Fast Markov Random Field Optimisation for Topologically Noisy 3D Shape Matching

Supplementary Material

8. Details on Triangle Label Space

We restrict the triangle label space such that each triangle of the source shape can be mapped to a vertex, an edge, or a triangle of the target shape. In addition, we sample new triangles (on the target shape) such that vertices of these new triangles are at most k edges apart and such that resulting triangles are oriented according to normal orientations of surface normals (we use $k = 2$).

Further, for the labels that map a triangle (of the source shape) to an edge (on the target shape) or to a triangle (on the target shape) we include all possible rotations. In particular, this means:

- a triangle $f_x = (x_1, x_2, x_3) \in \mathcal{F}_\mathcal{X}$ and an edge $(y_1, y_2) \in \mathcal{E}_\mathcal{Y}$ yield the labels
  - $\left(\binom{x_1}{y_1}, \binom{x_2}{y_2}, \binom{x_3}{y_2}\right), \left(\binom{x_1}{y_1}, \binom{x_2}{y_1}, \binom{x_3}{y_2}\right)$,
  - $\left(\binom{x_2}{y_1}, \binom{x_3}{y_2}, \binom{x_1}{y_2}\right), \left(\binom{x_2}{y_1}, \binom{x_3}{y_1}, \binom{x_1}{y_2}\right)$,
  - $\left(\binom{x_3}{y_1}, \binom{x_1}{y_2}, \binom{x_2}{y_2}\right), \left(\binom{x_3}{y_1}, \binom{x_1}{y_1}, \binom{x_2}{y_2}\right)$;
- a triangle $f_x = (x_1, x_2, x_3) \in \mathcal{F}_\mathcal{X}$ and a triangle $(y_1, y_2, y_3) \in \mathcal{F}_\mathcal{Y}$ yield the labels
  - $\left(\binom{x_1}{y_1}, \binom{x_2}{y_2}, \binom{x_3}{y_3}\right)$,
  - $\left(\binom{x_2}{y_1}, \binom{x_3}{y_2}, \binom{x_1}{y_3}\right)$,
  - $\left(\binom{x_3}{y_1}, \binom{x_1}{y_2}, \binom{x_2}{y_3}\right)$.

In summary, each triangle label space has $|\mathcal{V}_\mathcal{Y}| + 3 \cdot 2 \cdot |\mathcal{E}_\mathcal{Y}| + 3 \cdot |\mathcal{F}_\mathcal{Y}| + 3 \cdot D$ many elements, where D is the number of newly sampled triangles and for $k = 2$ we can estimate $D \approx |\mathcal{F}_\mathcal{Y}|$.

9. Relation to Assignment Matrix

Each element in the label space $\mathcal{L}$ (see Sec. 4.2) can be interpreted as an entry in an assignment matrix (i.e. a binary matrix with a single non-zero entry per row) where rows correspond to triangles on shape $\mathcal{X}$ and columns correspond to surface elements (i.e. vertices, edges, triangles) on shape $\mathcal{Y}$.

10. Relation to Dual Mesh Graph

Our definition of neighbouring triangles (cf. Eq. (2)) is equal to the *dual mesh* graph $\mathcal{X}_{\text{dual}} = (\mathcal{F}_\mathcal{X}, \mathcal{A}_\mathcal{X})$ by interpreting the triangles $\mathcal{F}_\mathcal{X}$ of shape $\mathcal{X}$ as vertices in the the dual mesh graph and by interpreting their adjacency relationship $\mathcal{A}_\mathcal{X}$ as edges in the dual mesh graph.

11. Details on Our Algorithm

In each iteration, the α-expansion algorithm aims to "expand" a single label α, i.e. its searches for vertices that, if assigned to label α, reduce the overall energy (cf. (1)) compared to the current label assignment. This can be done optimally using graph-cuts.

