Supplementary Material for Leveraging Intrinsic Properties for Non-Rigid Garment Alignment

Siyou Lin Boyao Zhou Zerong Zheng Hongwen Zhang Yebin Liu
Tsinghua University
Beijing, China
linsy21@mails.tsinghua.edu.cn bzhou22@mail.tsinghua.edu.cn zrzheng1995@foxmail.com
zhanghongwen@tsinghua.edu.cn liuyebin@mail.tsinghua.edu.cn

A1. The *GarmCap* Dataset

Our newly introduced dataset, *GarmCap*, contains 496 high-quality textured 3D garment scans in various poses. As shown in Fig. A1, this dataset includes four different garments: a T-shirt with rich color patterns (G01), a long coat with black-white strip patterns (G02), a thick coat with a fading graywhite texture (G03) and an orange coat (G04). Raw data are triangular meshes of full-body humans wearing these garments in static poses, collected in a cage with 128 cameras. For obtaining garments segmented from full-body scans, we first render each mesh to 128 random views and use SCHP [4] to obtain image-domain segmentation results. Each triangular face is then labeled as garment or non-garment by voting, using their corresponding image-domain labels. We then manually inspect each segmented mesh and fix possibly incorrect boundaries.

As preprocessing, we fit SMPL [6] body models to these meshes using image-domain keypoints plus the VPoser [8] body pose prior. Since texture images are in general of higher frequency than geometry, to obtain per-vertex colors, we first average texture colors over each triangular face, and then transfer these averaged face colors to neighboring vertices. Furthermore, we divide the color of each vertex by its ambient occlusion value to alleviate the effects of shadows.

A2. Implementation Details

In this section, we introduce implementation details of our method and the experimental setup for baselines.

For all neural deformation fields in our pipeline, we use a 5-layer multilayer perceptron (MLP) implementation with ELU [1] activations (except for the output layer). Each hidden layer has 128 neurons. For the number of LBO eigenfunctions, we compute 100 eigenfunctions for each scan mesh and estimate the initial functional map A_0 using all the 100 components. When used as input to the neural deformations and as intrinsic embeddings, we use $K = 50$ to reduce computation cost. For energy weights we choose $w_1 = 1$, $w_2 = 10$, $w_3 = 1000$ in Eq. (3), $w_4 = 1$, $w_5 = 0.05$, $w_6 = 100$ in Eq. (6) and $w_7 = 1$, $w_8 = 0.1$, $w_9 = 0.1$ in Eq. (11). For balancing geometry and color information as described in Section 3.5, we set different β_1 and β_2 values for each garment individually (Table A1) since their brightness can differ greatly. All optimizations are done with Adam [3], with learning rate set to 0.001 for smooth template acquisition and 0.005 for neural deformation fields. We optimize Eq. (3) for 5000 iterations, Eq. (6) for 3000 iterations and Eq. (11) for 300 iterations. For Chamfer distance, we use the PyTorch3D [10] implementation. Note that PyTorch3D uses squared distances when computing the Chamfer distance, which we use only for optimization. For evaluation, we modify its code to use ordinary distances to compute Chamfer distance.

Garment	Stage 1		Stage 2	
	β_1	β_2	β_1	β_2
G01	1	0.25	4	1
G02	1	0.05	4	1
G03	1	0.05	4	0.25
G04	1	0.25	4	1

Table A1. β_1 and β_2 for different garments.

Since the original ClothCap [9] pipeline works on 4D scans while our dataset contains only static poses that are not temporally continuous, we use the posed template T' as initialization to ClothCap [9]. For Neural Deformation Pyramid (NDP) [5] which does not explicitly model human poses, we also use T' as initialization. For Deep Shells [2] which is an unsupervised learning pipeline, we retrain it on our dataset using the official implementation and configurations.

