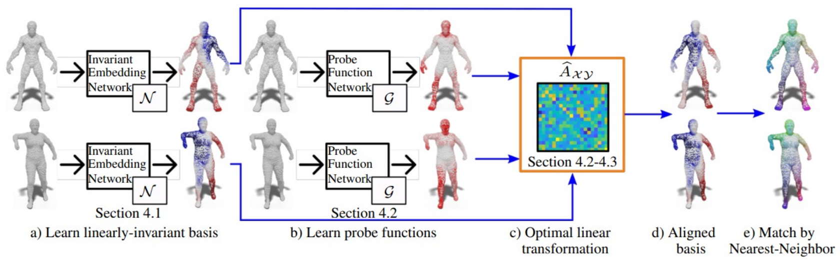
Overall Goal： Non-rigid point cloud matching

Estimate point-wise **correspondence**

Previous Methods：

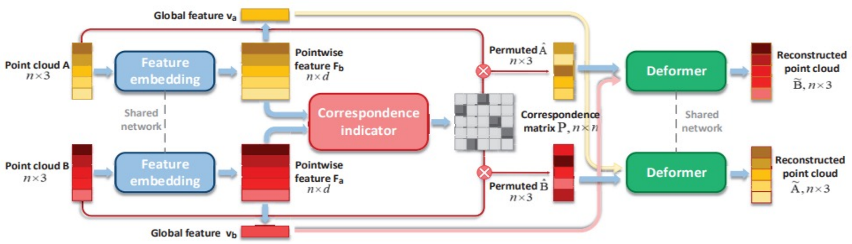
Linearly Invariant Embedding(ECCV 20)

- Require Ground truth correspondences
- Require large datasets
- Lack of generalization ability



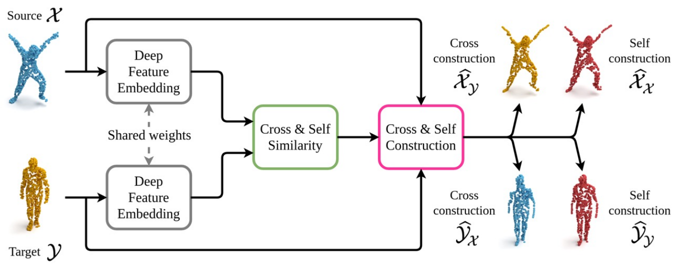
Corr-Net 3D(ICCV 21)

- Fixed point cloud size
- Require large datasets
- Can not handle artifacts



DPC(3DV 21)

- Require large datasets
- Heavily over-fitting



Challenges:

- Lack of intrinsic structure
- Varying sampling density
- Artifacts like noise, partiality…
- Lack of correspondences labels

Desiderata :

- Intrinsic geometry aware
- Computationally efficient
- Robustness
- Weak supervision

Previous Methods：

Linearly Invariant Embedding(ECCV 20)

- Require Ground truth correspondences
- Require large datasets
- Lack of generalization ability

Corr-Net 3D(ICCV 21)

- Fixed point cloud size
- Require large datasets
- Can not handle artifacts

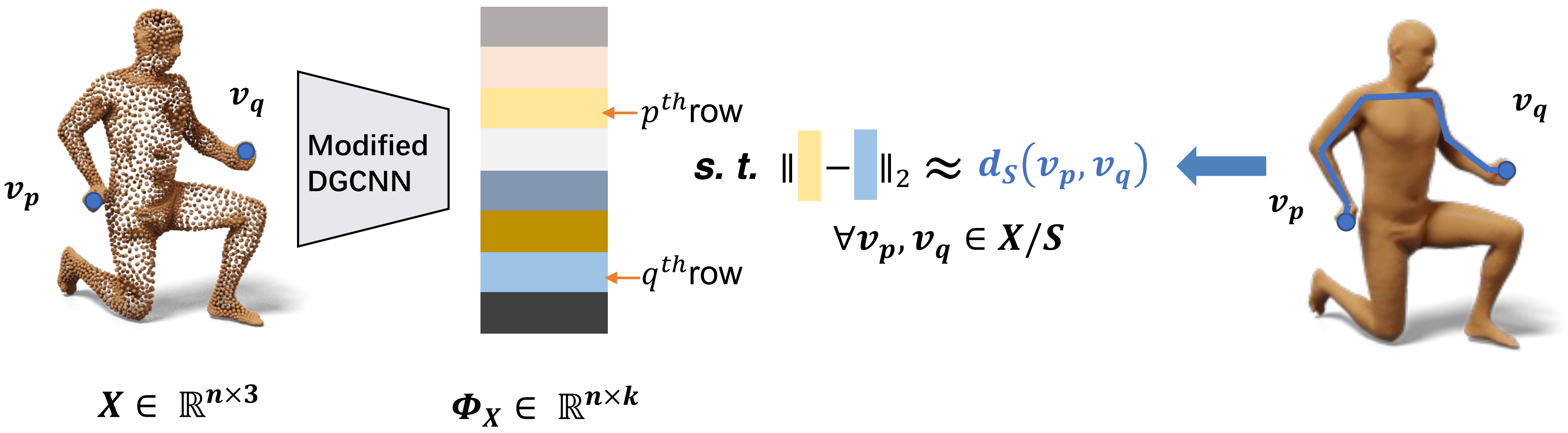
DPC(3DV 21)

