Degrees of Freedom Matter: Inferring Dynamics from Point Trajectories

Supplementary Material

Contents

6. Discussions on the Motion Model Bound

The singular values of the 3D linear transformation matrix carry essential physical meanings. Via singular value decomposition, the motion can be regarded as a consecutive operations of rotating to a new coordinate frame, performing scaling in each dimension based on the singular values, and rotating back to the original coordinate frame. Therefore, the upper bound of such deformation is indicated by the largest singular value.

In the main paper, we demonstrate the representation power of DPF [48] is bounded, based on Eq. (6) and Eq. (7). Here we present more details in terms of a theorem with proof.

Theorem 1. *Provided*

$$\nabla \boldsymbol{u} = \boldsymbol{W}_n \left(\prod_{i=0}^{n-1} \boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x}) \right) \tag{14}$$

and

$$\varphi_i = \cos(\boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{b}_i), \tag{15}$$

in which $\circ$ is the composition of element-wise multiplication and broadcasting a vector to a matrix, as well as $i = \{0, 1, ..., n-1\}$, the bound of the spectral norm of $\nabla \boldsymbol{u}$ is given by

$$\|\nabla \boldsymbol{u}\|_2 \leq d^n \cdot \prod_{i=0}^{n} \|\boldsymbol{W}_i\|_2, \tag{16}$$

in which n and d denote the number of hidden layers and the dimension of hidden layers, respectively.

Proof. Referring to the matrix norm properties [19, 62], we have the following inequalities on the spectral norm, *i.e.*

$$\|\nabla \boldsymbol{u}\|_2 = \left\| \boldsymbol{W}_n \left(\prod_{i=0}^{n-1} \boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x}) \right) \right\|_2 \tag{17}$$

$$\leq \|\boldsymbol{W}_n\|_2 \cdot \prod_{i=0}^{n-1} \|\boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x})\|_2 \tag{18}$$

$$\leq \|\boldsymbol{W}_n\|_2 \left(\prod_{i=0}^{n-1} \|\boldsymbol{W}_i\|_2 \right) \left(\prod_{i=0}^{n-1} \|\hat{\varphi}_i(\boldsymbol{x})\|_2 \right) \tag{19}$$

$$\leq \left(\prod_{i=0}^{n} \|\boldsymbol{W}_i\|_2 \right) \left(\prod_{i=0}^{n-1} \|\hat{\varphi}_i(\boldsymbol{x})\|_2 \right), \tag{20}$$

in which $\hat{\varphi}_i$ is the matrix with the same column φ_i, and has the shape of $\mathbb{R}^{d \times q}$. This corresponds to the shape of $\boldsymbol{W}_i$,

and hence it has $q = 3$ at the input layer and $q = d$ in the hidden layers. Therefore, we assume $q = d$ in the following derivations to obtain the upper bound.

Note the rank of the matrix $\hat{\varphi}_i$ is 1, and we can have

$$\|\hat{\varphi}_i(\boldsymbol{x})\|_2 = \|\hat{\varphi}_i(\boldsymbol{x})\|_F = \sqrt{d} \cdot \|\varphi_i(\boldsymbol{x})\|_2 \leq d, \tag{21}$$

according to $\|\varphi_i\|_\infty \leq 1$. Thus, we can derive

$$\|\nabla \boldsymbol{u}\|_2 \leq d^n \cdot \prod_{i=0}^{n} \|\boldsymbol{W}_i\|_2. \tag{22}$$

□

Although the constant factor d^n is large, it is only reached when every $|\varphi_i|$ is equal to 1, which is implausible in practice. In addition, the entries in $\boldsymbol{W}_i$ are from the uniform distribution with a tiny range around 0 [53], which further constrains the spectral norm of the Jacobian matrix. Due to the challenges of spectral analysis on high-dimensional random matrices, we can look into the degenerated 1D case, which is given by

$$\frac{du}{dx} = w_n \prod_{i=0}^{n-1} w_i \cos(w_i x + b_i). \tag{23}$$

In this case, we can easily derive

$$\left|\frac{du}{dx}\right| = \left| w_n \prod_{i=0}^{n-1} w_i \cos(w_i x + b_i) \right| \leq \prod_{i=0}^{n} |w_i|, \tag{24}$$

which indicates that the motion complexity is heavily bounded.

A statistical perspective. The boundedness can be also investigated from a statistical perspective. Starting with

$$\|\nabla \boldsymbol{u}\|_2 \leq \|\boldsymbol{W}_n\|_2 \cdot \prod_{i=0}^{n-1} \|\boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x})\|_2 \tag{25}$$

that is from Eq. (17), we can reason the entries of $\boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x})$ are converging to the standard normal distribution if the model weights are initialized as in SIREN [53]. Specifically, the entries of $\boldsymbol{W}_i$ are from the defined uniform distribution, φ_i is from the arcsine distribution, since the cosine activation function is equivalent to the phase-shifted sine activation function and the bias does not modify the distribution for high enough frequency [53, Theorem 1.8]. According to [7, Theorem 2.5], the largest singular value of $\boldsymbol{A}_i = \boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x})$ is bounded, having

$$\lim_{d \to \infty} \sup \lambda_1(d^{-1} \boldsymbol{A}_i \boldsymbol{A}_i^T) \leq 4, \tag{26}$$

in which d is the hidden dimension and λ_1 is the largest eigenvalue. Therefore, their compositions with $i = 0, \ldots, n-1$ are also bounded.