In our case, each label can only be assigned to a single vertex (i.e. a triangle on the source shape, see also Sec. 10) and thus we do not need to solve a graph-cut problem but we rather simply check if a label reduces the overall energy when assigned to a vertex compared to the current assignment. Alternatively and equivalently, for each vertex, we can also search for a label that improves the overall energy compared to the current label assignment. Thus, we implement the following algorithm:

1. We **initialise** the label assignment for each triangle (on the source shape) using the label with the lowest cost according to unary terms (in fact, we also incorporate the unary costs of its three neighbouring triangles when all four triangles get assigned labels such that they are mapped to four neighbouring triangles on the target shapes).
2. Then, until there is no improving label assignment, we **iterate** through all triangles on the source shape and search for a new label that improves the overall energy (compared to the current assignment).

We further follow the idea of adaptive cycles implemented in the original α-expansion algorithm [13]. This means, that we only "expand" labels in the next iteration if they resulted in an improved energy in the current iteration. If no label was leading to an improved energy, we "expand" all labels again.

We can estimate the runtime complexity of a single iteration with $\mathcal{O}(|\mathcal{F}_\mathcal{X}||\mathcal{F}_\mathcal{Y}|)$ since we iterate through all labels in each iteration (and since we restrict the label spaces, see also Sec. 8).

12. Detailed Proof

In the following, we provide a detailed proof that our pairwise term is a pseudometric, i.e. we show that the maximum between two pseudometrics remains a pseudometric.

Lemma 8 (Maximum of pseudometrics)**.** *The maximum $d_3(\ell_1, \ell_2) = \max\{d_1(\ell_1, \ell_2), d_2(\ell_1, \ell_2)\}$ of two pseudometrics d_1 and d_2 on the same metric space $\mathcal{L}$, i.e. $\ell_1, \ell_2 \in \mathcal{L}$, is also a pseudometric, i.e. for any $\ell_1, \ell_2, \ell_3 \in \mathcal{L}$, it holds that*

(i) $d_3(\ell_1, \ell_2) \geq 0$
(ii) $d_3(\ell_1, \ell_2) = d_3(\ell_2, \ell_1)$;
(iii) $d_3(\ell_1, \ell_3) \leq d_3(\ell_1, \ell_2) + d_3(\ell_2, \ell_3)$.

Proof. In the following we show (i)-(iii)

(i) $d_1(\ell_1, \ell_2) \geq 0$ and $d_2(\ell_1, \ell_2) \geq 0 \Rightarrow d_3(\ell_1, \ell_2) \geq 0$

(ii)

$$\begin{aligned} d_3(\ell_1, \ell_2) &= \max\{d_1(\ell_1, \ell_2), d_2(\ell_1, \ell_2)\} \\ &= \max\{d_1(\ell_2, \ell_1), d_2(\ell_2, \ell_1)\} \\ &= d_3(\ell_2, \ell_1) \end{aligned} \tag{5}$$

(iii)

$$\begin{aligned} d_3(\ell_1, \ell_3) &= \max\{d_1(\ell_1, \ell_3), d_2(\ell_1, \ell_3)\} \\ &\leq \max\{d_1(\ell_1, \ell_2) + d_1(\ell_2, \ell_3), \\ &\qquad d_2(\ell_1, \ell_2) + d_2(\ell_2, \ell_3)\} \\ &\leq \max\{d_1(\ell_1, \ell_2), d_2(\ell_1, \ell_2)\} + \\ &\quad \max\{d_1(\ell_2, \ell_3), d_2(\ell_2, \ell_3)\} \\ &= d_3(\ell_1, \ell_2) + d_3(\ell_2, \ell_3) \end{aligned} \tag{6}$$

The last inequality holds since $\max\{a+b, c+d\} \leq \max\{a, c\} + \max\{b, d\}$, i.e. the maximum of two sums is less or equal to sum of the maximum of the first and of the second summands.

