RINO: Rotation-Invariant Non-Rigid Correspondences

Supplementary Material

8. RINONet: Discussion and Proof

VN-Gradient Layer. Following the notation in [54], we introduce another formulation of our equivariant gradient layer. We first compute the spatial gradient $\mathbf{z} \in \mathbb{C}^{n\times 3}$ of the diffused feature $\mathbf{h} \in \mathbb{R}^{n\times 3}$ as $\mathbf{z} = \mathcal{G}(\mathbf{h})$, where $\mathcal{G}$ is the intrinsic spatial gradient operator defined on the shape surface. Then at each (single) vertex v, the local gradients of all feature channels are stacked to form $\mathbf{w} \in \mathbb{C}^{c\times 3}$ and obtain real-valued features $\mathbf{f} \in \mathbb{R}^c$ as:

$$\mathbf{f} \coloneqq \mathrm{Re}(\mathrm{diag}(\overline{\mathbf{w}}(\mathbf{A}\mathbf{w})^T)) \tag{6}$$

where $\mathbf{A}$ is a learnable square $\mathbb{C}^{c\times c}$ matrix, $\mathrm{diag}(\cdot)$ extracts the diagonal of an input (square) matrix and taking the real part by $\mathrm{Re}(\cdot)$ after a complex conjugate $\overline{\mathbf{w}}$ is a notational convenience for dot products between a pair of 2D vectors. The i^{th} entry of the output at vertex v is given by the inner product $\mathbf{f}^i = \mathrm{Re}(\sum_{j=1}^{c}\langle\overline{\mathbf{w}}^i, \mathbf{A}_{ij}\mathbf{w}^j\rangle)$ with vectors $\overline{\mathbf{w}}^i, \mathbf{w}^j \in \mathbb{C}^{1\times 3}$.

Note that Eq. 6 is the same as Eq. 3 in the main paper by using a more compact notation, which will help to prove Theorem 1 as shown below.

Proof of Theorem 1.

Proof. Given a rotation matrix $\mathbf{R} \in SO(3)$ applied on the input $\mathbf{w}$, we leverage the property of rotation group and linear algebra tricks.

$$\mathbf{f}^i(\mathbf{w}\mathbf{R}) = \mathrm{Re}(\sum_{j=1}^{c}\langle\overline{\mathbf{w}^i\mathbf{R}}, \mathbf{A}_{ij}(\mathbf{w}^j\mathbf{R})\rangle) \tag{7}$$

$$= \mathrm{Re}(\sum_{j=1}^{c}\langle\overline{\mathbf{w}}^i\mathbf{R}, \mathbf{A}_{ij}\mathbf{w}^j\mathbf{R}\rangle) \tag{8}$$

$$= \mathrm{Re}(\sum_{j=1}^{c}\langle\overline{\mathbf{w}}^i\mathbf{R}\mathbf{R}^T, \mathbf{A}_{ij}\mathbf{w}^j\rangle) \tag{9}$$

$$= \mathrm{Re}(\sum_{j=1}^{c}\langle\overline{\mathbf{w}}^i, \mathbf{A}_{ij}\mathbf{w}^j\rangle) \tag{10}$$

$$= \mathbf{f}^i(\mathbf{w}) \tag{11}$$

Hence, Eq. 6 and $\mathbf{f}$ is SO(3)-invariant. □

Note that applying $\tanh$ activation on invariant features does not change the invariance. Therefore, $\mathbf{g} \coloneqq \tanh(\mathbf{f})$ is also SO(3)-invariant.

Proof of Theorem 2.

Proof. We first prove the equivariance of the RINONet block, which boils down to show that Eq. 4 is equivariant. Since VN-MLP is equivariant [19], it remains to show that the network input $[\mathbf{u}, \mathbf{h}, \mathbf{e}]$ is equivariant. This is a direct consequence of the equivariance of heat diffusion $\mathcal{H}_t(\cdot)$ and Theorem 1. More specifically, if the input feature $\mathbf{u}$ is rotated by $\mathbf{R} \in SO(3)$, the corresponding diffusion feature $\mathbf{h}$ and gradient feature $\mathbf{e}$ will become $\mathbf{h}\mathbf{R}$ and $\mathbf{e}\mathbf{R}$ respectively. Hence their concatenation leads to $[\mathbf{u}\mathbf{R}, \mathbf{h}\mathbf{R}, \mathbf{e}\mathbf{R}] = [\mathbf{u}, \mathbf{h}, \mathbf{e}]\mathbf{R}$. Therefore Eq. 4 is equivariant.

Based on the result above, we now prove the invariance of the whole RINONet. As illustrated in Fig. 4, the RINONet is a linear composition of a chain of equivariant layers, followed by a VN-invariant layer and VN-linear layer. It is trivial to observe that:

- Composition of equivariance functions keeps the equivariance.
- Composition of equivariant function with invariant function leads to invariance.

Hence, the whole RINONet is SO(3)-invariant. □

9. Unsupervised SO(3)-Equivariant Matching

9.1. Unsupervised Loss

Our unsupervised loss $L_{\text{total}} = L_{\text{struct}} + L_{\text{couple}} + L_{\text{contr}}$ guides the RINONet to learn features that are aware of the surface geometry and orientation, while being robust to shape artifacts such as partiality, non-manifoldness, and noise perturbation. Below, we provide the full formulation for each component.

Structural Loss. The structural loss ensures that the estimated functional maps exhibit desirable geometric properties. Following prior works [20, 50], we enforce near-orthogonality and bijectivity for the standard functional map $\mathbf{C}$ and orthogonality for the complex functional map $\mathbf{Q}$. Although the orthogonality term penalizes deviation of both $\mathbf{C}$ and $\mathbf{Q}$ from being orthogonal, it corresponds to enforcing different properties: while the orthogonality of $\mathbf{C}$ encourages local area-preservation, it is a necessary property for $\mathbf{Q}$ to represent a valid push-forward [21, 22].

$$L_{\text{orth}}(\mathbf{C}_{\mathcal{X}\mathcal{Y}}) = \|\mathbf{C}_{\mathcal{X}\mathcal{Y}}^T\mathbf{C}_{\mathcal{X}\mathcal{Y}} - \mathbf{I}\| \tag{12}$$

$$L_{\text{orth}}(\mathbf{Q}_{\mathcal{X}\mathcal{Y}}) = \|\mathbf{Q}_{\mathcal{X}\mathcal{Y}}^T\mathbf{Q}_{\mathcal{X}\mathcal{Y}} - \mathbf{I}\| \tag{13}$$

The bijectivity term enforces that the map from $\mathcal{X}$ to $\mathcal{Y}$ and back to $\mathcal{X}$ is the identity map, promoting cycle-consistency.

$$L_{\text{bij}}(\mathbf{C}_{\mathcal{XY}}, \mathbf{C}_{\mathcal{YX}}) = \|\mathbf{C}_{\mathcal{XY}}\mathbf{C}_{\mathcal{YX}} - \mathbf{I}\| \tag{14}$$

Our total structural loss, applied bi-directionally, is:

$$\begin{aligned} L_{\text{struct}} =& \lambda_1 L_{\text{orth}}(\mathbf{C}_{\mathcal{XY}}) + \lambda_1 L_{\text{orth}}(\mathbf{C}_{\mathcal{YX}}) \\ &+ \lambda_2 L_{\text{orth}}(\mathbf{Q}_{\mathcal{XY}}) + \lambda_2 L_{\text{orth}}(\mathbf{Q}_{\mathcal{YX}}) \\ &+ \lambda_3 L_{\text{bij}}(\mathbf{C}_{\mathcal{XY}}, \mathbf{C}_{\mathcal{YX}}) + \lambda_3 L_{\text{bij}}(\mathbf{C}_{\mathcal{YX}}, \mathbf{C}_{\mathcal{XY}}) \end{aligned} \tag{15}$$

where λ_i are Langragian weights.