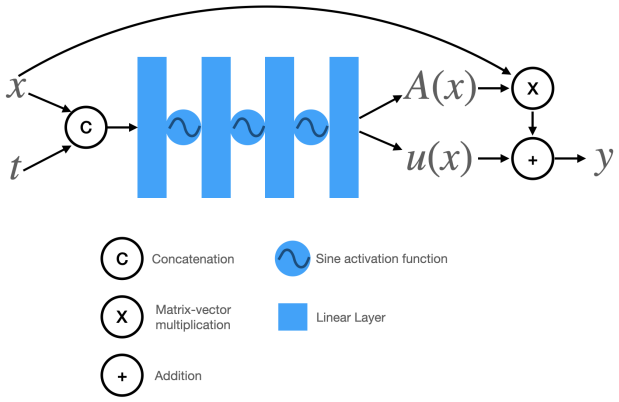

Figure A1. Illustration of the DOMA model architecture. The SIREN layers [53] produce an affine transformation, which maps the point from $\boldsymbol{x}$ to $\boldsymbol{y}$ at time t.

DPF	*-Trans*	*-SE(3)*	*-Scaled SE(3)*	*-Affinity*
$(6d + nd^2)(T-1)$	$7d + nd^2$	$13d + nd^2$	$14d + nd^2$	$16d + nd^2$

Table A1. The number of parameters in the employed SIREN network. T, n, d denote the number of frames in the sequence, the number of network hidden layers, and the hidden dimension, respectively. The suffixes denote different versions of DOMA.

7. Additional Discussions on DOMA

7.1. The Network

The model architecture. The DOMA models can be visualized in Fig. A1. In this case, the SIREN [53] network contains one input layer, one output layer, and two hidden layers. In the case of the 'SE(3)' and 'scaled SE(3)' models, 6D continuous rotation representations [72] are produced by the output layer, which are then orthogonalized to rotation matrices.

The model sizes. Since the 1D temporal dimension is incorporated in the input layer, DOMA models have $\mathcal{O}(1)$ complexity w.r.t. the motion sequence length. The sizes of different models are summarized in Tab. A1.

7.2. Additional Discussions on Novelties

Modeling the deformation field or the motion field is not a new task. Instead, various methods have been developed within respective tasks, such as geometry deformation, neural rendering, dynamic scene reconstruction, avatar creation, etc. Their exploited motion methods are diverse in terms of the neural architecture, positional encoding, underlying deformation models, and so on. However, an important aspect is often overlooked: the motion field should be spatiotemporally regularized by nature. To fill this gap, we leverage the SIREN [54] network, and extend the start-

of-the-art work DPF [48] to a multi-frame smooth affinity field model. By introducing additional DOFs at the output layer, we find the model representation power is improved in a different way from enlarging the model hidden layers, and come up with a solution to increase the model capacity while retaining the model size. Moreover, we introduce a smoothness regularization term to overcome overfitting, which does not assume the underlying motion is *e.g*. rigid like in [44]. The effectiveness of DOMA is demonstrated with experiments in Sec. 4 and the supp. mat.

The advantage of DOMA is more obvious when the ground truth motion is more complex. An example is modeling the loose long skirt motion. As shown in Tab. A8, DOMA-Affinity is consistently superior to DPF on the 'felice' sequences of Resynth. Another example is modeling the fluid dynamics, which is investigated in Sec. 9.2. We can see DOMA-Affinity outperforms DPF significantly. Since DPF only models deformations between the canonical frame and frame t, it cannot ensure the temporal smoothness between t and $t+1$.

Despite aiming at different tasks, our work is also related to object shape and view recovery from images. Kanazawa *et al*. [22] propose a framework to learn from an annotated image collection, and recover the 3D shape in a canonical frame, the camera pose, and the texture of an object from a single image. The 3D object shape is parameterized by a learned mean shape and per-instance predicted deformation. To encourage additional properties such as surface smoothness and regularized deformation, generic priors are leveraged in the training loss. Goel *et al*. [16] extend this framework to learn from an image collection without annotations of the keypoints and the camera. To further improve the performance, Gharaee *et al*. [15] propose to predict a set of keypoints to represent the shape, corresponding to positions on the category-specific mean shape in 3D. Afterwards, the camera pose is estimated by a robust PnP network [6]. These solutions of decoupling the instance-level shape into the mean shape and the deformation also inspire us how to model motions. Furthermore, we are encouraged by these works to reconstruct dynamic scenes from multi-view videos as future work.

8. Experiment Details

8.1. Additional Presentations on Point Motion Prediction (Sec. 4.1)

We leverage and modify the codebase of ResFields [37] for the implementations of baselines MLP-ReLU and DCT-NeRF [58].

bear3EP_Agression
demon_JazzDancing
dragonOLO_act25
michelle_StepHipHopDance
mutant_Defeated
tigerD8H_Swim17
vampire_Breakdance1990
vanguard_JoyfulJump

Table A2. The leveraged DeformingThings4D [27] sequences in Sec. 4.1.

8.1.1 Dataset

The 7 sequences from DeformingThings4D [27] are listed in Tab. A2. For each sequence, we extract the first 100 frames and regard the first frame as the canonical frame.

8.1.2 Baselines

MLP-ReLU and MLP-ReLU PE.6. MLPs with ReLU [24] are frequently used to warp points in existing works. In our experiment, the architecture contains 6 hidden layers of 128 hidden dimensions. In Tab. 1, the Fourier positional encoding [38] is not used in 'MLP-ReLU', but is applied in 'MLP-ReLU PE.6' with 6-level resolutions.