- Require large datasets
- Heavily over-fitting

Solution: To develop an intrinsic geometry aware embedding scheme

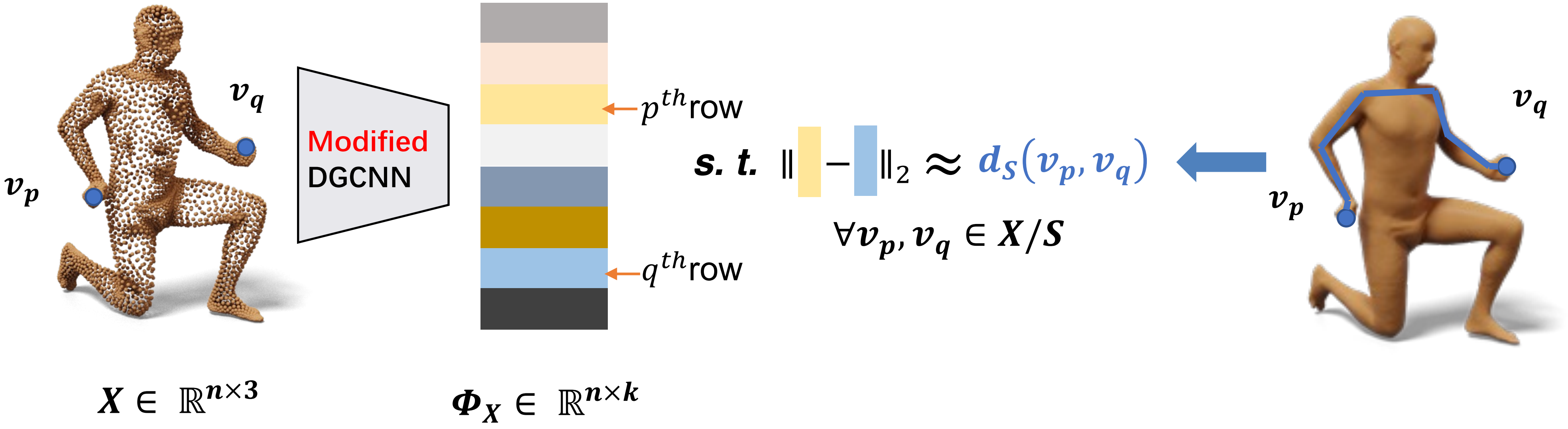
Contributions:

- Neural Intrinsic Embedding (NIE): a learning-based framework to **efficiently** embed point cloud in an **intrinsic geometry aware** way.



