Supplementary Material for AutoAvatar: Autoregressive Neural Fields for Dynamic Avatar Modeling

Ziqian Bai[1,2*] Timur Bagautdinov[2] Javier Romero[2] Michael Zollhöfer[2]
Ping Tan[1] Shunsuke Saito[2]

[1]Simon Fraser University [2]Reality Labs Research

We provide additional implementation details and evaluations in this supplementary material, including network architectures (Sec. A), analysis on input pose accuracy (Sec. B), comparison between our SDF decoding based on k-NN and the closest surface projection approach proposed in Neural Actor [3] (Sec. C), ablations on number of used history frames (Sec. D) and vertex subsampling (Sec. E), time and memory cost (Sec. F), an experiment for a clothed human (Sec. G), and other limitations (Sec. H and Sec. I). Please refer to the supplementary video for qualitative comparisons and animation results.

A Network Architectures

In our experiments, we use a UV map of resolution 256×256, $T = 3$, and $k = 20$. Before the k-nearest neighbor (k-NN) query in Sec. 3.3, we subsample 3928 vertices by poisson-disk sampling on the SMPL mesh, and only use these subsampled vertices for k-NN computation. This subsampling ensures that vertices are distributed uniformly, leading to consistent area coverage by k-NN selection (see Supp. Mat. for more discussions). Before being fed into the UNet, $L(\boldsymbol{p}_{t+1})$ and $\{L(\dot{\boldsymbol{p}}_{t+i})\}$ are compressed to 32 channels using 1×1 convolutions. The UNet uses convolution and transposed convolution layers with untied biases, a kernel size of 3, no normalization, and LeakyReLU with a slope of 0.2 as the non-linear activation, except for the last layer which uses TanH. The SDF decoder is implemented as an MLP, which takes as input 64-dim features from the UNet, positional encoded d_j and c_j up to 4-th order Fourier features. The number of intermediate neurons in the first part of the MLP is $(128, 128, 129)$, where the output is split into a 128-dim feature vector and a 1-dim scalar, which is converted into non-negative weights by softmax across the k-NN samples. After weighted average pooling, the aggregated feature is fed into another MLP with a neuron size of $(128, 128, 1)$ to predict the SDF values. The MLPs use Softplus with $\beta = 100$ and a threshold of 20 as non-linear activation except for the last layer which does not apply any activation.

B Analysis on Input Pose Accuracy

We investigate how the accuracy of the input SMPL fitting influences the results on subject 50002 of DFaust [1]. As discussed in Sec. 4.1, the fitted SMPL parameters in

*Work done while Ziqian Bai was an intern at Reality Labs Research, Pittsburgh, PA, USA.

Table A: **Quantitative Evaluation on Input Pose Accuracy on Subject 50002.** We show the results of our approach and SNARF [2] using the poses provided by the AMASS [6] dataset and the ones after refinement using all vertices in the registered meshes. While SNARF is greatly influenced by the accuracy of pose parameters, the slight improvement in our method illustrates its robustness to SMPL fitting errors. In addition, our approach significantly outperforms SNARF even after pose refinement in most settings except for the 16-30 rollouts in the interpolation set.

(a) Mean Scan-to-Prediction Distance (mm) ↓

		Rollout (# of frames)					
		1	2	4	8	16	30
Interpolation Set							
AMASS [6]	SNARF [2]	7.898	7.715	7.588	7.840	7.898	8.238
	Ours	1.731	2.127	2.953	4.325	5.606	6.455
Refined Poses	SNARF [2]	3.982	4.001	3.964	4.068	4.029	4.158
	Ours	1.417	1.703	2.259	3.241	4.044	4.601
Extrapolation Set							
AMASS [6]	SNARF [2]	8.083	8.126	8.160	8.246	8.050	8.025
	Ours	1.259	1.479	1.984	2.883	4.023	4.867
Refined Poses	SNARF [2]	4.624	4.632	4.672	4.749	4.548	4.447
	Ours	1.149	1.329	1.745	2.486	3.313	3.855

(b) Mean Squared Error of Volume Change ↓

		Rollout (# of frames)				
		2	4	8	16	30
Interpolation Set						
AMASS [6]	SNARF [2]	0.01623	0.01590	0.01688	0.01703	0.01829
	Ours	0.00990	0.01135	0.01417	0.01597	0.01815
Refined Poses	SNARF [2]	0.01401	0.01349	0.01430	0.01426	0.01524
	Ours	0.00849	0.01002	0.01248	0.01389	0.01558
Extrapolation Set						
AMASS [6]	SNARF [2]	0.01228	0.01244	0.01333	0.01292	0.01264
	Ours	0.00602	0.00756	0.00977	0.01082	0.01140
Refined Poses	SNARF [2]	0.01094	0.01092	0.01148	0.01099	0.01080
	Ours	0.00559	0.00691	0.00871	0.00953	0.01000

DFaust are provided by the AMASS [6] dataset that uses sparse points on the registered data as approximated motion capture marker locations and computes the parameters using MoSh [4]. We observe that the provided pose parameters sometimes exhibit small misalignment with respect to the input scans. While the fitting quality in the AMASS dataset is sufficient for our approach, we also evaluate the performance on more accurate pose parameters by using all the vertices on the registered meshes. More specifically, we first compute a better template by unposing the registered meshes in the first frame of each sequence using the LBS skinning weights of the SMPL template, and averaging over all the sequences. Using this new template, we optimize pose parameters for each frame with an L2-loss on all the registered vertices. Note that in this experiment, we use the original template with the refined pose parameters instead of the refined template in order not to unfairly favor our method over SNARF [2].