DCT-NeRF [58]. The Fourier positional encoding is not applied. Rather than outputting the target point location $\boldsymbol{y}$, this baseline method produces the coefficients of a DCT basis that is jointly learned from the data. Similar technology is also employed in [28].

BANMO [66]. BANMO is a solution to reconstruct the avatar of a generic object, *e.g*. cat, from a monocular video. The avatar bones are modelled by a set of 3D Gaussians, and the skinning weight is a combination of a Gaussian-based weighting function and a neural network. The 3D location of a query point is encoded by Fourier encoding [38]. The rest pose code is derived by a linear layer, and the pose code is derived by the Fourier encoding of the frame and a linear layer. In our experiment, we adopt its avatar deformation module into our setting and use the hyper-parameters as in the original paper [66]. Provided a set of training point trajectories, we optimize the 3D Gaussians and the relevant networks as in [66]. During testing, we animate the testing points in the canonical frame to produce the trajectories, based on the learned Gaussians and networks.

BoneCloud. Based on BANMO [66] and KeyTr [41], we propose this BoneCloud method, which is a learnable bone basis. Compared to BANMO, this BoneCloud method does

not employ any nonlinear neural network. Instead, it has a point cloud in the canonical frame, and each point stores a time sequence of SE(3) transformations. The skinning weights are created by a pre-fixed radial basis function. Consequently, a 3D point $\boldsymbol{x}$ in the canonical frame can be transformed to $\boldsymbol{y}$ at frame t, via linear blend skinning. Specifically, it is given by

$$\begin{pmatrix} \boldsymbol{y} \\ 1 \end{pmatrix} = \left(\sum_k w_k \boldsymbol{T}_k^t \right) \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \tag{27}$$

and

$$\tilde{w}_k = \exp\left(-\sigma \|\boldsymbol{x} - \boldsymbol{v}_k\|_2\right) \tag{28}$$

$$w_k = \frac{\tilde{w}_k}{\sum_k \tilde{w}_k}, \tag{29}$$

in which $\boldsymbol{v} \in \mathbb{R}^3$ is a bone in the bone cloud, k is the index of the bone, $\boldsymbol{T}_k^t \in SE(3)$ denotes the transformation of the bone k at time t. In our experiment, we leverage 1024 points as bones. During training, we leverage the provided point trajectories to optimize the bone locations at the canonical frame and the bone transformations at individual time steps. During testing, we transform the testing points in the canonical frame to individual target frames, so as to produce the point trajectories.

8.1.3 More Results on the Synthetic Dataset

Corresponding to Fig. 2 in the main paper, we show the qualitative results of all DOMA variants in Fig. A2. We can see that the affinity field is able to represent all explored linear transformations. This indicates the output layer highly influences the motion types that the model can represent.

In order to investigate how the hidden dimension influences the DOF representation, we increase the hidden dimension of DOMA-Trans from 128 to 256. The results are shown in Tab. A4. Without smoothness regularization, we can see that a higher hidden dimension slightly improves the performance in some cases, but degrades the performance on translation, probably due to overfitting. When applying the smoothness regularization to overcome overfitting, motion prediction on translation is significantly improved, whereas the performances on other linear transformations are much worse. On the other hand, the performances of DOMA-Affinity on all motion types are consistently and considerably improved by the smoothness regularization. Based on these observations, we can conclude that

- Both the hidden dimension and the DOFs represented by $\boldsymbol{A}$ can influence the model representation power.
- Increasing the hidden dimension improves the performance but not always. Overfitting could occur.
- The smoothness regularization can improve the performance significantly if the ground truth DOF is explicitly modeled at the output layer. Otherwise, it can degrade the performance.
- Increasing the hidden dimension cannot simply increase the DOF representations. Otherwise, the smoothness regularization should lead to consistent improvements for all linear transformations.

Methods	Rotation	Scaling	Shearing	Translation
-Trans	2725.4	1817.8	1619.5	1042.4
-SE(3)	730.6	1991.4	1138.3	899.4
-Scaled SE(3)	801.1	685.8	1524.7	1096.2
-Affinity	1486.0	915.4	622.1	822.4
-Trans-E	38.0	1669.6	753.6	38.8
-SE(3)-E	20.0	1761.3	832.7	26.4
-scaled SE(3)-E	21.2	1161.8	961.1	24.0
-Affinity-E	19.2	155.7	864.0	15.7
-Trans-H	4919.9	2056.4	2446.8	37.8
-SE(3)-H	52.4	2012.4	1665.0	36.9
-scaled SE(3)-H	29.3	22.1	688.0	30.3
-Affinity-H	**5.4**	**26.3**	**8.5**	**28.8**

Table A3. Results on Synthetic sequences w.r.t. EPE (in $\times 10^{-4}$). '-E' denotes the elasticity loss proposed in Nerfies [44], and '-H' denotes our smoothness loss. Best results are in boldface.

Methods	Rotation	Scaling	Shearing	Translation
-Trans-128d	2725.4	1817.8	1619.5	1042.4
-Trans-256d	2187.1	1846.9	1515.8	1231.8
-Affinity-128d	1486.0	915.4	622.1	822.4
-Trans-H-128d(0.1)	4919.9	2056.4	2446.8	37.8
-Trans-H-256d(0.1)	4911.8	2078.7	1733.7	217.4
-Trans-H-256d(1)	10945.3	8400.4	8701.2	**16.2**
-Affinity-H-128d(0.1)	**5.4**	**26.3**	**8.5**	28.8

Table A4. Results on Synthetic sequences as in Tab. A3 in the main paper. Numbers denote EPE in $\times 10^{-4}$. '-128d' and '-256d' denote the hidden dimension of the SIREN network. The number in () denotes the weight of the smoothness loss term. Best results are in boldface.