□

13. Matching Extraction

A solution of (SM-MRF), yields a label assignment for each triangle of shape $\mathcal{X}$ which in turn can be interpreted as a mapping of the three vertices of a single triangle to three vertices on the target shape, cf. (3). As such, the shared vertices of neighbouring triangles might be mapped to different vertices on the target shape. In our formalism, we minimise their geodesic distance and thus we expect these mappings to be nearby vertices on the target shape. To extract the final vertex to vertex matching from these individual mappings we find their geodesic center, i.e. the vertex on the target shape which is closest in terms of geodesic distance to all mapped vertices, cf. Fig. 8.

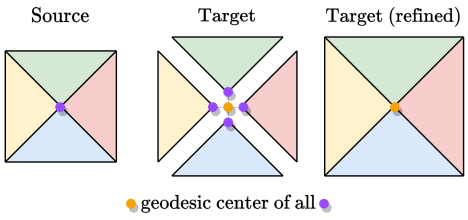

Figure 8. Illustration of our **matching extraction method**. For the final vertex to vertex mapping we find the vertex (in orange) that is in the geodesic center of all mapped vertices of the purple vertex.

	WKS	Ours+WKS	SHOT	Ours+SHOT
Mean Geo. Error	0.21	**0.19**	0.34	**0.32**
Mean Dir. Energy.	5.71	**3.94**	16.9	**12.6**

Table 2. **Feature descriptor ablation.** Our method improves both geodesic errors and Dirichlet energies when combined with WKS [6] or SHOT [72] descriptors, evaluated on ten shapes of FAUST.

Weighting Parameter λ	1	10	100	1000
Mean Geo. Error	0.038	0.037	**0.036**	0.37
Mean Dir. Energy.	1.25	1.11	**1.05**	**1.05**

Table 3. Ablation study w.r.t. **scaling parameter** λ.

Robust Unary Cost	no Ψ	$c = 0.01$	$c = 0.1$	$c = 1.0$
Mean Geo. Error	0.037	0.038	**0.036**	0.38
Mean Dir. Energy.	1.26	1.17	**1.05**	**1.08**

Table 4. Ablation study on the **robust unary cost function** Ψ.

14. Additional Results

In this section we provide more experimental results. First we show several ablation studies. After that we show that our approach is also applicable to partial shapes. Finally, we conduct experiments on high resolution shapes. In summary, the experiments in the main paper and these additional experiments emphasise that our method bridges the gap to overcome inherent challenges of 3D shape matching and topologically noisy 3D shape matching. In particular, our method is accurate, leads to smooth results by incorporating neighbourhood relations, is applicable to various types of input noise, and remains efficient through our custom implementation of the α-expansion algorithm.

14.1. Ablation Studies

In **??**, we evaluate the performance of our method w.r.t. different features and we can see that both for WKS [6] as well as for SHOT [72] descriptors our method improves results. This suggests the generalisablity of our approach to other feature descriptors.

Further, in Tab. 3 we conduct ablation studies w.r.t. to the scaling parameter λ from which we can see that $\lambda = 100$ is best in this test.

In addition, in Tab. 4 we ablate on the parameter c involved in the robust unary cost function Ψ.

Similarly, in **??**, we ablate the effects of the robust pairwise cost function ϕ.

In Fig. 9, we evaluate the performance of our method w.r.t. varying discretisation of the shapes. From that, we can see that our method is relatively robust to different sampling densities.

Finally, we conduct a robustness experiment w.r.t. topo-

Robust Pairwise Cost	with ϕ	without ϕ
Mean Geo. Error	0.038	**0.036**
Mean Dir. Energy.	1.21	**1.05**

Table 5. Ablation study on the **robust pairwise cost function** ϕ.