Coupling Loss. The coupling loss ensures mutual consistency between the three map representations derived from the invariant features: the soft pointwise map $\mathbf{\Pi}$, the functional map $\mathbf{C}$, and the complex functional map $\mathbf{Q}$. Similar to previous work [3, 11, 49, 57], we couple the soft pointwise map $\mathbf{\Pi}_{\mathcal{YX}}$ with $\mathbf{C}_{\mathcal{XY}}$ by converting the pointwise map to its corresponding functional map $\hat{\mathbf{C}}_{\mathcal{XY}} = \mathbf{\Phi}_{\mathcal{Y}}^{\dagger}\mathbf{\Pi}_{\mathcal{YX}}\mathbf{\Phi}_{\mathcal{X}}$:

$$L_{\mathbf{\Pi C}} = \|\mathbf{C}_{\mathcal{XY}} - \hat{\mathbf{C}}_{\mathcal{XY}}\|_F^2 \tag{16}$$

Additionally, we propose to couple $\mathbf{\Pi}_{\mathcal{YX}}$ with the orientation-aware complex functional map $\mathbf{Q}_{\mathcal{XY}}$. This is achieved by first converting $\mathbf{\Pi}_{\mathcal{YX}}$ to $\hat{\mathbf{C}}_{\mathcal{XY}}$ and then converting $\hat{\mathbf{C}}_{\mathcal{XY}}$ to its complex functional map equivalent, $\hat{\mathbf{Q}}_{\mathcal{XY}}$, which can be done in closed-form using singular value decomposition (SVD) [21, Appendix F]:

$$L_{\mathbf{\Pi Q}} = \|\mathbf{Q}_{\mathcal{XY}} - \hat{\mathbf{Q}}_{\mathcal{XY}}\|_F^2 \tag{17}$$

The total coupling loss is:

$$L_{\text{couple}} = \lambda_4(L_{\mathbf{\Pi C}}^{\mathcal{XY}} + L_{\mathbf{\Pi C}}^{\mathcal{YX}}) + \lambda_5(L_{\mathbf{\Pi Q}}^{\mathcal{XY}} + L_{\mathbf{\Pi Q}}^{\mathcal{YX}}) \tag{18}$$

We empirically found that directly coupling $\mathbf{C}$ and $\mathbf{Q}$ jeopardizes training convergence, especially during the early stages of training, when feature quality is still low, resulting in inferior performance (Tab. 8). Our strategy of coupling through the more relaxed pointwise map provides a more stable regularization.

Contrastive Loss. This term encourages the learned invariant features to be distinct across the surface by ensuring bijective functional self-maps, $\hat{\mathbf{C}}_{\mathcal{XX}}$ and $\hat{\mathbf{C}}_{\mathcal{YY}}$.

$$L_{\text{contr}} = \lambda_6 \left(\|\hat{\mathbf{C}}_{\mathcal{XX}} - \mathbf{I}\|_F^2 + \|\hat{\mathbf{C}}_{\mathcal{YY}} - \mathbf{I}\|_F^2 \right)$$

where $\hat{\mathbf{C}}_{\mathcal{XX}} = \mathbf{\Phi}_{\mathcal{X}}^{\dagger}\mathbf{\Pi}_{\mathcal{XX}}\mathbf{\Phi}_{\mathcal{X}}$ and $\mathbf{\Pi}_{\mathcal{XX}} = \text{Softmax}(\mathbf{F}_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}^T/\tau)$ [13], and similar for shape $\mathcal{Y}$.

9.2. Orientation-Preserving Isometry

The functional map $\mathbf{C}$ is orientation-preserving, analogous to [21]. Here, we provide the full proof for the ease of the reader.

Theorem 3. *Let the functional map $\mathbf{C}$ (possibly infinite-dimensional) estimated from features be an isometry and $\mathbf{Q}$ the complex functional map estimated by the gradient of the features. Then the map $\mathbf{C}$ is an orientation-preserving isometry.*

Proof. Idea: we aim to prove that $\mathbf{C}$ and $\mathbf{Q}$ satisfy the following equation (cf. Eq.10 in [22]):

$$\langle \mathbf{X}, \nabla_{\mathcal{X}}(f \circ \varphi)\rangle_{T_p\mathcal{X}} = \langle \mathbf{Q}(\mathbf{X}), \nabla_{\mathcal{Y}} f\rangle_{T_{\varphi(p)}\mathcal{Y}} \tag{19}$$

where $\varphi : \mathcal{X} \to \mathcal{Y}$ is the isometric pointwise map, and $\mathbf{C} : \mathcal{L}^2(\mathcal{Y}) \to \mathcal{L}^2(\mathcal{X})$ is the corresponding pullback, which transfers $\text{SO}(3)$-invariant features, namely $\mathbf{C}(\mathbf{F}_{\mathcal{Y}}) = \mathbf{F}_{\mathcal{X}}$. Furthermore, the complex functional map, also known as the pushforward $\mathbf{Q} : T\mathcal{X} \to T\mathcal{Y}$ transfers the gradient of the invariant features $\mathbf{Q}(\nabla_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}) = \nabla_{\mathcal{Y}}\mathbf{F}_{\mathcal{Y}}$ and is complex-linear. Given an arbitrary feature function $f : \mathcal{Y} \to \mathbb{R}$, a tangent vector field $\mathbf{X} \in T\mathcal{X}$ and a point $p \in \mathcal{X}$, we have the following equalities:

$$\begin{aligned} &\langle \mathbf{Q}(\nabla_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}), \nabla_{\mathcal{Y}} f\rangle_{T\mathcal{Y}} = \langle \nabla_{\mathcal{Y}}\mathbf{F}_{\mathcal{Y}}, \nabla_{\mathcal{Y}} f\rangle_{T\mathcal{Y}} && (20)\\ &= \langle d\varphi^{-1}(\nabla_{\mathcal{Y}}\mathbf{F}_{\mathcal{Y}}), d\varphi^{-1}(\nabla_{\mathcal{Y}} f)\rangle_{T\mathcal{X}} && (21)\\ &= \langle \nabla_{\mathcal{X}}(\mathbf{C}\mathbf{F}_{\mathcal{Y}}), \nabla_{\mathcal{X}}(\mathbf{C}f)\rangle_{T\mathcal{X}} && (22)\\ &= \langle \nabla_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}, \nabla_{\mathcal{X}}(\mathbf{C}f)\rangle_{T\mathcal{X}} && (23) \end{aligned}$$

The first equality follows by applying the definition $\mathbf{Q}(\nabla_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}) = \nabla_{\mathcal{Y}}\mathbf{F}_{\mathcal{Y}}$. The second equality holds since the metric is preserved by the pullback $d\varphi^{-1}$. (This assumes the invertibility of φ, which means exact correspondences.) The third equality utilizes the fact that the pullback commutes with the gradient. And the final equality follows due to the definition $\mathbf{C}(\mathbf{F}_{\mathcal{Y}}) = \mathbf{F}_{\mathcal{X}}$.