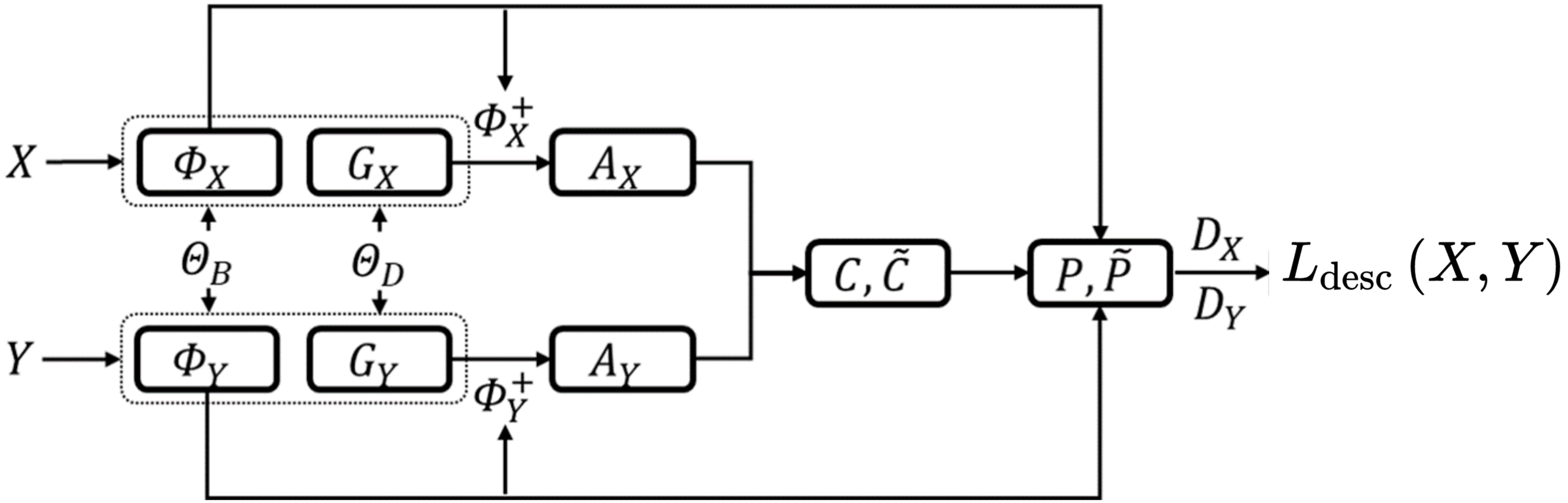
Contributions:

- Neural Intrinsic Embedding (NIE): a learning-based framework to **efficiently** embed point cloud in an **intrinsic geometry aware** way.
- A modification of DGCNN: **robust to sampling bias**.