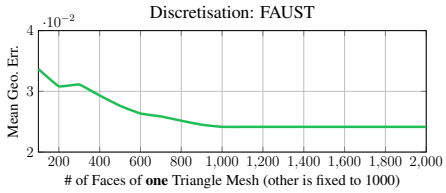

Figure 9. Performance of our method under **varying resolutions**, where we fix one shape to 1000 triangles and vary the resolution of the other shape. We report mean geodesic errors over five instances of the FAUST dataset. While we can see discretisation artefacts (i.e. higher geodesic errors on lower resolutions), overall we can conclude that our method is robust to discretisation differences.

k	0	2	4	6	8
Mean Geo. Error	0.027	0.056	0.093	0.131	0.175
Mean Dir. Energy	0.428	1.939	5.720	11.23	17.47

k	0	2	4	6	8
Mean Geo. Error	0.014	0.025	0.034	0.043	0.055
Mean Dir. Energy	0.737	2.725	5.315	9.767	17.80

Figure 10. **Mesh topology noise ablation study** on lower resolution shape (top), i.e. shapes with 1k triangles and higher resolution shapes (bottom), i.e. shapes with 10k triangles.

logical (mesh) noise on five shapes of the FAUST dataset. For each pair of source and target shapes we alter the mesh of the target shape, i.e. we randomly rewire edges within a k-ring neighborhood (for $k \in \{0, 2, 4, 6, 8\}$). In Fig. 10, we can see respective mean geodesic errors and Dirichlet energies for lower and higher resolution shapes with topological mesh noise. These results suggest that the geodesic errors produced by our method are linearly related to the amount of noise linearly with the amount of topological mesh noise.

14.2. Partial Shape Matching

In Tab. 6, we show results for partial shape matching on the SHREC16 cuts test set [24]. We note that we compare only methods that are applicable for this setting.

14.3. High-Resolution Shape Matching

In Fig. 11, we plot mean geodesic errors over mean Dirichlet energies for shapes at a resolution of 1000 triangles. From that, we can see that our method is the only method that consistently produces accurate (low geodesic errors) and smooth results (low Dirichlet energies). For better visibility, we repeat qualitative results of the main paper (while increasing their size) and show them with additional qualitative results in Fig. 13. In Fig. 12, we show results for shapes sampled to a resolution of 10k triangles (to this end we first subdivide each triangle into four triangles and then downsample shapes to 10k triangles using algorithms [33] such that resulting meshes have a more uniform triangle density). Since geometrically consistent methods SpiderMatch [62], GeCo [63], and SupaMatch [2] do not scale to these resolutions (cf. Fig. 5) we do not report numbers for these methods on higher resolution shapes.

Method	SHREC'16 cuts (1k)		SHREC'16 cuts (10k)	
	Geo.Err.	Dir.Eng.	Geo.Err.	Dir.Eng.
ULRSSM [16]	0.071	1.26	<u>0.061</u>	<u>2.33</u>
KernelMatch [83]	0.309	6.60	0.319	26.97
SpiderMatch [62]	0.066	0.61	n.a.	n.a.
GeCo [63]	**0.062**	**0.44**	n.a.	n.a.
Ours	<u>0.064</u>	<u>0.49</u>	**0.057**	**1.83**

Table 6. Quantitive results on **partial** shapes (SHREC'16 cuts) at a shape resolution of 1k triangles and 10k triangles. Methods SpiderMatch and GeCo do not scale to 10k triangles (indicated as n.a.). We can see that our method is competitive also in the partial-to-full setting as it yields second best results (on low resolution shapes) and best results (on high resolution shapes).

15. TOPFAUST Visualisations

In Fig. 14, we visualise the twenty test shapes of the TOPFAUST dataset that we have created, see Sec. 5 for more details. We can see that the Poisson surface reconstruction leads to merging of close surfaces such that the genus of shapes changes when e.g. the hands of the shapes touch or when one foot touches the other leg.

We will release code to recreate the TOPFAUST dataset from the original FAUST shapes.