Runtime analysis. In addition, we compare our derived analytical gradients with auto-diff of Pytorch [45] w.r.t. the runtime. We set the smoothness loss weight to 0.1, and train DOMA-Affinity for 1000 iterations. This experiment is conducted with Ubuntu 20.04, NVIDIA TITAN RTX 24GB, CUDA 11.4, 32GB RAM. The results are shown in Tab. A5. We can see the analytical gradients improve the efficiency consistently. Compared to the standard auto-diff, the runtime is reduced by 28%.

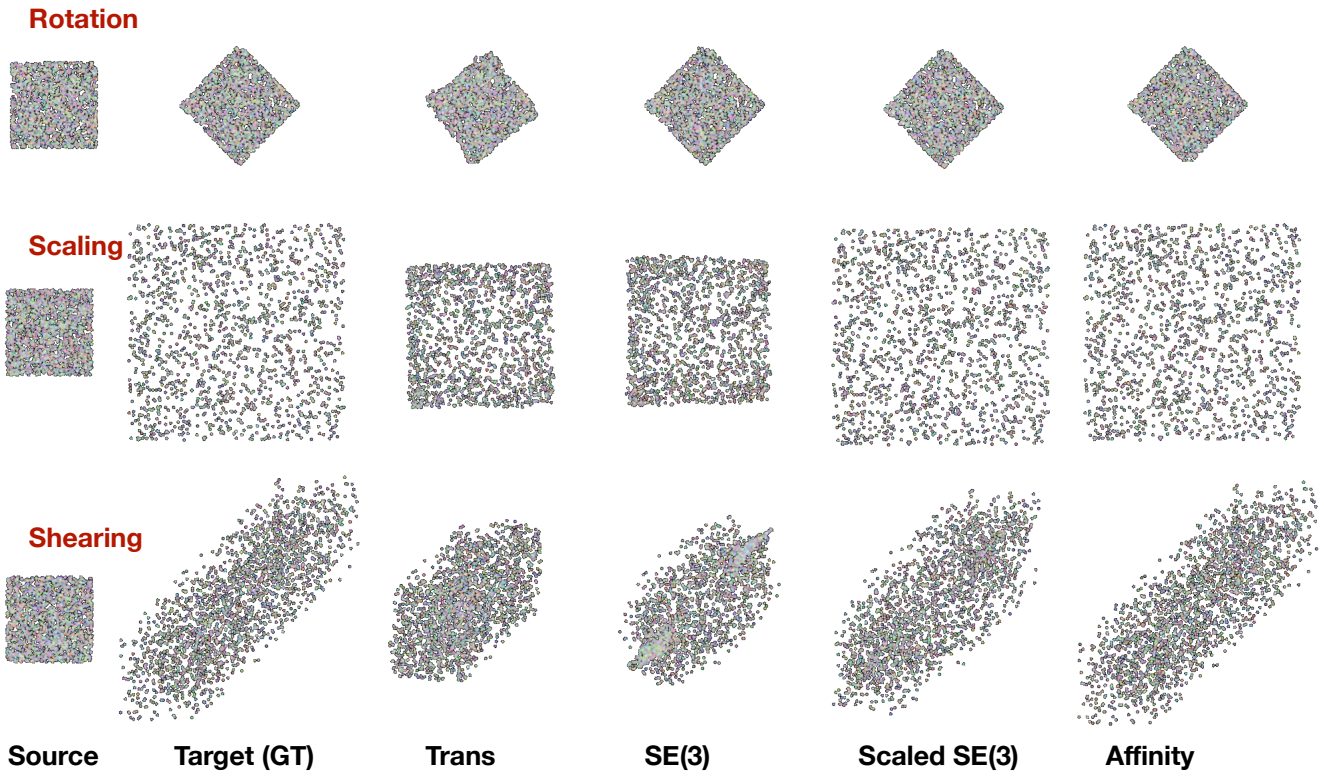

Figure A2. Illustrations of results on the Synthetic sequences. The smoothness regularization is applied. Rows show types of motions, and columns show the testing points in the canonical frame, a target frame, and estimated results from different methods, respectively.

Methods	Rotation	Scaling	Shearing	Translation	average
auto-diff [45]	133.68	132.36	133.44	133.51	133.25
analytical grad	96.08	96.26	95.78	95.92	96.01

Table A5. Comparison between our derived analytical gradients and auto-diff of Pytorch. Runtime is measured in seconds.

8.2. Additional Presentations on Guided Mesh Alignment (Sec. 4.2)

8.2.1 Dataset

We employ the ReSynth dataset [31, 32] in this study. Specifically, we choose 16 sequences from 4 subjects in the *packed* sequences in the *test* split (see Tab. A6). For each sequence, we first perform down-sampling by every 2 frames, and then select the first 30 frames for experiments. The first frame in each sequence is regarded as the canonical frame.

The motion complexity depends on the subject and the clothing type. As shown in Fig. A3, sequences with 'rp_felice_posed_004' are more complex than others, because of the loose long skirt. In this case, the points on the long skirt are far away from the body surface, which is aligned and guided by the SMPL-X [46] mesh vertices. Other subjects have tight clothes.