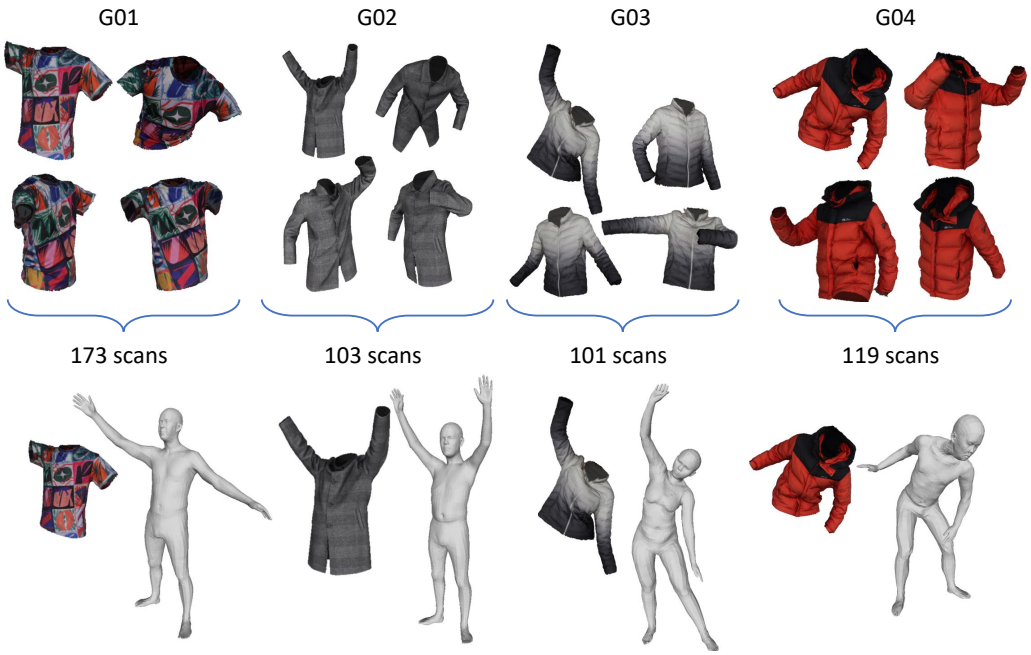

Figure A1. Examples of garment scans in our dataset. We have also fitted a SMPL [6] model for each garment scan.

A3. Why Non-linearity is Necessary

In this section we provide a theoretical analysis of why it is necessary to introduce non-linearity for refining intrinsic embedding alignment. For simplicity, let us consider 1D manifolds, *i.e.*, intervals in $\mathbb{R}$. Suppose $M = N = [0, 1]$ are two 1D shapes to be aligned. The 1D LBO on M (without discretization) is $-\mathrm{d}^2/\mathrm{d}x^2$, whose eigenfunctions with homogeneous Neumann boundary conditions can be analytical computed as (same for N):

$$\phi_k^M(x) = \cos(k\pi x), \quad \lambda_k^M = (\pi k)^2, \tag{A1}$$

where $k = 0, 1, 2, \cdots$.

Suppose now the ground truth correspondence between M and N is a non-isometric one, given by $y = P(x) = x^2$ ($x \in M,\ y \in N$). Let us take $z^M = \phi_1^M(x)$ and $z^N = \phi_1^N(y)$ as their intrinsic embeddings. We now ask: Is it possible to recover the point-to-point map $y = P(x) = x^2$ through some functional map $A(z)$? In other words, does there exist some intrinsic space transformation $z \mapsto A(z)$ such that

$$A(z^M) = z^N ? \tag{A2}$$

In fact, by plugging the expressions for z^M and z^N and $y = x^2$ into $A(z^M) = z^N$, we can find the only A satisfying the above conditions to be:

$$A(\cos(\pi x)) = \cos(\pi x^2) \tag{A3}$$

$$\implies A(z) = \cos(\pi(\cos^{-1}(z)/\pi)^2),\ z \in [-1, 1], \tag{A4}$$

which is clearly non-linear. This 1D example demonstrates that if we only compute a finite number of LBO eigenfunctions for intrinsic embedding, there may not exist a linear functional map that corresponds to a non-isometric deformation. Therefore, non-linearity between intrinsic embeddings must be introduced.