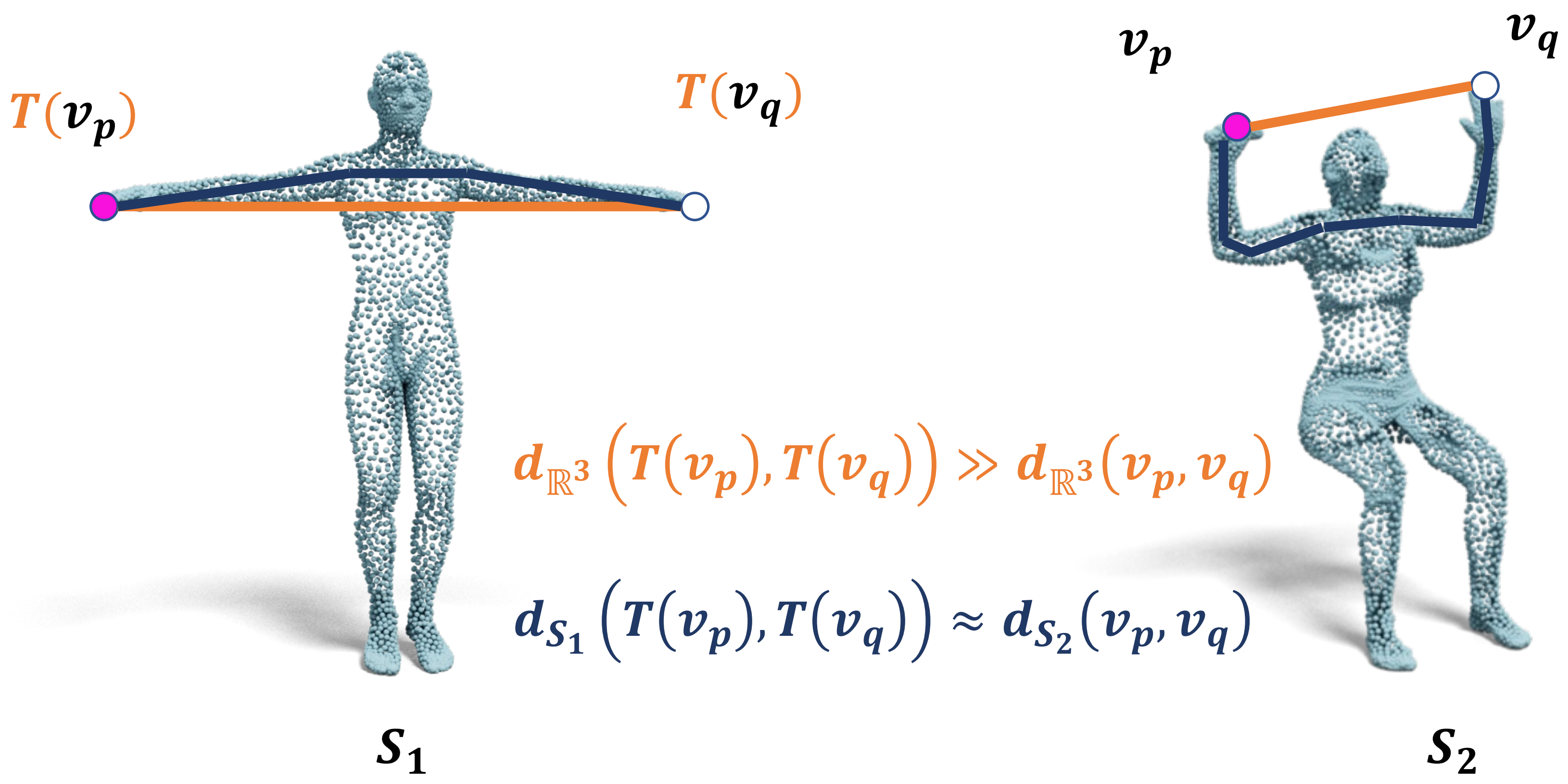


Contributions:

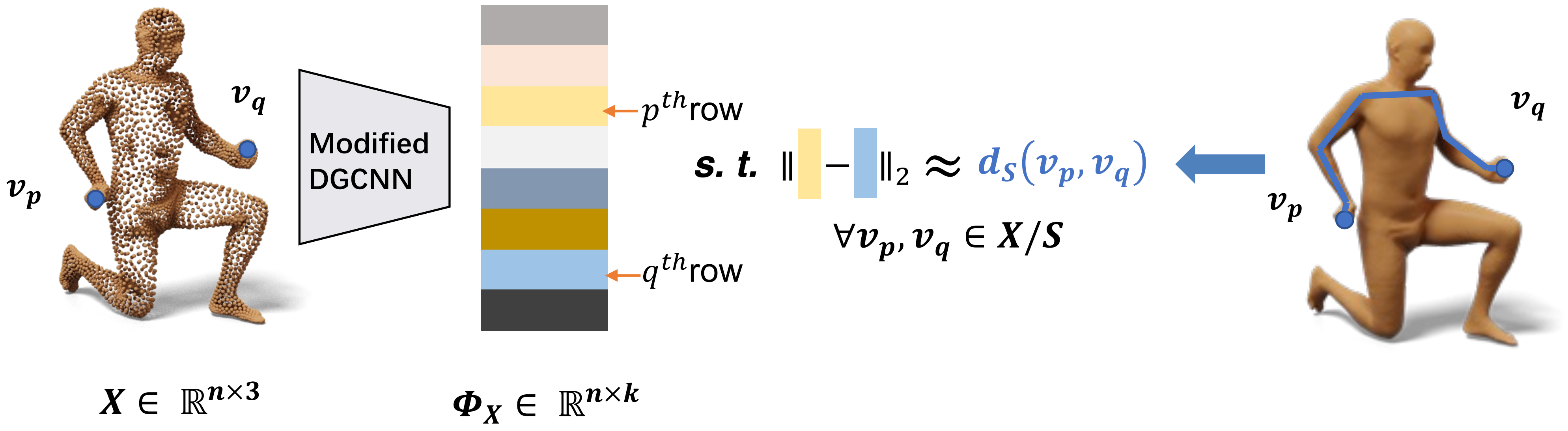
- Neural Intrinsic Embedding (NIE): a learning-based framework to **efficiently** embed point cloud in an **intrinsic geometry aware** way.
- A modification of DGCNN: **robust to sampling bias**.
- Neural Intrinsic Mapping (NIM): **a weakly supervised** network for non-rigid point cloud matching.