Subjects	Actions
rp_aaron_posed_002	96_jerseyshort_hips 96_jerseyshort_squats 96_longshort_flying_eagle 96_longshort_tilt_twist_left
rp_celina_posed_005	96_jerseyshort_hips 96_jerseyshort_squats 96_longshort_flying_eagle 96_longshort_tilt_twist_left
rp_felice_posed_004	96_jerseyshort_hips 96_jerseyshort_squats 96_longshort_flying_eagle 96_longshort_tilt_twist_left
rp_janna_posed_032	96_jerseyshort_hips 96_jerseyshort_squats 96_longshort_flying_eagle 96_longshort_tilt_twist_left

Table A6. Employed Resynth sequences in Sec. 4.2.

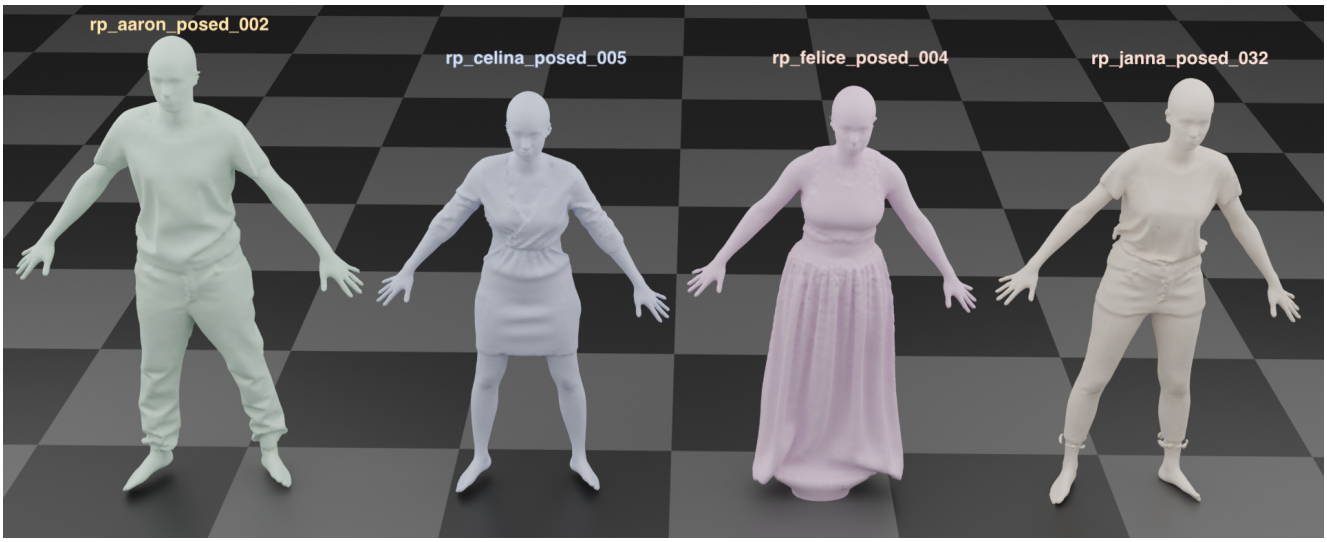

Figure A3. Illustrations of the 4 subjects in our employed sequences. These meshes are created by Poisson surface reconstruction based on the provided oriented points in the canonical frames. We have created 16 such meshes for individual sequences, which are roughly at the same pose, *i.e*. the A-pose.

8.2.2 Performances of All Model Variants

In Tab. 3, we only show the performance of the DPF baseline, DOMA-Trans, and DOMA-Affinity. Here we show the performances of all DOMA variants under the same experiment setting. The results are presented in Tab. A7. We can draw similar conclusions as in Sec. 4.2. The performance of the affinity field is similarly better than other variants, in particular on the Chamfer distances.

8.2.3 Analysis on Clothing Types

In addition to the averaged performance on all sequences, we have also observed consistent trends on individual subjects that have different clothing types. In this experiment, we leave 'rp_felice_posed_004' out of others, and perform evaluations separately. For compactness, we only show the comparison between our proposed affinity field and the frame-wise DPF models [48], with the weights of the AIAP loss term and the motion smoothness term being $(1, 0.001)$. The results are shown in Tab. A8. We can see that the affinity field outperforms DPF [48] on the subject with the long skirt, whereas performs worse on subjects with tight clothing. A probable reason is that points that are close to the body surface can be effectively guided by the SMPL-X mesh vertices. Due to much more model parameters, the frame-wise DPF models can overfit to the guidance points, and hence produces better results on the body surfaces and the tight clothes. Simultaneously, it produces more artifacts and discontinuities at regions that are far away from the guidance points, leading to inferior performance to the affinity field.