The following theorem shows that these two aspects (namely, the underlying point-to-point map being non-isometric, and using only a small number of eigenfunctions of the LBO) are indeed the root cause for introducing non-linearity.

Theorem 1. *Let M and N be two manifolds with LBO eigenfunctions $\phi_k^M(x)$ and $\phi_k^N(y)$. Suppose each eigenvalue has multiplicity one. Let $\Phi^{M,K}$ denote the $(K+1)$-dimensional intrinsic embedding of M,* i.e.*, we assign to each $x \in M$ the intrinsic coordinates $z^M(x) = [\phi_0^M(x), \phi_1^M(x), \cdots, \phi_K^M(x)] \in \mathbb{R}^{K+1}$. Let $\Phi^{N,K}$ and $z^N(y)$ be similarly defined for N. Suppose $P : M \to N, x \mapsto y$ is a point-to-point map. Then there exists a linear functional map $A : \mathbb{R}^{K+1} \to \mathbb{R}^{K+1}$ such that $A(z^M(x)) = z^N(P(x))$ for all $x \in M$ if one of the following conditions is met:*

1. *P is an isometry.*
2. *K is large enough (possibly ∞) such that*

$$\begin{aligned}&\mathrm{span}\{\phi_0^N(P(x)), \cdots, \phi_K^N(P(x))\} \\ &\subset \mathrm{span}\{\phi_0^M(x), \cdots, \phi_K^M(x)\}.\end{aligned} \tag{A5}$$

Proof. 1. It is well known that LBO eigenfunctions are invariant under isometric transformations (see, *e.g.*, [11]). Therefore, if P is strictly isometric, the LBO eigenfunctions must be in one-to-one correspondence (up to a sign change). The linear functional map A can be chosen as a diagonal matrix with ± 1 entries.

2. By the definition of Eq. (A5), there exist a_{ij} such that

$$\phi_i^N(P(x)) = \sum_{j=0}^{K} a_{ij}\phi_j^M(x), \tag{A6}$$

Then A can be defined as the linear transformation given by the matrix (a_{ij}).

□

While the above theorem does not forbid the existence of a linear functional map when P is non-isometric and K is small, we find such cases to be quite rare in practice. As the theorem suggests, under the linear functional map framework [7], one can generally only work with near-isometric deformations (*e.g.*, as-rigid-as-possible) or compute a large number of eigenfunctions that nearly span the whole function space. However, for garments, the underlying ground-truth correspondence can be highly non-rigid and non-isometric. On the other hand, computing a large number of eigenfunctions is a computationally expensive task. To bypass these obstacles, our work employs a non-linear neural deformation field for intrinsic fitting, which allows accurately recovering the underlying non-isometric point-to-point map with only 50 eigenfunctions.

A4. More Evaluation Results

A4.1. Comparisons

We present more qualitative comparison results in Fig. A2. To visualize the geometric deformation, we apply a colored grid pattern to the source shape and observe its distortion in the fitted results. Note that while ClothCap [9] can produce results that look correct in terms of geometry and texture, the geometric distortion would still be unnatural. Please also watch our supplementary video for 360°-view presentations.

A4.2. Ablation Studies

We present complete quantitative results in Table A2. Since the garments in our dataset have very different characteristics in terms of both geometry and texture, we list the metrics and discuss the effect of different components separately.