Let $\mathbf{X} := \nabla_{\mathcal{X}}\mathbf{F}_{\mathcal{X}}$, we arrive at the Eq. 19 that we initially want to prove. According to Theorem 2 in [22], the functional map $\mathbf{C}$ is an orientation-preserving isometry. □

In particular, this theorem also holds for invariant (resp. equivariant) features estimated by RINONet.

10. Experimental Details

Implementation details. Our RINONet takes the xyz coordinate of the mass-centered 3D shape as input. To facilitate a fair comparison, we chose the internal feature dimension in our RINONet block as $c = 42$ such that the flattened feature has $126 (= 42 \times 3)$ dimensions, which is similar to the original DiffusionNet, whose hidden feature dimension is 128. Note that our network has only $327,714$ training

	Ours	URSSM	DUOFM	CnsFM	SMS	SmpFM	HbrFM
#params(k, ↓)	**327.7**	510.3	477.8	477.8	1,266.2	1,848.4	494.3
runtime(s, ↓)	0.065	**0.023**	0.071	0.055	15.311	0.171	0.043
mGeoErr (↓)	**4.6**	24.8	25.2	9.1	57.6	9.9	26.4

Table 5. **Comparison of runtime, #params and mGeoErr on SMAL.**

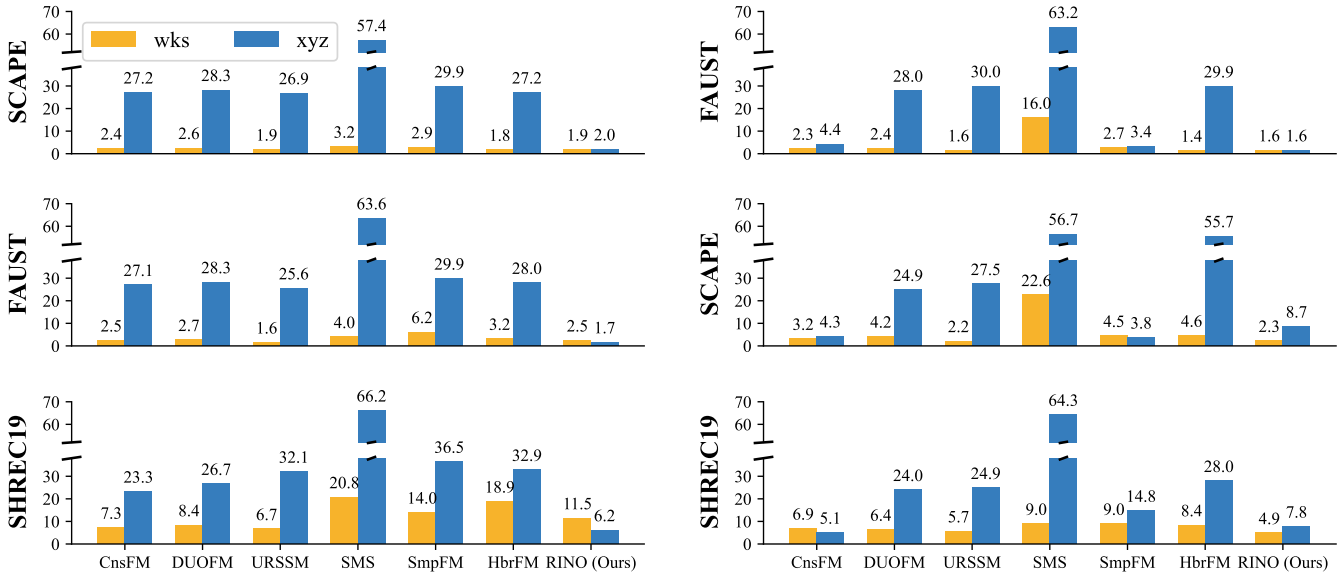

Figure 12. **MGeoErr(↓) on SCAPE, FAUST, SCHREC19.** (Left) trained on SCAPE; (Right) trained on FAUST. Y-axis dataset labels indicate the test dataset. MGeoErr is annotated on top of each bar. All baselines significantly deteriorate when changing from wks to xyz, whereas ours is more robust to these changes. To our best knowledge, RINO is the first method, which learns competitive features and correspondences directly from raw 3D geometry in a purely data-driven manner.

parameters, which is approximately $2/3$ of the DiffusionNet ($510,336$ trainable parameters), owing to the vector neuron representation. The dimension of the final output features is set to 256, same as baselines. We chose 200 eigenbases of the LBO and 30 eigenbases of the connection Laplacian for computing $\mathbf{C}$ and $\mathbf{Q}$ respectively. All hyperparameters λ in the total loss are set as 1.0, except λ_2 and λ_5 to be 0.1.

Datasets. To study the performance under near-isometric deformation, we use SCAPE [1], FAUST [8] and SHREC19 [41]. The SCAPE dataset contains a single person in 71 different poses, where the train/test split is 51/20. The FAUST dataset contains 10 persons in 10 different poses, leading to a total of 100 shapes, which are split into 80/20 for training and testing. The challenging SHREC19 dataset contains 44 human shapes with different mesh connectivity and is only used as a test set.

For non-isometric cases, we consider the SMAL [70] and DT4D [38] datasets. SMAL contains 49 four-legged animal shapes of 8 species, from which 5 are used for training and 3 unseen species for testing, resulting in a 29/20 split [22]. The DT4D dataset contains challenging humanoid shapes with large non-isometric deformation across different shape classes. Following [11, 22], we employ 9 different classes of humanoid shapes, resulting in a split of 198/95 shapes.

For real-world scan, we utilize the FAUSTSCAN [8] training dataset, which is the raw scan to create the registered FAUST dataset. The FAUSTSCAN training dataset contains a total of 100 shapes with 10 people in 10 different poses, which is split in the same way for train/test as FAUST. Since it is real-world captured human scans, it has the typical artifacts of missing parts due to occlusion or lack of sensor coverage and scan noise. We subsample the shapes to a resolution of $\sim$5k vertices for all evaluated methods.

We further stress test our method using a customized non-manifold version of FAUST, which is a dedicated non-manifold test set derived from the original test partition (20 test shapes), where half the shapes are decimated (#vertex reduced from 5k to 3k).