Intrinsic & Extrinsic



Neural Intrinsic Embedding (NIE)



Neural Intrinsic Embedding (NIE)

We design loss function to maintain the **geodesic distance** between points in NIE :

Relative Geodesic Loss

$$L_G(\Theta_B) = \sum_i \sum_{(p,q)\in S_i} \frac{\left| d_E^i(v_p, v_q) - d_S(v_p, v_q) \right|^2}{d_S(v_p, v_q)^2}$$

KL Loss

$$P_S^p(v_q) = \frac{\exp(-\alpha d_S(v_p, v_q))}{\sum_{q'} \exp(-\alpha d_S(v_p, v_{q'}))}, \forall v_q \in S_i$$

$$L_{KL}(\Theta_B) = \sum_i \sum_p KL\left(P_E^p, P_S^p\right)$$

Neural Intrinsic Embedding (NIE)

Further, we take a self-supervised approach to **avoid rank deficiency** :

Bijectivity Loss

$$C_{ab} = \Phi_i^{b\dagger} \Pi_{ba} \Phi_i^a, C_{ba} = \Phi_i^{a\dagger} \Pi_{ab} \Phi_i^b$$

$$L_B(\Theta_B) = \sum_{a,b,i} \|C_{ab}C_{ba} - I\|_F^2 + \|C_{ba}C_{ab} - I\|_F^2$$

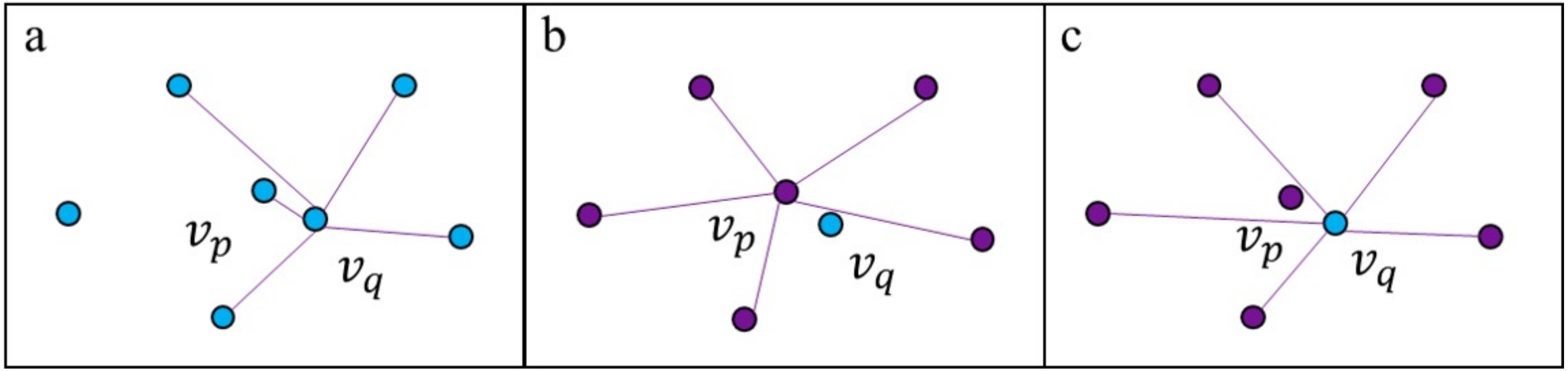
where Φ_i^a and Φ_i^b are sub-embeddings of X_i^a and X_i^b which be the first even-index and odd-index 1000 vertices from the FPS sampled vertices form X_i.

Neural Intrinsic Embedding (NIE)

Our method takes in only point cloud and produces segmentation that is **intrinsic geometry-aware**.