For the coarse stage, we evaluate the proposed intrinsic neural deformation field by replacing it with direct vertex optimization (DVO-S1) and a coordinate-based neural deformation field (CoordNet-S1). As Table A2 shows, using intrinsic features to define a deformation field allows us to handle more difficult types of garments. More specifically, this choice leads to superior performance for G02 (the knee-length coat), because we need to rely on the geodesic continuity of intrinsic features to "pull open" the seam at the lower part of G02, while other deformation modules produce unnatural distortions (Fig. A3). Using intrinsic features also allows the coarse stage deformation to be robust to poorly initialized templates. As shown in Fig. A4, an inaccurately estimated SMPL may lead to a self-intersecting template initialization. Our intrinsic deformation network allows pulling apart the intersecting parts and still achieves accurate alignment. Therefore, even though DVO or coordinate-based neural fields can also achieve good results for simpler garments (G01, G03 and G04), utilizing intrinsic features is necessary for more difficult cases as exhibited above.

For the refining stage, Table A2 shows that, compared with linear ICP refinement (LR-S2) [7] or no refinement (NR-S2) in the intrinsic space, our non-linear neural deformation field can lead to better accuracy in both geometry and texture. The improvement in texture alignment over other choices is more obvious for G01 which contains rich color patterns.

For utilizing texture color information, Table A2 shows that not using texture (NoTex) can still lead to high geometric accuracy but with an observable drop in texture accuracy. It is interesting to note that even without using texture colors, our method still outperforms all baseline methods [2, 5, 9]. This suggests that even if our method is used without color information, it can still produce high-quality correspondences.

Please also watch our supplementary video for 360°-view presentations.

References

[1] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv: Learning*, 2015. 1

[2] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, 34, 2020. 1, 3, 4

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[4] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3260–3271, 2022. 1