For partial shape matching, we employ SHREC16-Partiality [17, 25]. It has two sub-datasets, namely CUTS and HOLES, representing different partial modalities. It

contains 120 training and 153 testing shapes for CUTS; 80 training and 200 testing shapes for HOLES, respectively.

Computational complexity. The performance metrics, including the number of trainable parameters and mean runtime on the SMAL test set, are detailed in Tab. 5. For a fair comparison, we employ the same computation unit with four Intel Xeon E5-2697 CPU and one NVIDIA TITAN Xp GPU consistently for all methods. Our model establishes a new frontier by being both significantly smaller (approximately 30% fewer parameters than the next-best method) and substantially more accurate (half of the error of the closest competitor). While the current implementation demonstrates a marginal increase in runtime, it operates within the competitive range of all other methods, requiring less than 0.1 seconds to process a shape pair with over 5k vertices. This performance gap could be effectively mitigated through the development of a CUDA-accelerated implementation of the VN-layers.

11. Additional Experiments

11.1. xyz versus wks

Existing methods, including URSSM and HbrFM, demonstrate a selective preference for shape descriptors. Specifically, they employ xyz coordinates for the SMAL and SHREC16-Partiality datasets, but utilize wks for others, such as SCAPE and FAUST. This differentiation is likely necessitated by the significant non-isometries inherent in SMAL and SHREC16-Partiality, which inherently reduce the stability of wks. To rigorously analyze this effect, we calculated the mean values of the feature channels across semantically corresponding points for all shapes in both the SMAL and SCAPE datasets. The empirical evidence in Fig. 13 confirms that wks possesses a higher variance within SMAL than within SCAPE, whereas the xyz coordinate shows greater consistency. The inherent ability of xyz to represent raw shape geometries makes it a compelling choice for purely data-driven 3D learning in both research and practical applications.

11.2. Near-Isometric Matching

We compare performance on the near-isometric datasets SCAPE, FAUST and SHREC19 using both xyz and wks as input features. Specifically, we train on SCAPE and FAUST, and test across all three datasets, with quantitative results shown in Fig. 12. While all methods perform well when utilizing wks, ours is uniquely effective when relying solely on xyz, underscoring its extraordinary robustness to input type and its ability to learn salient shape information directly from raw data. This represents, to our knowledge, the first instance where xyz has achieved competitive results, if not better, against wks for near-isometric shapes.

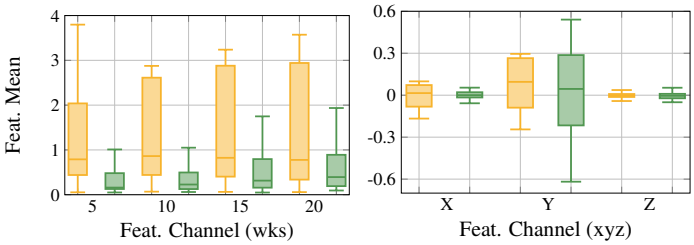

Figure 13. **Statistics of wks vs. xyz.** The box plot illustrates the statistical distribution of the mean feature channel values at semantically corresponding points across the entire shape collections of the SMAL and SCAPE datasets. The methodology involved aggregating the channel-wise mean of features across corresponding points for every shape, yielding a feature matrix (with dimensions #correspondences $\times$ #feature channels), where each column's distribution is plotted as a box. The statistics shown for the 5th, 10th, 15th, and 20th wks channels demonstrate a pronounced difference in behavior between the datasets. Specifically, the wks statistics for SMAL display a substantially larger variance than those for SCAPE. This higher dispersion is a direct consequence of the non-isometries present in the SMAL dataset, which contrasts sharply with the stability observed in the xyz descriptor statistics.

	FAUST	FAUST non-manifold
URSSM(wks)	1.6	17.6
HbrFM(wks)	1.4	16.6
Ours(wks)	1.6	15.6
Ours(xyz)	1.6	2.7

Table 6. **MGeoErr(↓) on FAUST non-manifold shapes.**

The high performance of baselines using wks also suggests that the near-isometric matching task has been addressed very well, hinting that the community should shift focus to more challenging non-isometric and real-world cases. The experiments also show that RINO generalized well across different datasets, suggesting that equivariance/invariance does not hinder generalization. Furthermore, the shapes in SHREC19 have all different discretizations, ours performs well in this challenging scenario under both xyz and wks as input features. Despite the discretization-dependent VN-EdgeConv operation at the beginning of RINO, the subsequent RINONet block learn adaptive diffusion time accordingly, rendering the entire RINONet robust to varying discretizations.

11.3. Matching Non-manifold Shapes

To underscore the advantage of directly processing raw geometry, we evaluate models pre-trained on FAUST against our customized FAUST non-manifold test set. Note that this setup violates the "nice mesh" assumption required by intrinsic descriptors like wks. Consequently, as shown in Tab. 6, methods relying on wks as input suffer a significant decline in matching performance. Crucially, our proposed

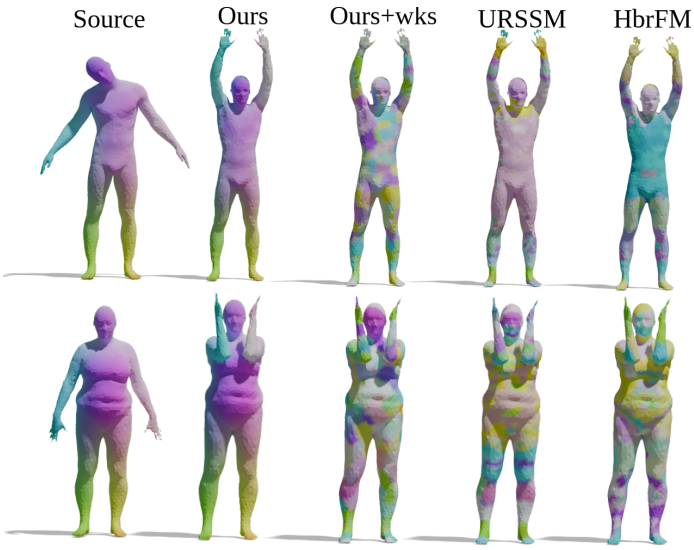

Figure 14. **Non-manifold shape matching.** Methods employing the wks as input exhibit a significant performance deterioration, even for SoTA methods such as URSSM and HbrFM, an observation that also applies to our proposed method (Ours+wks). Fortunately, our SO(3)-invariance network allows for effective learning directly from the raw xyz coordinates (Ours). This approach offers robustness against the difficulties posed by non-manifoldness, bypassing the fragility observed when using wks.

method, when directly utilizing the raw xyz coordinates, exhibits remarkable robustness with only a negligible performance degradation. This result highlights the enhanced robustness conferred by xyz input when confronting the topological complexities of non-manifold and highly discrepant data. See Fig. 14 for visual examples.