16. Raw Data

In Tab. 7 and Tab. 8, we show raw shape matching results on shapes at resolution of 1000 triangles. For better interpretability, we recommend using the plots in Fig. 11.

In Tab. 9 and Tab. 10, we show raw shape matching results on shapes at resolution of 10k triangles. For better interpretability, we recommend using the plots in Fig. 12.

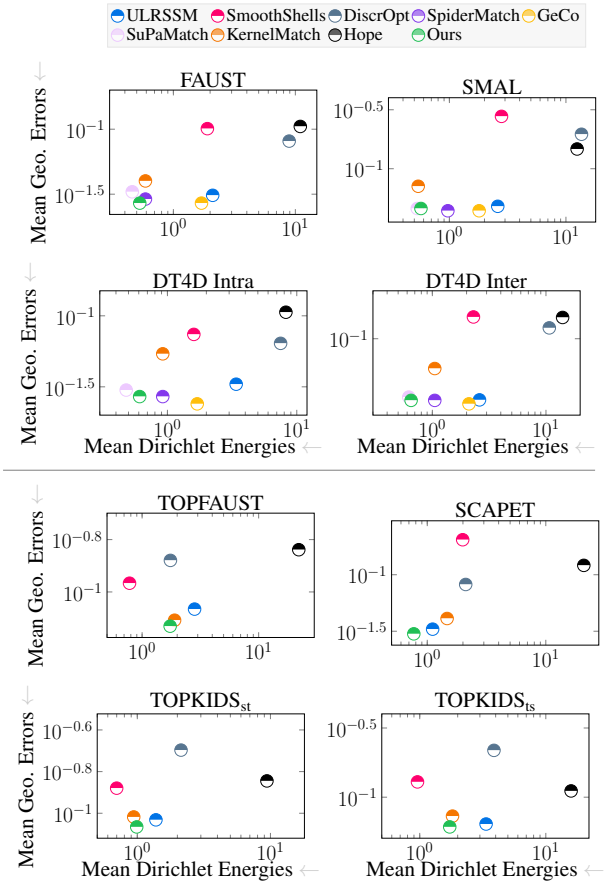

Figure 11. Shape matching results on shapes with resolution of 1000 triangles. (Top) Comparison of methods w.r.t. geodesic errors and *Dirichlet energies* on **clean data**. We can see that our method is both efficient and accurate. (Bottom) Comparison of methods w.r.t. geodesic errors and *Dirichlet energies* on **topologically noisy shapes**. We can see that our method is the only one that yields both *smooth* and *accurate* results on clean and topologically noisy data while other methods are either not applicable to both scenarios, or are either just smooth or just accurate.