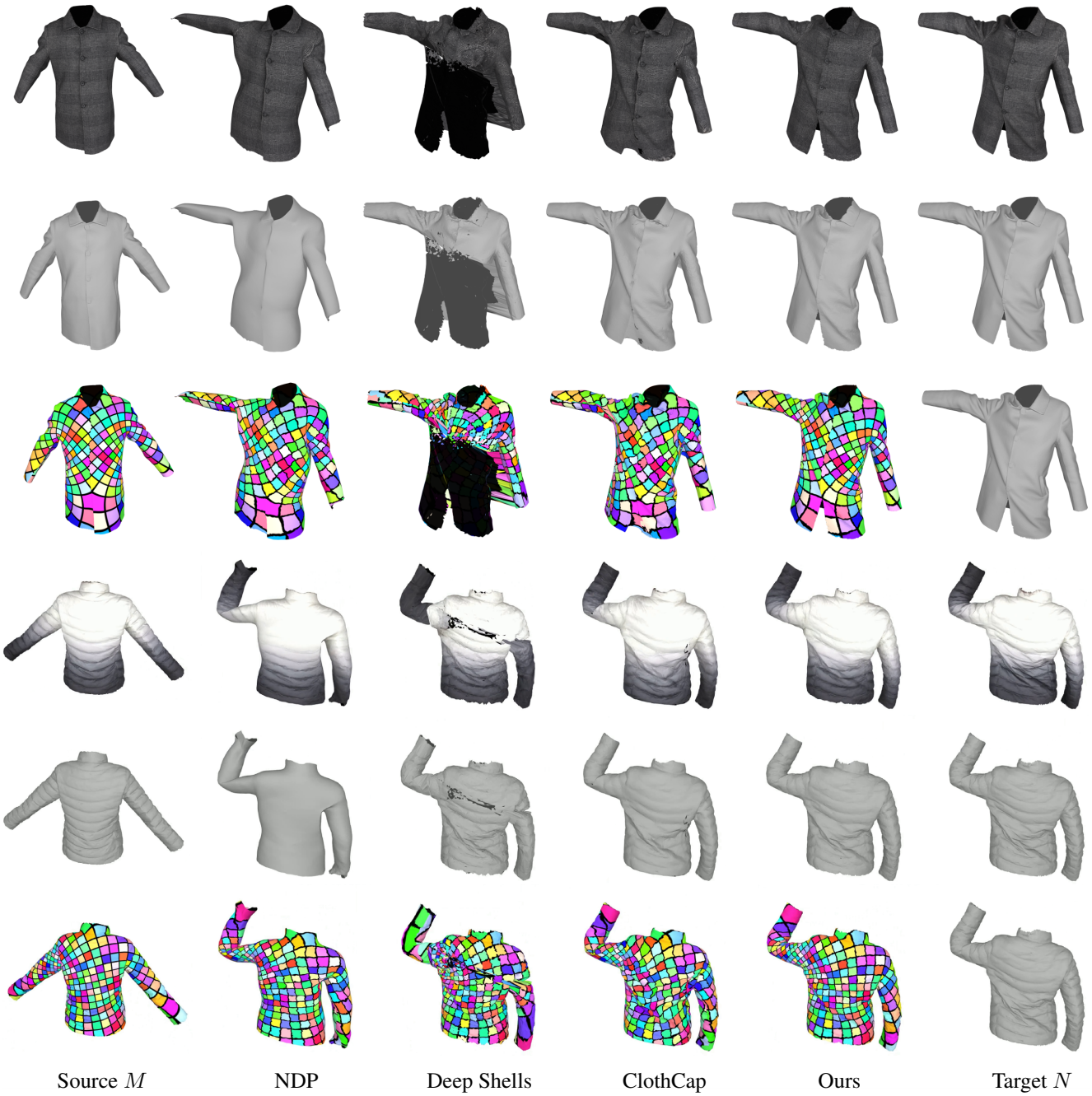

Figure A2. More comparison results with NDP [5], Deep Shells [2] and ClothCap [9]. Aside from texture and geometry, we also exhibit a colored grid pattern to visualize the geometric distortion.

[5] Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. In *Advances in Neural Information Processing Systems*, 2022. 1, 3, 4

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):1–16, 2015. 1, 2

[7] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4), jul 2012. 3

[8] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face,

Garment	Method	Chamfer Dist. ($\times 10^{-3}$) ↓	Cos. Sim. ↑	LPIPS-AlexNet ($\times 10^{-2}$) ↓	LPIPS-VGG ($\times 10^{-2}$) ↓	SSIM ↑
G01	Ours	2.144	**0.985**	2.442	4.195	**0.949**
	DVO-S1	**2.142**	**0.985**	2.459	4.218	0.948
	CoordNet-S1	2.143	**0.985**	**2.437**	**4.183**	**0.949**
	LR-S2	2.172	0.984	2.938	4.850	0.936
	NR-S2	2.186	0.984	3.146	5.131	0.930
	NoTex	2.156	0.984	4.562	6.737	0.904
G02	Ours	**2.780**	**0.988**	**5.396**	**8.866**	**0.824**
	DVO-S1	2.899	0.985	8.327	10.868	0.813
	CoordNet-S1	2.793	0.987	5.785	9.188	0.821
	LR-S2	2.834	0.987	5.681	9.146	0.821
	NR-S2	2.897	0.987	5.928	9.344	0.819
	NoTex	2.783	**0.988**	5.616	9.097	0.821
G03	Ours	2.500	**0.977**	4.161	5.639	0.943
	DVO-S1	2.525	0.976	4.773	6.201	0.938
	CoordNet-S1	2.490	**0.977**	**4.089**	**5.573**	**0.944**
	LR-S2	2.600	0.976	4.322	5.746	0.942
	NR-S2	2.677	0.975	4.393	5.815	0.941
	NoTex	**2.472**	**0.977**	4.276	5.744	0.942
G04	Ours	3.085	0.967	5.274	7.766	0.939
	DVO-S1	3.058	0.967	5.763	8.157	0.935
	CoordNet-S1	**3.039**	**0.968**	**4.938**	**7.502**	**0.942**
	LR-S2	3.131	0.966	5.449	7.854	0.938
	NR-S2	3.138	0.966	5.535	7.938	0.938
	NoTex	**3.039**	**0.968**	5.758	8.060	0.936