Euclidean LIE GPS OURS Geodesic

A modification of DGCNN

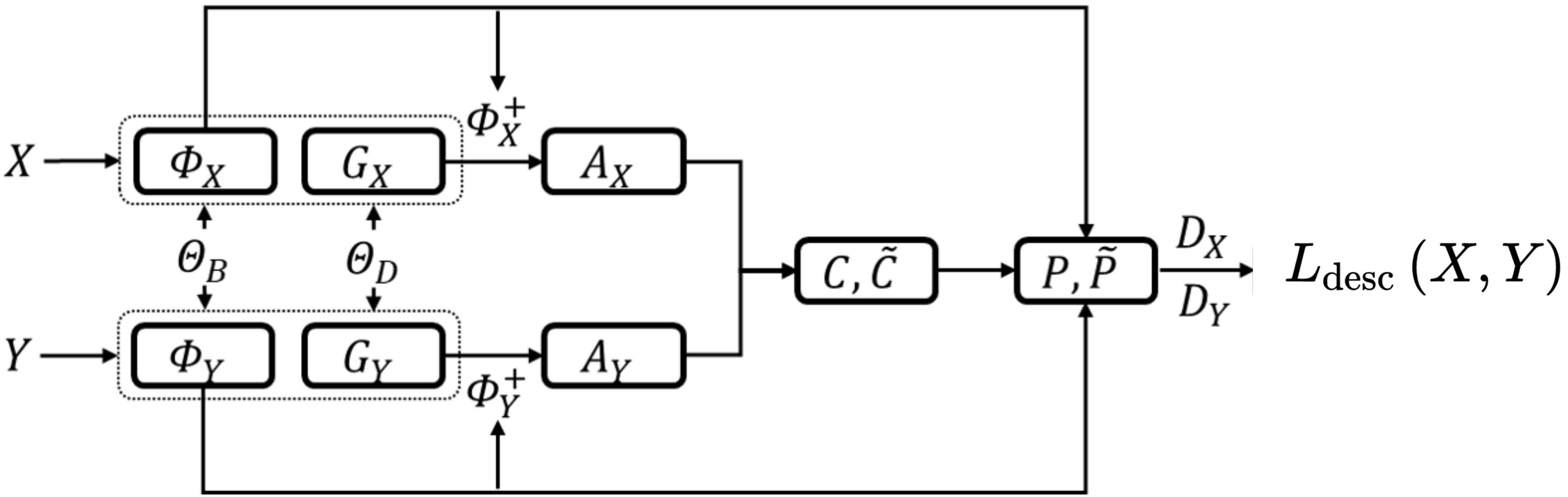


Given a point cloud X, we conduct FPS on X to obtain a evenly distributed subset X_s. Then, for a point v_q, instead of searching directly its k−NN within X, we find its nearest neighbor, v_p, in X_s, and then assign the k−NN of v_p within X_s to v_q.

Neural Intrinsic Mapping (NIM) network

We adopt the framework of [1] with the following changes:

- Replace the eigenbasis with our trained neural intrinsic embedding(NIE).



[1] Litany, O., Remez, T., Rodola, E., Bronstein, A., & Bronstein, M. (2017). Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision* (pp. 5659-5667).

Neural Intrinsic Mapping (NIM) network

- Leverage NIE's intrinsic information to introduce a self-supervised loss[2]:

$$L_{\text{desc}}(X,Y) = L_{\text{isometric}}(X,Y) + L_{\text{cyclic}}(X,Y)$$

Cyclic Loss

$$L_{\text{cyclic}}(X,Y) = \frac{1}{|X|^2}\left\|\left(D_X - (\tilde{P}P)D_X(\tilde{P}P)^T\right)\right\|_F^2 + \frac{1}{|Y|^2}\left\|\left(D_Y - (P\tilde{P})D_Y(P\tilde{P})^T\right)\right\|_F^2$$

Isometric Loss

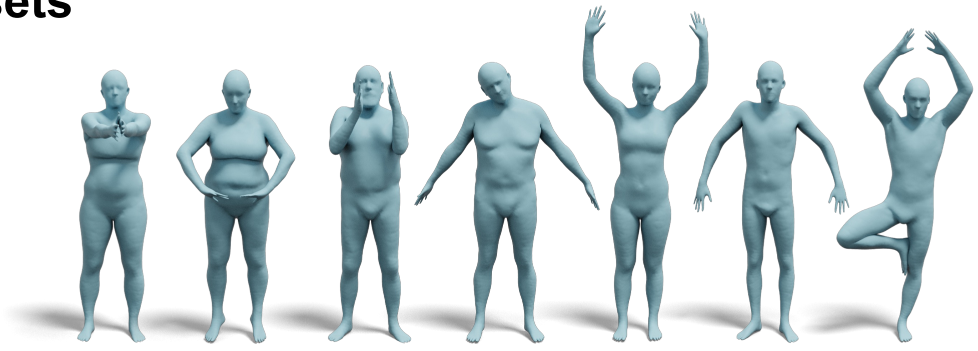
$$L_{\text{isometric}}(X,Y) = \frac{1}{|X|^2}\left\|\left(D_X - \tilde{P}D_Y\tilde{P}^T\right)\right\|_F^2 + \frac{1}{|Y|^2}\left\|\left(D_Y - PD_XP^T\right)\right\|_F^2$$

where D_X, D_Y are the geodesic distance matrix regarding X and Y , respectively.