	$\mathcal{L}_{CD} \downarrow$	$\mathcal{L}_n \downarrow$	*STD(E)*↓	*STD(V)*↓
DPF [48]	1.149	**0.122**	**11.6**	24.6
-Trans	1.230	0.128	12.8	22.9
-SE(3)	1.343	0.134	16.2	22.9
-Scaled SE(3)	1.273	0.127	16.2	22.8
-Affinity	**1.142**	0.125	11.9	**22.8**
DPF-A [48]	1.166	**0.119**	**10.3**	24.2
-Trans-A	1.195	0.123	10.4	**23.0**
-SE(3)-A	1.278	0.123	11.5	**23.0**
-Scaled SE(3)-A	1.20	0.120	11.3	23.0
-Affinity-A	**1.151**	0.122	10.6	**23.0**
DPF-H [48]	1.142	**0.123**	10.3	24.2
-Trans-H	1.207	0.128	10.8	**22.9**
-SE(3)-H	1.230	0.127	12.2	23.0
-Scaled SE(3)-H	1.189	0.125	11.5	22.9
-Affinity-H	**1.127**	0.127	**10.1**	**22.9**
DPF-AH [48]	1.189	**0.120**	9.3	24.3
-Trans-AH	1.240	0.124	9.3	**23.0**
-SE(3)-AH	1.265	0.124	10.9	23.1
-Scaled SE(3)-AH	1.255	0.126	8.7	23.0
-Affinity-AH	**1.187**	0.124	**8.9**	**23.0**

Table A7. Results of guided mesh alignment on our selected Resynth sequences. $\mathcal{L}_{CD}$ is in $\times 10^{-4}$. *STD(E)* and *STD(V)* are given in millimeters. This table is supplementary to Tab. 3 in the main paper.

8.2.4 Influence of Hidden Dimensions

We set the hidden dimension to 128 by default in the main paper and the above experiments. Here we increase it to 256 and re-evaluate the performances. According to our analysis of the motion model bound (see Sec. 3 and 6), increasing the hidden dimension is able to improve the model

Subjects	Methods	*CD*↓	*CDN*↓	*STD(E)*↓	*STD(V)*↓
rp_felice_posed_004	DPF-AH [48]	3.086	0.194	**14.4**	25.8
	-Affinity-AH	**2.857**	**0.190**	15.3	**21.9**
others	DPF-AH [48]	**0.557**	**0.096**	7.6	23.8
	-Affinity-AH	0.630	0.102	**6.8**	**23.3**

Table A8. Evaluation of methods on the Resynth sequence 'rp_felice_posed_004' as discussed in Section 4.2.

Subjects	Methods	*CD*↓	*CDN*↓	*STD(E)*↓	*STD(V)*↓
all sequences	DPF-AH [48]	1.047	**0.100**	10.1	24.2
	-Trans-AH	1.058	0.106	10.6	**23.2**
	-SE(3)-AH	1.055	0.105	10.9	**23.2**
	-Scaled SE(3)-AH	1.060	0.104	10.3	**23.2**
	-Affinity-AH	**1.023**	0.105	**9.8**	**23.2**
rp_felice_posed_004	DPF-AH [48]	2.740	0.151	**15.6**	25.3
	-Trans-AH	2.724	0.157	16.0	22.6
	-SE(3)-AH	2.683	0.149	18.5	**22.5**
	-Scaled SE(3)-AH	2.694	**0.148**	17.5	22.6
	-Affinity-AH	**2.544**	0.152	16.5	**22.5**
others	DPF-AH [48]	**0.483**	**0.083**	8.3	23.8
	-Trans-AH	0.503	0.089	8.8	**23.4**
	-SE(3)-AH	0.513	0.090	8.4	23.5
	-Scaled SE(3)-AH	0.516	0.089	7.9	23.5
	-Affinity-AH	0.515	0.090	**7.6**	23.5

Table A9. Evaluations based on the models with 256D hidden variables. Other settings are identical with Tab. 3 and A8. Best results are highlighted in boldface.

representation power on the motion complexity.

The results are presented in Tab. A9. Compared to models with 128D hidden variables (see Tab. 3 and Tab. A8), models with 256D hidden variables consistently produce better results. With this new setting, the performance gaps between individual methods tend to vanish. The temporal smoothness tends to degrade though.

In the meanwhile, we can see that the affinity field still has comparably better performance than frame-wise DPF [48], but produces smoother results, leading to the same observation and conclusion as demonstrated in Sec. 4.2. Focusing on the performances on different sequences, we can see DPF [48] still outperforms DOMA models on tight clothing w.r.t. alignment, but the gap becomes smaller compared to Tab. 3. The affinity field outperforms DPF on the loose long skirt sequence by a large margin. Furthermore, from the model size perspective, our DOMA models are still significantly more lightweight than the DPF [48] baseline.

9. Additional Experiments

9.1. Learning 2D Image Deformation

Similar to the experiments on the 3D synthetic dataset, here we conduct an experiment on 2D image deformation, in order to further investigate the representation power of DOMA models.

Data, evaluation, and methods. We use a RGB image of cat that has 512×512 of pixels. As our synthetic dataset, we perform translation, rotation, scaling, and shearing in the 2D domain, and produce 30 frames. In each case, we randomly choose 25% points for training the motion field, and use the remaining for testing. The evaluation metric is the same as in Sec. 4.1. DOMA-Trans and DOMA-Affinity with different hidden dimensions are applied in this experiment. No regularization is used during training.

Results. The quantitative evaluation is shown in Tab. A10, and some qualitative results are shown in Fig. A4. We can see that the affinity model outperforms the translation model consistently with different hidden dimensions. In particular, the performance of the affinity model is superior when the hidden dimension is smaller. These results demonstrate the advantages of additional DOFs.

9.2. Inferring Dynamics of Fluid Fields

In Sec. 4.1, we have investigated the model representation power based on the DeformingThings4D [27] sequences. Despite various model shapes and movements, they are limited to elastic deformations of solid objects. In this section, we propose a more challenging scenario, modeling a fluid field. To perform empirical studies, we follow [1] to simulate how liquid moves in a bounded field with Unity3D, and record the particle trajectories (see Fig. A5).