Table A2. Quantitative results of ablation studies. We report the metrics for each garment separately since they have very different geometry and texture characteristics. The best results are in boldface and the second best results are underlined.

and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[9] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4), jul 2017. 1, 3, 4

[10] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 1

[11] Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, page 225–233, Goslar, DEU, 2007. Eurographics Association. 3

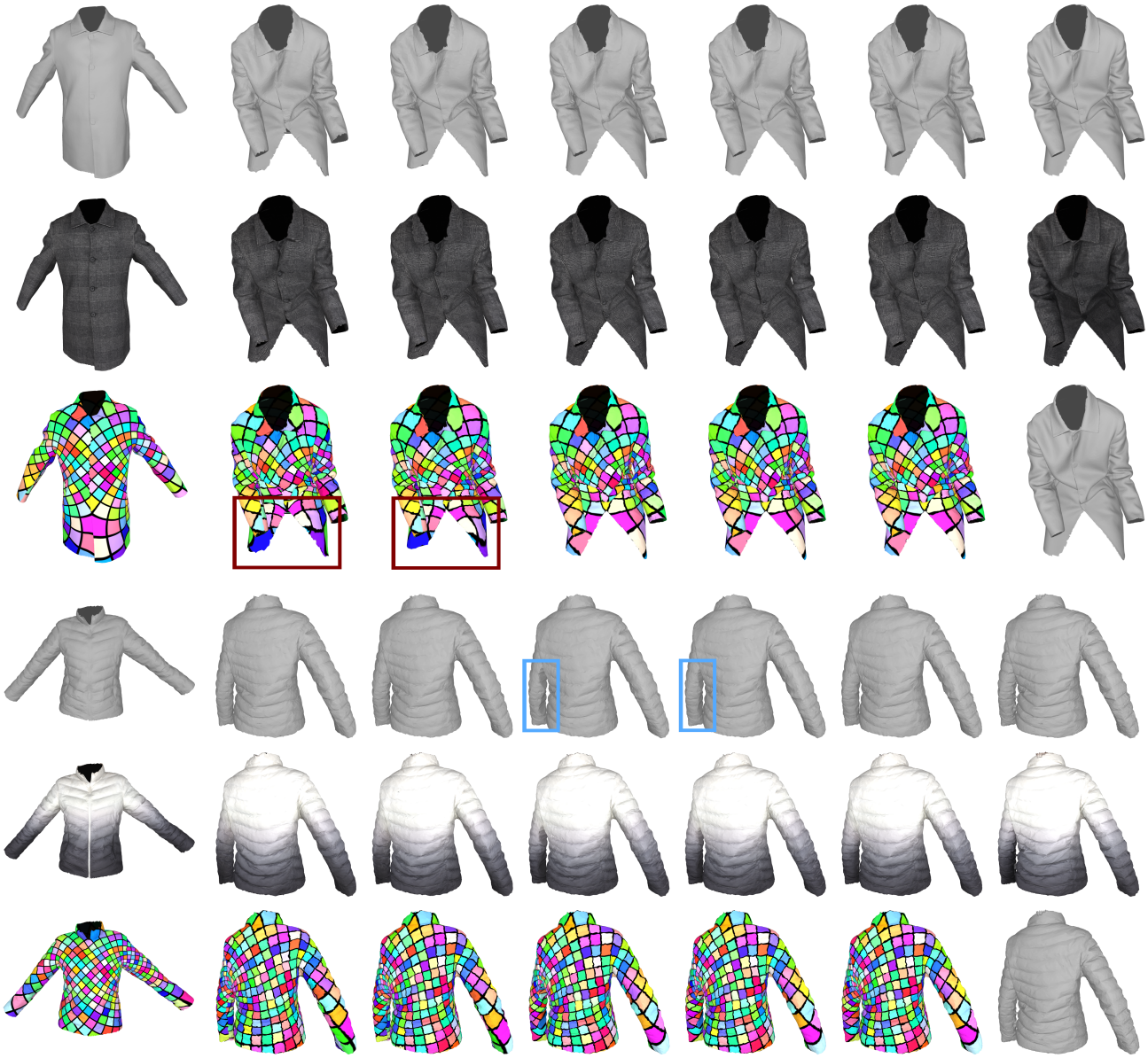

Figure A3. Ablation study on different modules.