[2] Ginzburg, D., & Raviv, D. (2020). Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16* (pp. 36-52). Springer International Publishing.

Experimental Results

Datasets

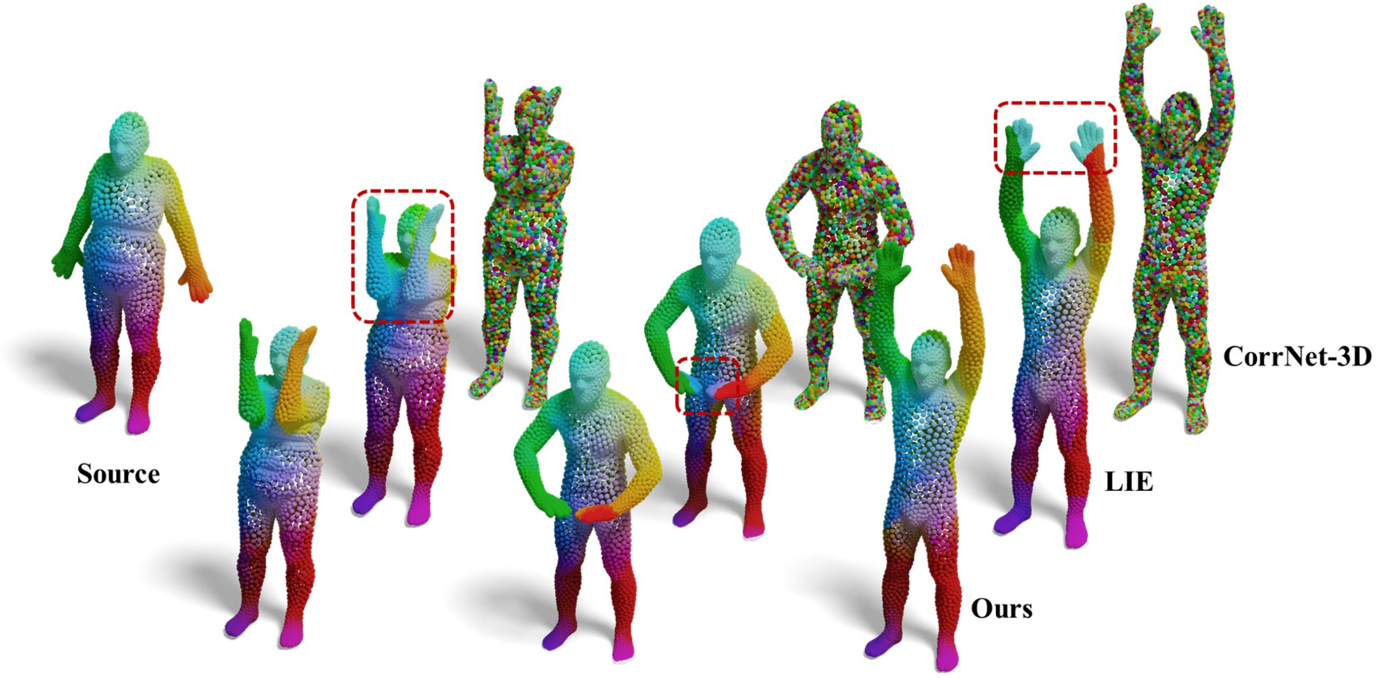


(a) FAUST_r

(b) SCAPE_r

Quantitative and Qualitative results

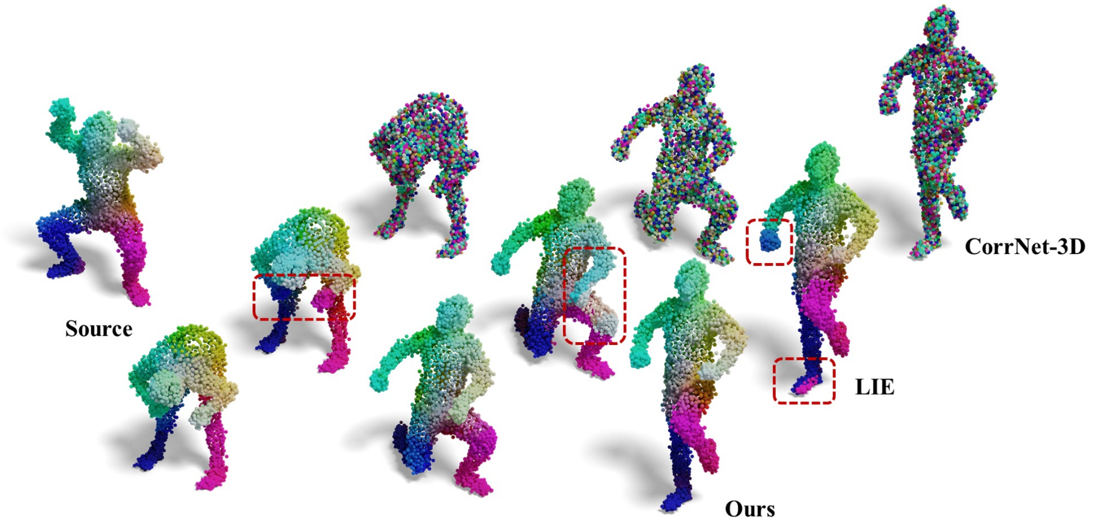
Method	F	S	F on S	S on F
BCICP [37]	15.	16.	\	\
SURFMNet(U) [39]	15.	12.	32.	32.
UnsupFMNet(U) [17]	10.	16.	29.	22.
NeuroMorph(U) [13]	8.5	30.	29.	18.
FMNet(S) [27]	11.	12.	30.	33.
WSupFMNet(W) [41]	**3.3**	**7.3**	**12.**	**6.2**
NIM w/ LBO basis 20	5.8	13.	22.	16.
3D-CODED(S) [16]	**2.5**	31.	31.	33.
CorrNet-3D(U) [56]	63.	58.	58.	63.
LIE(S) [28]	3.6	12.	19.	12.