11.4. Real-World Scans

To evaluate all methods under real-world scans, we utilize the FAUSTSCAN training dataset, which contains typical artifacts of missing parts due to occlusion or lack of sensor coverage and scan noise. We reduce scan vertices to 5k, due to the computation of LBO and its eigendecomposition, and no further preprocessing is performed to preserve the noise, holes and artifacts in the original scans. The quantitative results are reported in Tab. 2 (col. FSCAN). Ours performs all baselines regardless of input features and qualitative results are illustrated in Fig. 22. This strong performance suggests a big step towards shape matching methods for real-world applications.

11.5. Human Segmentation

We compare against the SoTA DiffusionNet [54] across two feature inputs: hks and xyz. The results in Tab. 7 show that our proposed method achieves comparable accuracy to DiffusionNet when using hks. The key distinction arises with raw xyz input: RINONet exhibits complete stability, maintaining its performance with or without SO(3)-augmentation during training, whereas DiffusionNet suffers a substantial degradation when augmentation is removed. This robust, stable performance stems directly from the inbuilt SO(3)-invariance of our architecture, which simplifies the training pipeline and translates to enhanced training efficiency. In Fig. 15 we present additional qualitative results.

Method	**Human dataset [40]**
DiffusionNet(hks, w/o aug)	91.3%
Ours(hks, w/o aug)	91.0%
DiffusionNet(xyz, w/ aug)	90.3%
DiffusionNet(xyz, w/o aug)	83.6%
Ours(xyz, w/ aug)	90.6%
Ours(xyz, w/o aug)	90.1%

Table 7. **Segmentation accuracy (↑) on human dataset [40].**

No.	**Ablation Setting**	**SMAL**
1	w\o L_{struct}	7.3
2	w\o L_{couple}	7.6
3	w\o L_{contr}	5.7
4	w\o the Qmaps branch	5.1
5	w\o $L_{\mathbf{\Pi Q}}$	5.0
6	Ours + coupling $\mathbf{C}, \mathbf{Q}$	26.9
7	Ours	**4.6**

Table 8. **Ablation study on SMAL.**

11.6. RNA Segmentation

To test RINO on other diverse data modalities, we train it on RNA dataset [46] with the same training protocol as DiffusionNet (i.e. the I/I setting). RINONet achieves 85.8% accuracy, versus 72.1% of DiffusionNet (cf. Fig. 16).

11.7. Rigid Registration

As a proof-of-concept, we train RINO on the Stanford Bunny dataset [60], predicting per-point SO(3)-invariant features. More specifically, we take two bunnies of different resolution and randomly rotate one of them. Then, a correspondence search is conducted in the feature space, and the rigid registration is computed using Procrustes Analysis. Our final mean geodesic error of correspondences is 2.4, and the Chamfer distance is 0.018. A visual result is shown in Fig. 17.

11.8. Ablation

We ablate the design of both our architecture and loss terms, and report the results on the challenging non-isometric SMAL dataset in Tab. 8. To this end, we compare each row with the last row (our full model) to assess the importance of individual terms. The performance degrades most

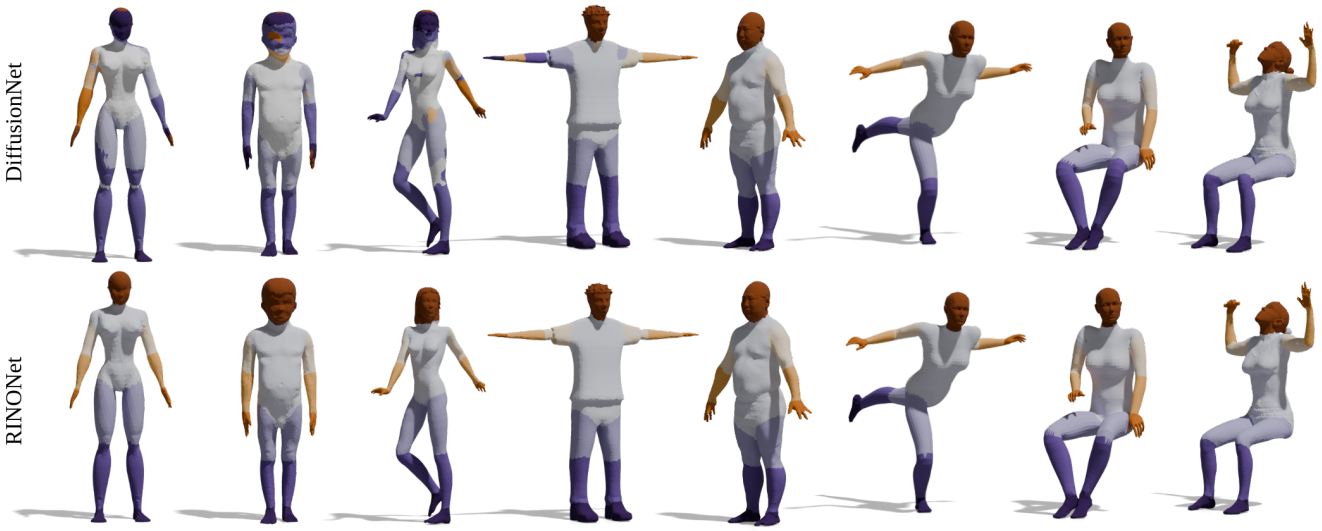

Figure 15. **Human Segmentation.** We show the segmentation results of DiffusionNet and our RINONet on the composite dataset [40]. Our RINONet produces sharper and more semantically meaningful segments.

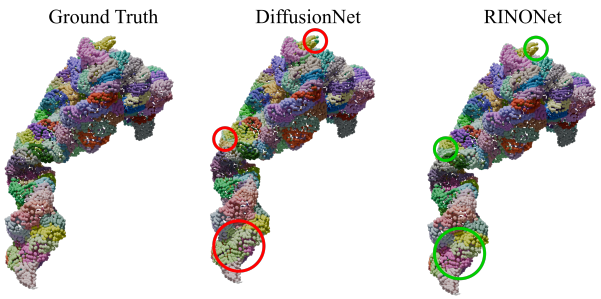

Figure 16. RNA segmentation

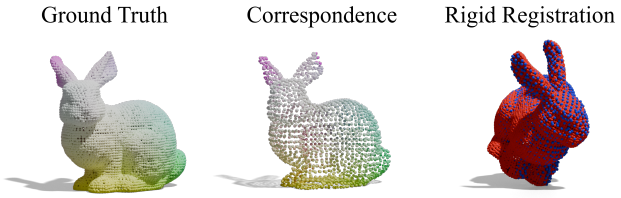

Figure 17. 3D rigid registration on Stanford dataset (bunny)

when L_{struct} or L_{couple} is removed (first 2 rows), indicating the central role of structural property and consistency of different map representations. Interestingly, this finding echoes with previous works [11, 49, 50]. Removing the L_{contr} term has a moderate impact on the final results, since it helps to learn more distinctive rotation-invariant features. Looking at the fourth and fifth rows, we conclude that the integration of complex functional maps encourages the first-order consistency of rotation-invariant features, which in turn helps to improve the accuracy of estimated correspondences. Finally, comparing the last two rows shows that excessive coupling can hinder learning performance.