The entire sequence contains 931 frames and 27,000 particles. We randomly choose 50% for training the motion field and use the rests for testing. We find that all methods investigated in this paper are not able to reconstruct the entire sequence. Thus, we down-sample the entire sequence by every 2 frames, and then trim the down-sampled sequence into 10-frame clips. Specifically, the frame indices of the clips are $\{(t, t+10)\}_{t=10,15,\ldots,325}$, in which the sequences with trivial motions, *e.g.* static state in the beginning and steady state in the end, are excluded. This pre-processing will lead to 22 clips in total.

In each clip, the first frame is regarded as the canonical frame. Points in the canonical frame are transformed into individual target frames, and their averaged L1 distances to the ground truth are minimized during training. The evaluation metric is identical to Sec. 4.1.

In this experiment, we compare frame-wise DPF [48], DOMA-Trans, and DOMA-Affinity. These two DOMA models have 128D hidden variables and 2 hidden layers. To avoid overfitting, the frame-wise DPF models have 2 hidden layers and 105D hidden variables, in order to keep most sequences having comparable training errors. Results are

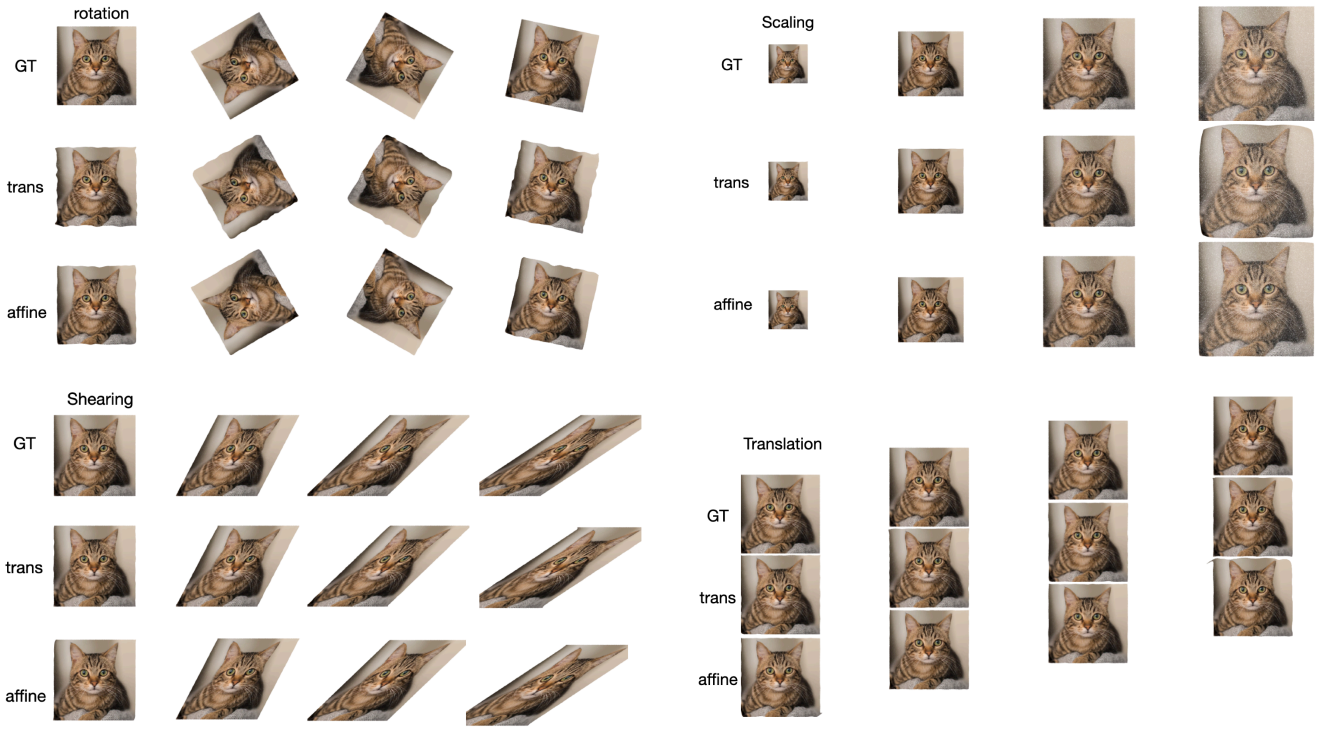

Figure A4. Illustration of modeling 2D image deformation, in which the hidden dimension is 64. 'GT', 'trans', and 'affine' denote the ground truth, DOMA-Trans, and DOMA-Affinity, respectively.

	Rotation		Scaling		Shearing		Translation	
	-Trans	-Affinity	-Trans	-Affinity	-Trans	-Affinity	-Trans	-Affinity
hdim=32	60.8	85.0	31.2	8.9	11.7	5.4	100.4	44.0
hdim=64	64.5	33.9	15.3	6.2	10.5	4.3	25.8	18.7
hdim=128	35.4	20.3	8.2	4.6	6.4	3.8	20.2	20.4

Table A10. Evaluations in the 2D image deformations. As in Tab. A3, the numbers denote EPE in $\times 10^{-4}$, and are the lower the better.

presented in Tab. A11. We can see that the DOMA models considerably outperform the DPF baseline. In addition, the affinity field model performs comparably better than the translation field model. Together with the experiments in Sec. 4.1, we can conclude that DOMA models are superior to the frame-wise DPF method [48], as well as other state-of-the-art baselines. Fig. A6 illustrates some examples of how these methods perform. We can see that the frame-wise DPF method can lead to significant discontinuities between frames, and less accurate motion prediction than DOMA.