Ours(W)	5.5	**11.**	**15.**	**8.7**



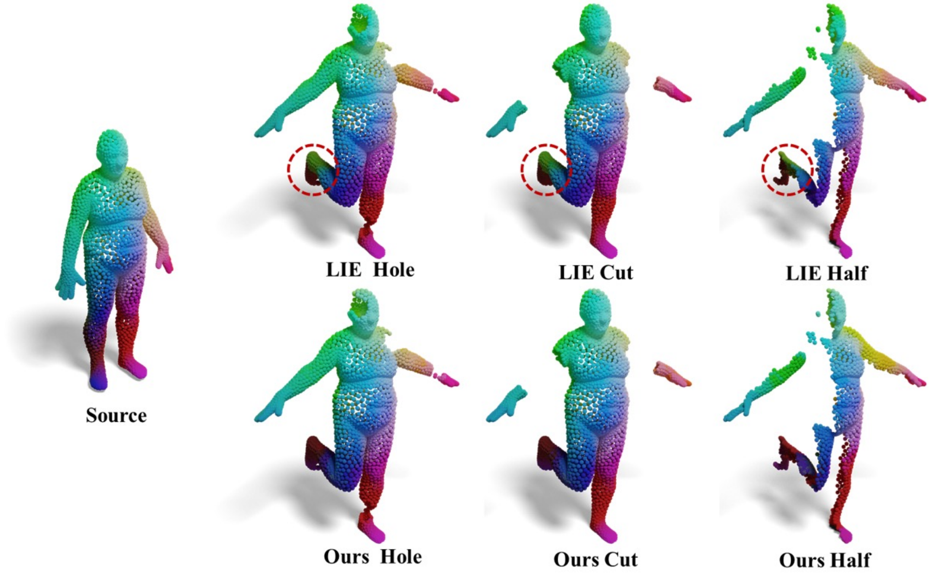
Experimental Results

Robustness to noise and various partiality

Method	S	F
CorrNet-3D [56]	52.	54.
LIE [28]	20.	15.
Ours	**10.**	**6.5**
CorrNet-3D Noise	58.	62.
LIE Noise	20.	15.
Ours Noise	**11.**	**7.2**



Method	half	hole	cut
LIE [28]	15.	15.	16.
Ours	**10.**	**7.0**	**7.2**



Experimental Results

Ablation study of training loss terms

Method	OPT	Geo.Err	Mat.Err
L_G	4.4	**8.8**	13.2
$L_G + L_B$	3.5	12.4	11.8
$L_G + L_B + L_{KL}$	3.3	10.6	11.5
Full model with sample	**3.1**	9.5	**11.0**

Method	Training	Surreal	
	Testing	S	F
Ours w/o sample		12.	8.1
Ours w sample		**10.**	**6.5**

Thanks for your time!