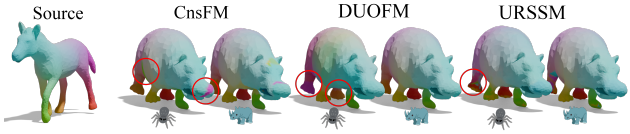

Figure 19. Backbone comparison. Spider emoji denotes DiffusionNet and rhinoceros emoji denotes RINONet.

	mGeoErr (↓)	DUOFM	CnsFM	URSSM	HbrFM
(i)	DiffusionNet	25.2	9.1	24.8	26.4
	RINONet	**5.7**	**7.3**	**5.1**	**7.4**
		SCAPE	FAUST	SMAL	DT4D
(ii)	DiffusionNet	26.9	30.0	24.8	59.3
	RINONet	**2.0**	1.6	5.1	5.9
	RINONet + CFMaps	**2.0**	**1.5**	**4.6**	**5.3**

Table 9. Ablation: SO(3)-invariant features & CFMaps

To study the relative performance gain of invariant features and CFMaps, we ablate further in Tab. 9 by two experiments under the SO(3)/SO(3) setup. (i) We take baselines and swap in our RINONet as the feature extractor and experiment on SMAL; RINONet always improves over DiffusionNet, highlighting the benefit of SO(3)-invariant features. (cf. Fig. 19) (ii) We compare DiffusionNet and RINONet without CFMaps, and RINONet with CFMaps, training them on four different datasets, showing that the largest performance gain comes from SO(3)-invariant features, which can be further improved by CFMaps.

11.9. More Qualitative Results

We show more matching results of RINO on the challenging noisy shapes in Fig. 20, and ours is the only method that can produce reliable correspondences under the noise level $\sigma =$ 6e-3, where baselines already start to produce spurious correspondences.

We also present additional qualitative results on the SHREC16-Partiality datasets. Here we show all partial shapes of all categories in Fig. 23-36, and RINO manifests strong robustness under this challenging incomplete shape by directly consuming the raw 3D geometry. The highly accurate correspondences suggest remarkable quality of the learned rotation-invariant features.

11.10. Failure Cases

We notice that for highly non-isometric pairs, our estimated correspondences are sometimes locally non-smooth. As shown in Fig. 21, RINO confuses when one humanoid misses completely his lower arm. This leads to wrong correspondences in the region of missing parts in the other shape. We hypothesize that incorporating further intrinsic prior during training or post-processing could solve these artifacts.