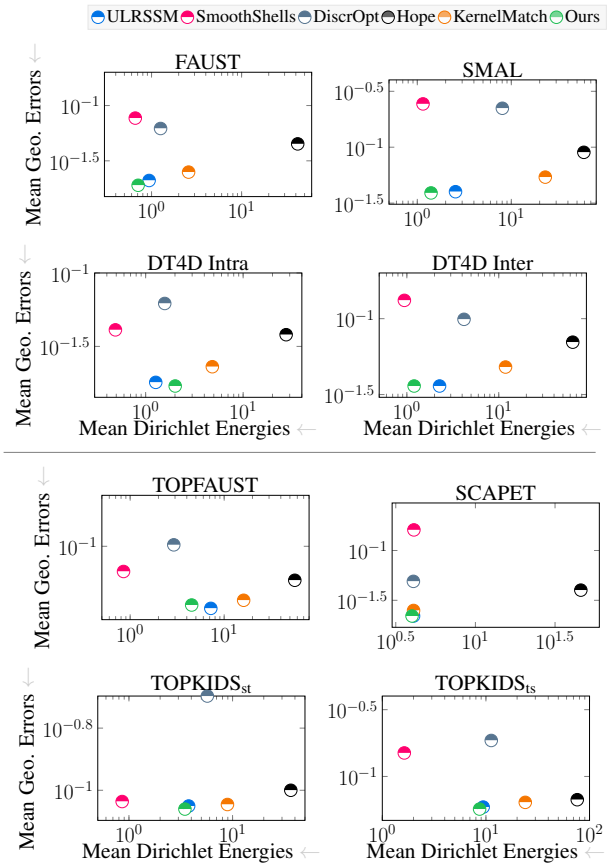

Figure 12. Shape matching results on shapes with resolution of 10k triangles. (Top) Comparison of methods w.r.t. geodesic errors and *Dirichlet energies* on **clean data**. We can see that our method is both efficient and accurate. (Bottom) Comparison of methods w.r.t. geodesic errors and *Dirichlet energies* on **topologically noisy shapes**. Also on higher resolution shapes, we can see that our method is the only one that yields both *smooth* and *accurate* results on clean and topologically noisy data.

Figure 13. **Qualitative** results on datasets DT4D Intra (first column), FAUST (second column), SMAL (third column), DT4D Inter (fourth column), TOPKIDS$_{st}$ (fifth column), TOPKIDS$_{st}$ (sixth column), SCAPET (seventh & eighth column), and TOPFAUST (ninth column). Columns marked with "*" are repeated from the main paper. From examples marked with red arrows, we can see that our method leads to smoother results (compared to e.g. ULRSSM), does not yield wrongly matched spots (as for example KernelMatch does), and does not produce left-right flips (as DiscreteOpt and SmoothShells do).