Methods	*Scene Flow Error*↓
DPF [48]	412.57
DOMA-Trans	201.66
DOMA-Affinity	**182.46**

Table A11. Motion prediction of unseen points on fluid simulation sequences. The numbers EPEs in $\times 10^{-4}$. Best results are in boldface.

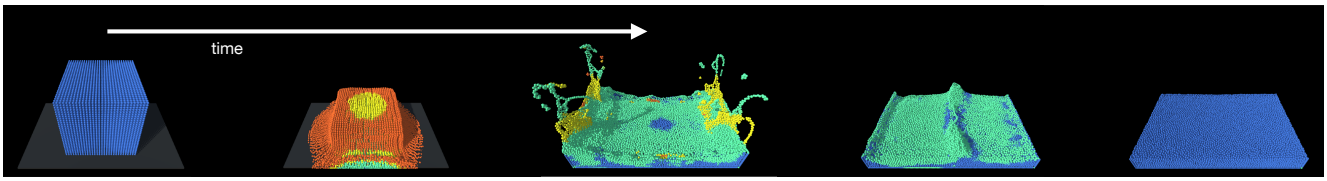

Figure A5. The particle system to simulate a fluid field in Unity3D, which is implemented based on [1].

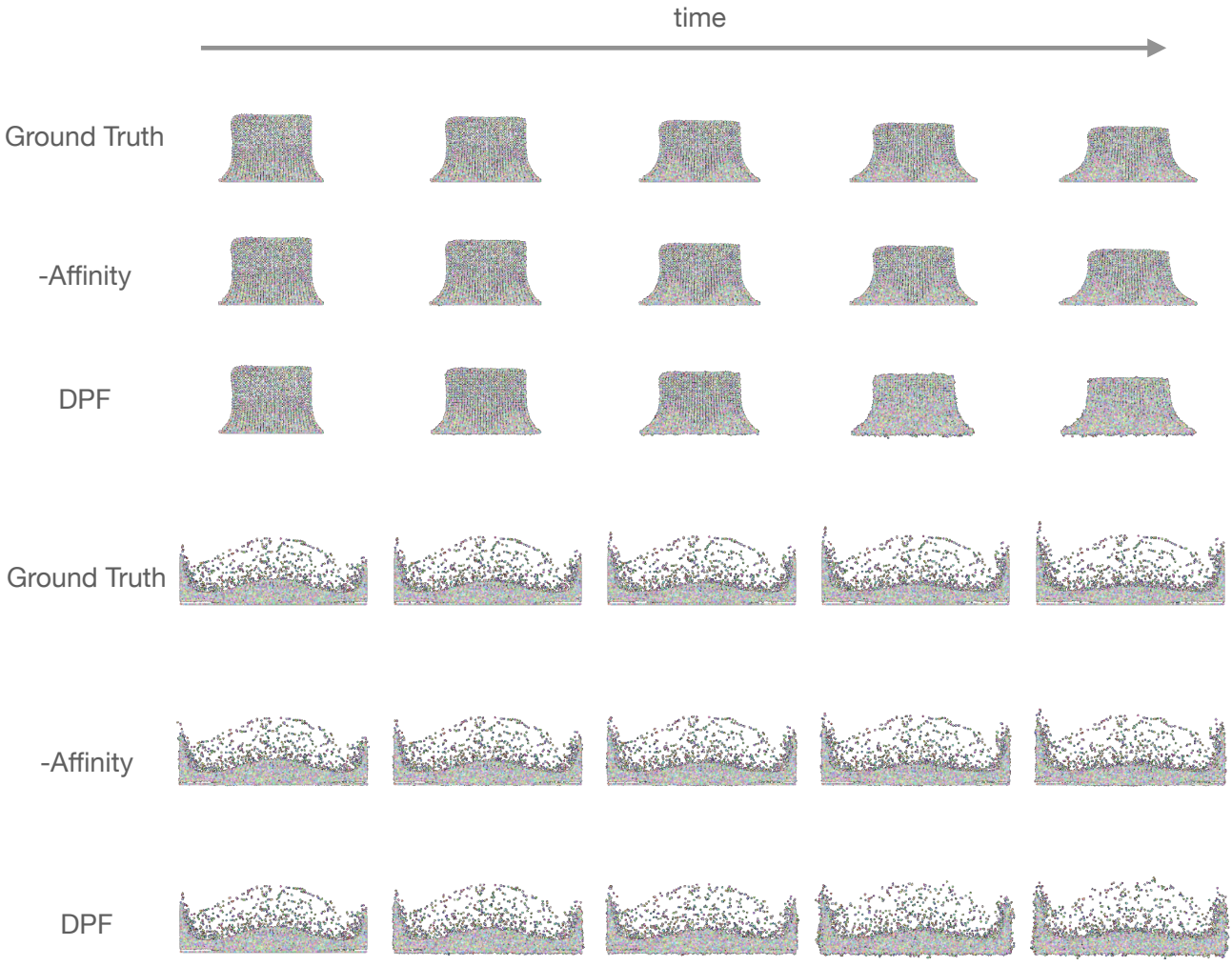

Figure A6. Some results of particle motion prediction in the simulated fluid field. The scene is rendered from the front view.