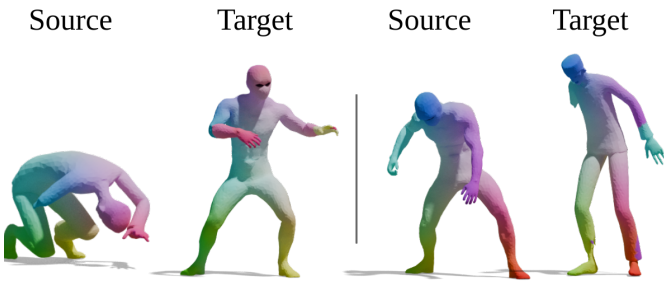

Figure 21. **Failure cases.**

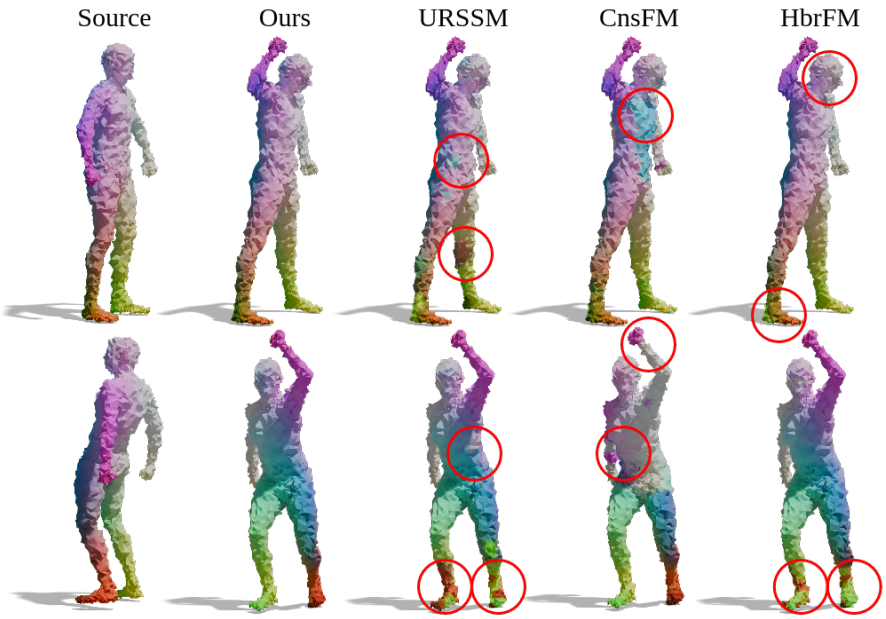

Figure 20. **Matching shapes with Gaussian perturbation.**

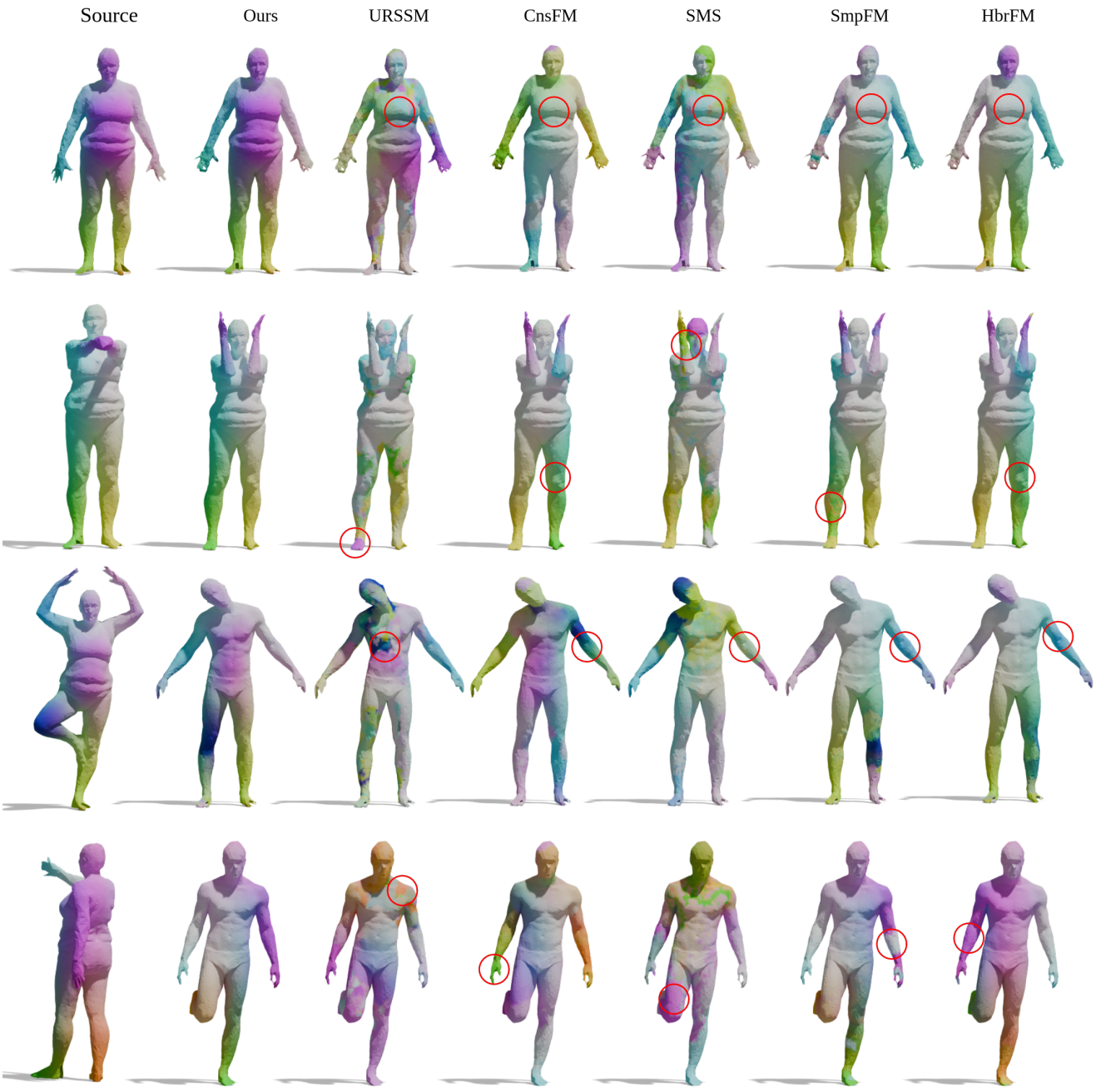

Figure 22. **Qualitative results on the real-world FAUSTSCAN.**

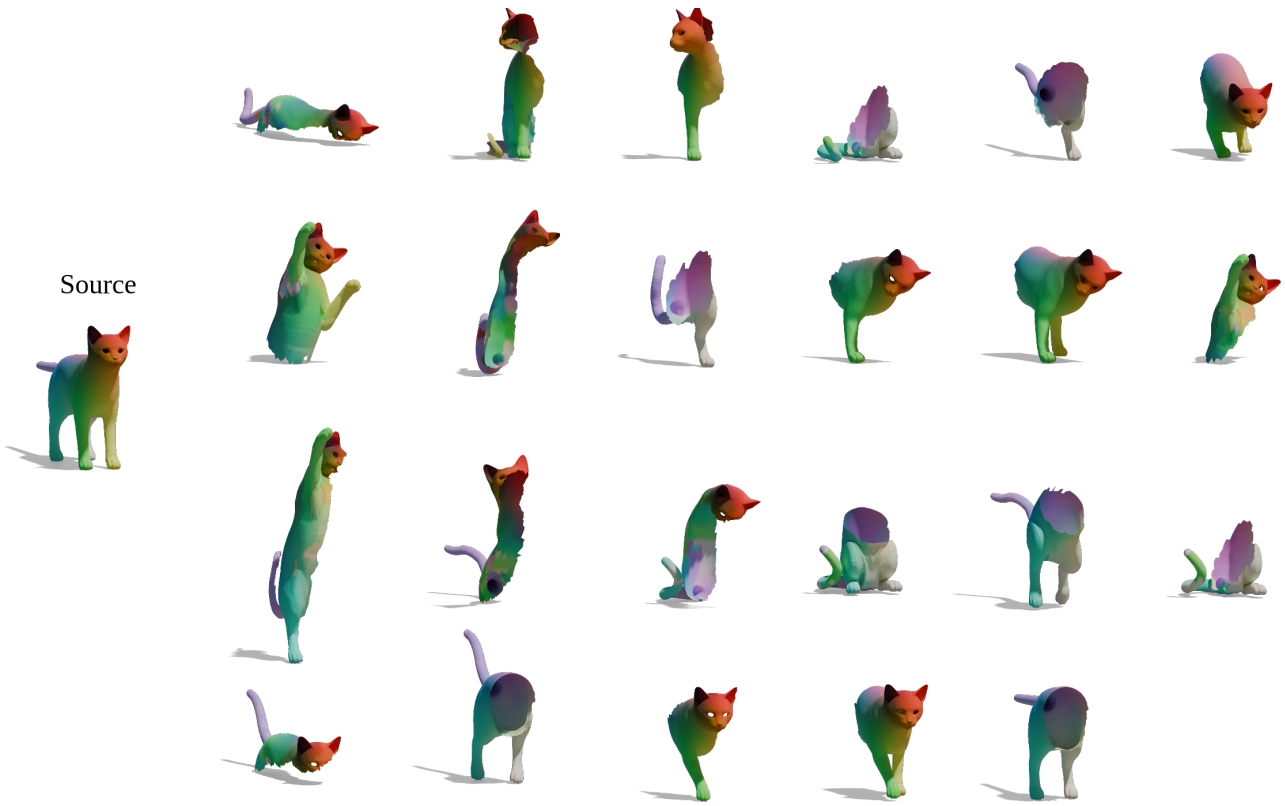

Figure 23. SHREC16-Partiality CUTS cat

Figure 24. SHREC16-Partiality CUTS david

Figure 25. SHREC16-Partiality CUTS centaur

Figure 26. SHREC16-Partiality CUTS dog

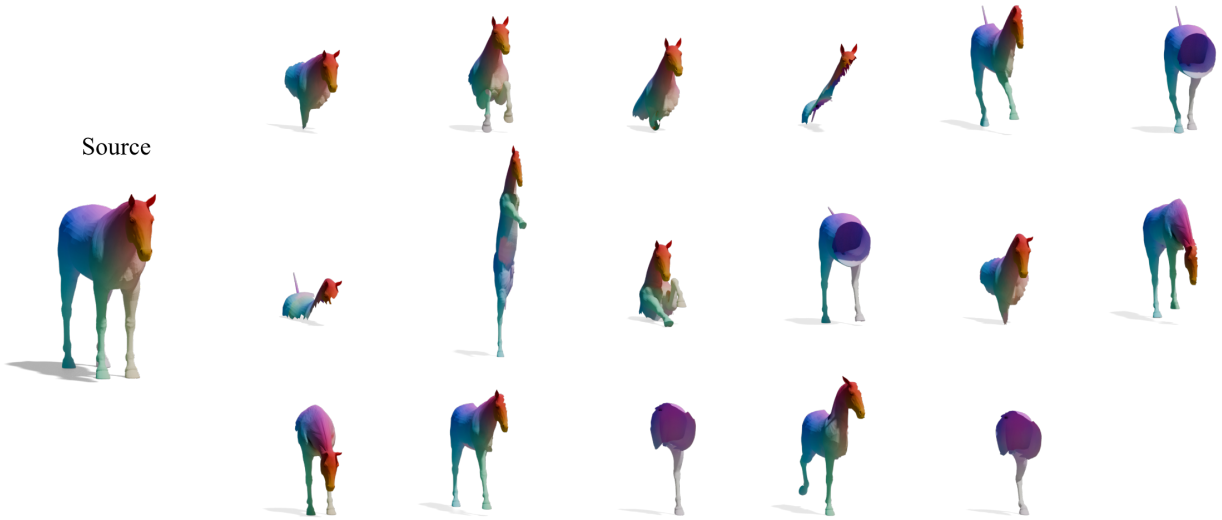