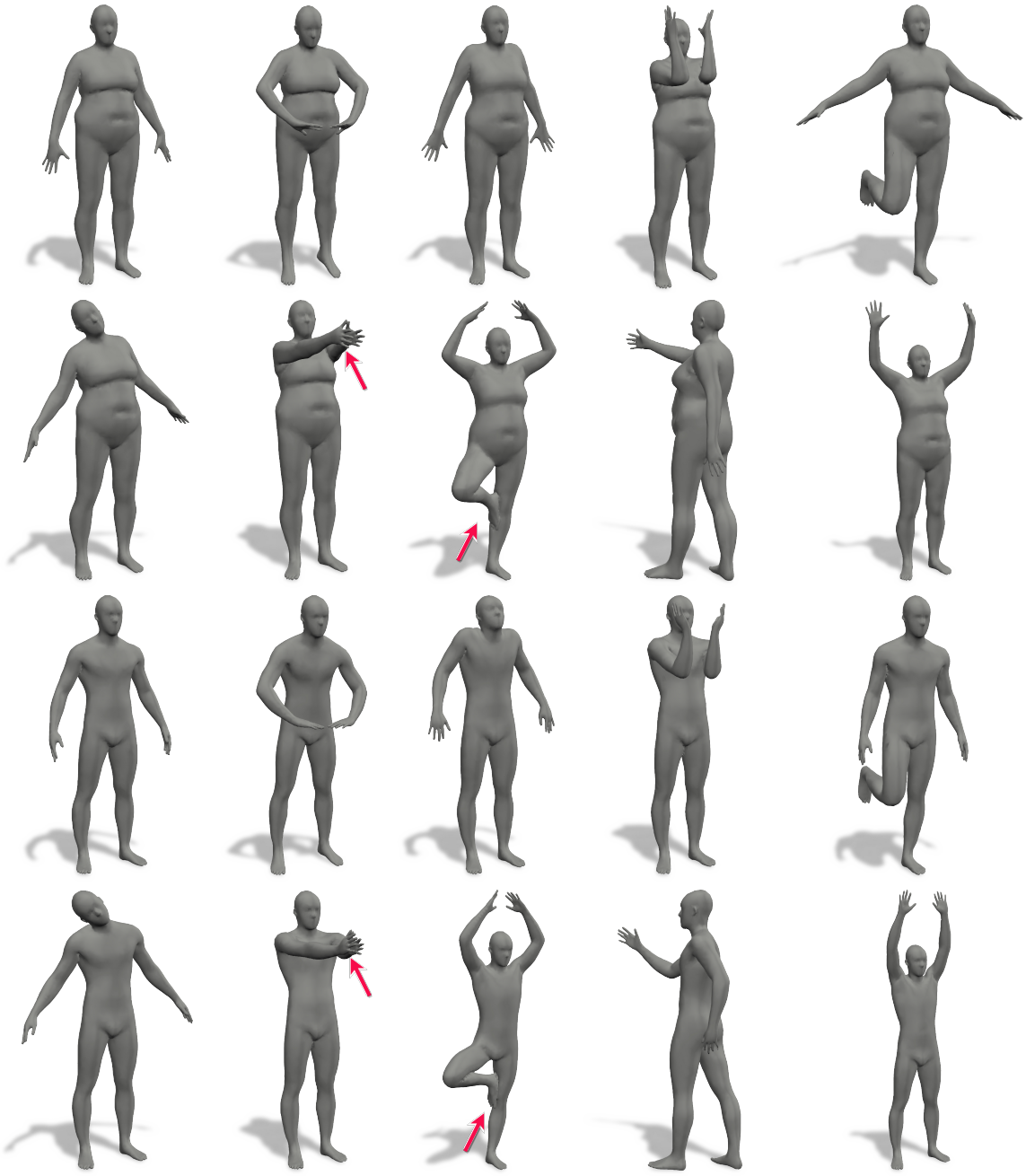

Figure 14. Visualisation of the twenty test set shapes (we use the same test set shapes as for FAUST dataset, see [16] for more details) of our newly created **TOPFAUST** dataset. For example, at positions of red arrows, nearby surfaces are close together resulting in a change of genus of the shape. Furthermore, for example in the armpits, close surfaces merge such that the local neighbourhoods of vertices change. From these 20 test set pairs, we sample pairs of shapes such that at least one shape contains a change in genus.

Method	FAUST	SMAL	DT4D Intra	DT4D Inter
ULRSSM [16]	0.031	0.048	0.033	0.041
SmoothShells [26]	0.379	0.376	0.373	0.420
DiscrOpt [60]	0.110	0.268	0.075	0.170
Hope [27]	0.105	0.147	0.106	0.136
KernelMatch [83]	0.040	0.071	0.054	0.065
SpiderMatch [62]	0.029	**0.044**	0.027	0.041
GeCo [63]	**0.027**	**0.044**	**0.024**	**0.039**
SuPaMatch [2]	0.033	0.046	0.030	0.043
Ours	**0.027**	0.046	0.027	0.041

Method	FAUST	SMAL	DT4D Intra	DT4D Inter
ULRSSM [16]	2.11	2.62	3.43	2.59
SmoothShells [26]	1.88	2.81	1.62	2.29
DiscrOpt [60]	8.92	13.61	7.52	10.72
Hope [27]	10.98	12.47	8.24	14.01
KernelMatch [83]	0.59	0.97	0.92	1.05
SpiderMatch [62]	1.71	1.83	1.74	2.07
GeCo [63]	**0.46**	**0.53**	**0.48**	**0.62**
SuPaMatch [2]	0.52	0.54	0.51	0.60
Ours	0.53	0.57	0.61	0.65

Table 7. Raw results on clean shapes at a resolution of 1000 triangles: (Left) Comparison of **mean geodesic errors** (↓). (Right) Comparison of **mean Dirichlet energies** (↓). In Fig. 11 (top), we present both quantities in a single plot to allow a straightforward assessment of which method lies on the Pareto front.