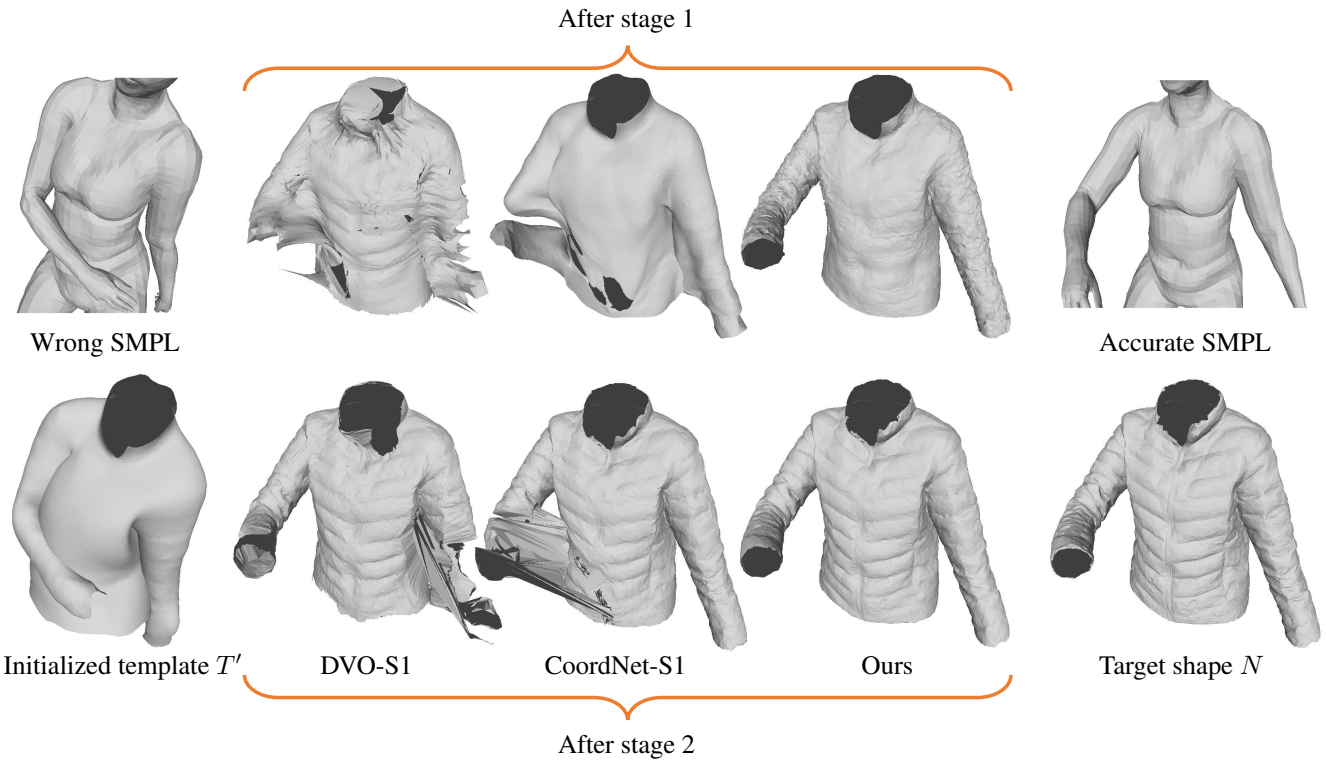

Figure A4. Using an intrinsic deformation network is robust to poor template initializations. Even if the initialization contains self-intersections, our intrinsic deformation network allows pulling away the intersecting parts.