Figure 27. SHREC16-Partiality CUTS horse

Figure 28. SHREC16-Partiality CUTS victoria

Source

Figure 29. SHREC16-Partiality CUTS michael

Figure 30. SHREC16-Partiality HOLES cat

Figure 31. SHREC16-Partiality HOLES david

Figure 32. SHREC16-Partiality HOLES centaur

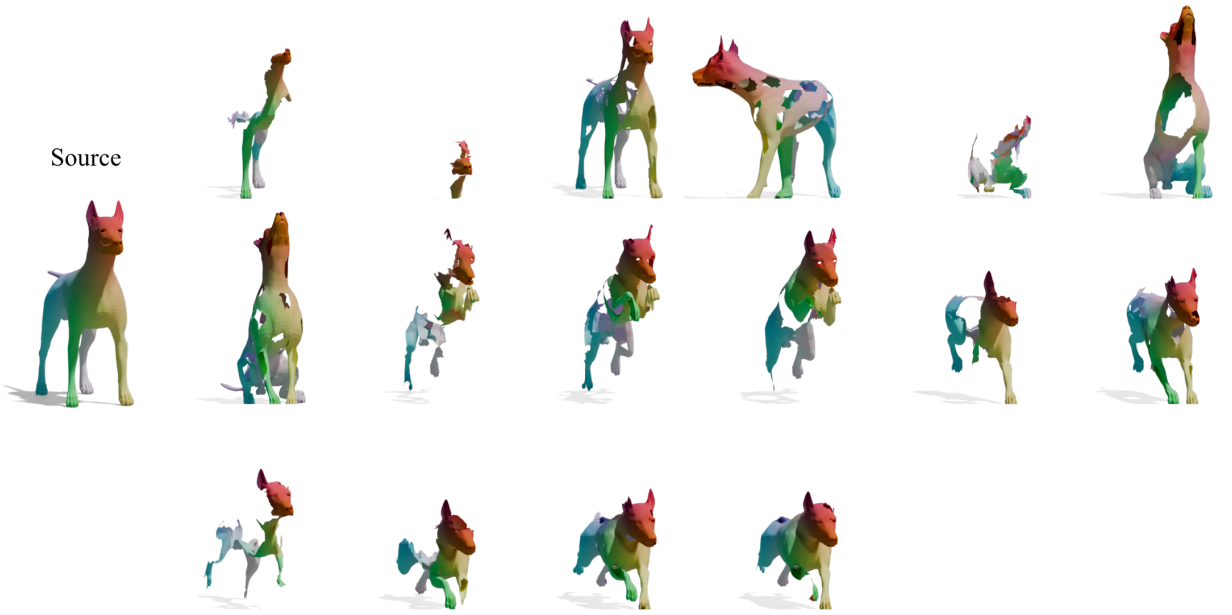

Figure 33. SHREC16-Partiality HOLES dog

Figure 34. SHREC16-Partiality HOLES horse

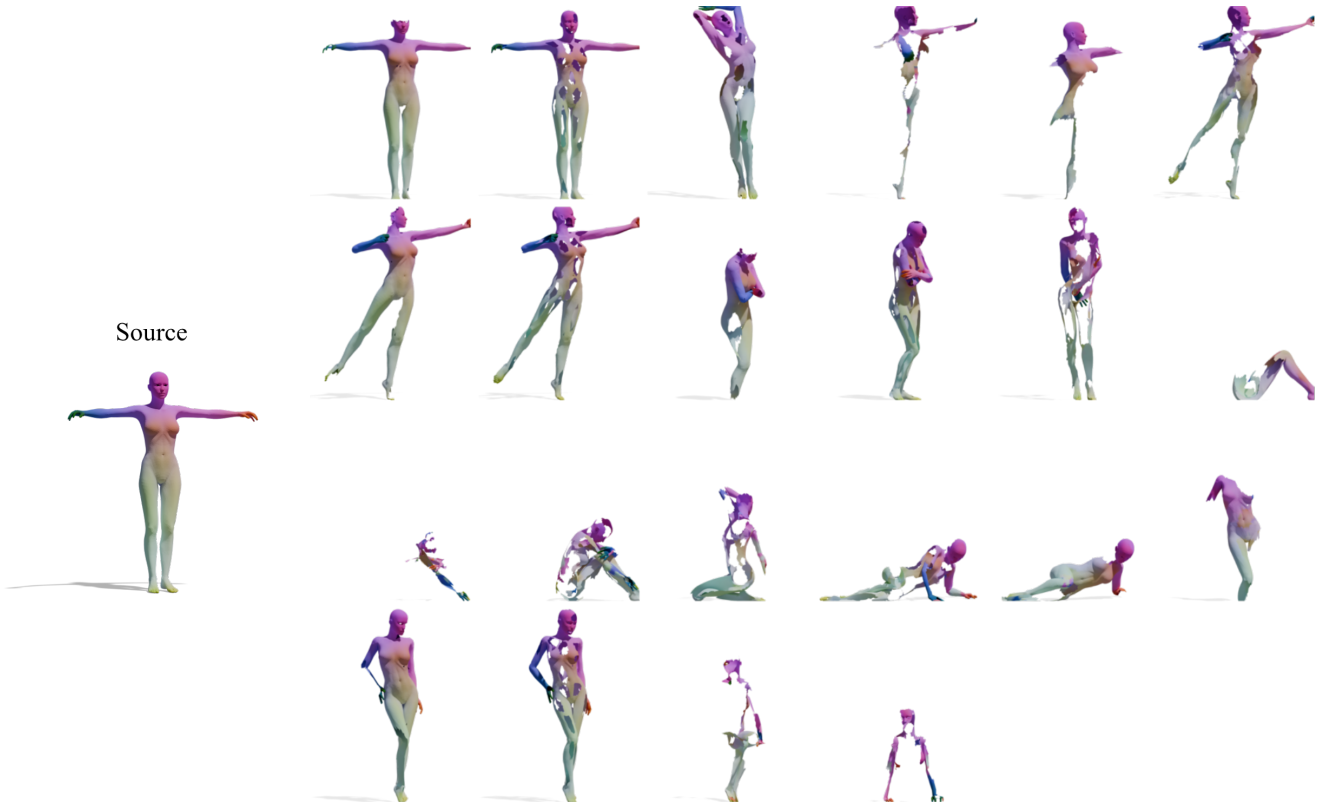

Figure 35. SHREC16-Partiality HOLES victoria

Figure 36. SHREC16-Partiality HOLES michael