Method	TOPFAUST	$\text{TOPKIDS}_{\text{st}}$	$\text{TOPKIDS}_{\text{ts}}$	SCAPET
ULRSSM [16]	0.086	0.093	0.064	0.033
SmoothShells [26]	0.108	0.132	0.129	0.204
DiscrOpt [60]	0.132	0.201	0.218	0.082
Hope [27]	0.145	0.143	0.111	0.121
KernelMatch [83]	0.078	0.096	0.073	0.041
Ours	**0.074**	**0.086**	**0.061**	**0.030**

Method	TOPFAUST	$\text{TOPKIDS}_{\text{st}}$	$\text{TOPKIDS}_{\text{ts}}$	SCAPET
ULRSSM [16]	2.82	1.38	3.36	1.11
SmoothShells [26]	**0.78**	**0.70**	**0.96**	1.99
DiscrOpt [60]	1.75	2.13	3.88	2.11
Hope [27]	22.10	9.41	15.87	20.93
KernelMatch [83]	1.90	0.94	1.82	1.47
Ours	1.74	0.99	1.73	**0.77**

Table 8. Raw results on topologically noisy shapes at a resolution of 1000 triangles: (Right) Comparison of **mean geodesic errors** (↓). (Left) Comparison of **mean Dirichlet energies** (↓). In Fig. 11 (bottom), we present both quantities in a single plot to allow a straightforward assessment of which method lies on the Pareto front.

Method	FAUST	SMAL	DT4D Intra	DT4D Inter
ULRSSM [16]	0.021	0.040	0.018	**0.036**
SmoothShells [26]	0.077	0.244	0.041	0.132
DiscrOpt [60]	0.062	0.223	0.062	0.099
Hope [27]	0.045	0.090	0.038	0.070
KernelMatch [83]	0.025	0.054	0.023	0.048
Ours	**0.019**	**0.039**	**0.017**	**0.036**

Method	FAUST	SMAL	DT4D Intra	DT4D Inter
ULRSSM [16]	0.94	2.55	1.27	2.27
SmoothShells [26]	**0.66**	**1.15**	**0.49**	**0.94**
DiscrOpt [60]	1.26	8.03	1.57	4.19
Hope [27]	42.0	59.1	27.6	63.9
KernelMatch [83]	2.58	22.98	4.83	11.86
Ours	0.71	1.40	2.01	1.24

Table 9. Raw results on clean shapes at a resolution of 10k triangles: (Left) Comparison of **mean geodesic errors** (↓). (Right) Comparison of **Dirichlet energies** (↓). In Fig. 12 (top), we present both quantities in a single plot to allow a straightforward assessment of which method lies on the Pareto front.

Method	TOPFAUST	$\text{TOPKIDS}_{\text{st}}$	$\text{TOPKIDS}_{\text{ts}}$	SCAPET
ULRSSM [16]	**0.043**	0.089	0.059	**0.022**
SmoothShells [26]	0.071	0.092	0.150	0.161
DiscrOpt [60]	0.102	0.201	0.186	0.049
Hope [27]	0.063	0.100	0.067	0.040
KernelMatch [83]	0.048	0.090	0.064	0.025
Ours	0.045	**0.087**	**0.057**	**0.022**

Method	TOPFAUST	$\text{TOPKIDS}_{\text{st}}$	$\text{TOPKIDS}_{\text{ts}}$	SCAPET
ULRSSM [16]	7.21	3.77	9.41	4.09
SmoothShells [26]	**0.85**	**0.86**	**1.63**	4.11
DiscrOpt [60]	2.91	5.71	11.26	4.08
Hope [27]	56.38	36.49	76.13	45.91
KernelMatch [83]	16.11	8.93	23.98	4.09
Ours	4.51	3.47	8.69	**4.00**

Table 10. Raw results on topologically noisy shapes at a resolution of 10k triangles: (Left) Comparison of **mean geodesic errors** (↓). (Right) Comparison of **mean Dirichlet energies** (↓). In Fig. 12 (bottom), we present both quantities in a single plot to allow a straightforward assessment of which method lies on the Pareto front.