In Tab. A, we report the mean absolute error of scan-to-prediction distance (mm) and the mean squared error of volume change for our method and SNARF. Tab. A shows that SNARF has a large error reduction with refined poses, indicating that SNARF is highly sensitive to the accuracy of the SMPL fit. We also observe that after pose refinement, SNARF overfits more to training poses (e.g., interpolation) as SNARF cannot model history-dependent dynamic deformations. In contrast, our method is more robust to the fitting errors, and significantly outperforms SNARF in most settings except for 16-30 rollouts in the interpolation set. Note that the results with longer rollouts favor "mean" predictions over more dynamic predictions, and do not inform us of the plausibility of the synthesized dynamics (see the discussion in Sec. 4.2).

C k-NN vs. Closest Surface Projection

As discussed in Sec. 3.3, our SDF decoding approach uses k-nearest neighbors (k-NN) of the SMPL vertices instead of closest surface projection [3]. Fig. A illustrates the limitation of this alternative approach proposed in Neural Actor [3]. As shown in Fig. A, we observe that associating a query location with a single closest point on the surface leads to poor generalization to unseen poses around regions with multiple body parts in close proximity (e.g. around armpits). In contrast, our approach, which associates query points with multiple k-NN vertices, produces more plausible surface geometry even for unseen poses.

D Ablation on Number of Used History Frames

As discussed in Sec. 3, our method takes in the information of $T = 3$ history frames to infer the future body shape. We provide an ablation study on the number of history frames T used by our method. Tab. B shows that small T ($T = 1, 2$) lead to less accurate predictions. We chose $T = 3$ for a good trade off between accuracy and computational cost.

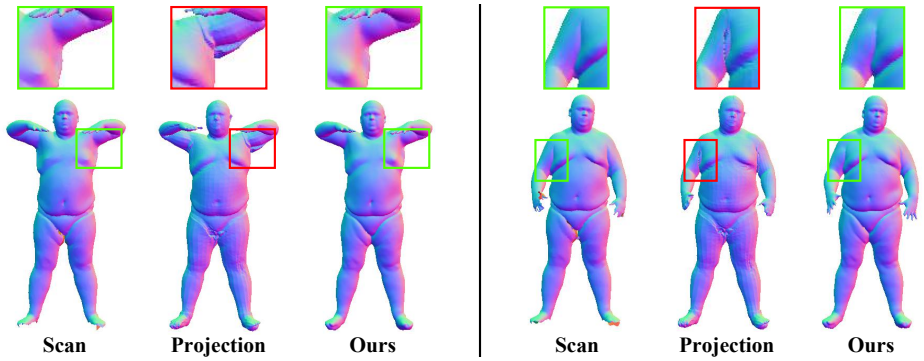

Fig. A: **k-NN vs. Closest Surface Projection.** While the closest surface projection suffers from artifacts around armpits, our SDF decoding based on k-NN produces more plausible surface geometry for unseen poses.

Table B: Mean Scan-to-Prediction Distance (SPD) and Mean Squared Error of Volume Change (VC) on DFaust 50002 Subject.

	T=1	T=2	T=3	T=4
SPD (Rollout=1)	1.671	1.437	**1.415**	1.427
VC (Rollout=2)	0.00808	0.00738	0.00732	**0.00710**

E Vertex Subsampling for k-NN Query

As discussed in Sec. 3.4, we subsample 3928 vertices by poisson-disk sampling on the SMPL mesh for k-nearest neighbor (k-NN) computation, to ensure uniformly distributed vertices and consistent area coverage by k-NN selection. In fact, when using all SMPL vertices, we sometimes observe artifacts around joints where vertex density tends to be non-uniform, such as the elbow in Fig. B. If we further reduce vertices to one half (from 3928 to 1649), the errors increase from 1.415 / 0.00732 (SPD / VC) to 1.517 / 0.00772.

F Time and Memory Cost

Our method takes $\sim$ 2.1s and $\sim$ 4.7GB memory on the GPU to infer one frame, with one RTX A5000 and one i9-10920X.

G Limitation: Clothing Deformations

We also apply our method on the CAPE [5] dataset that contains 4D scans of clothed humans. We select the subject `03375_longlong`, which exhibits the most visible dynamic deformations for clothing. We exclude 6 sequences (`athletics`, `frisbee`, `volleyball`, `box_trial1`, `swim_trial1`, `twist_tilt_trial1`) from training, and use them for testing. We employ as input the template and SMPL poses provided

Fig. B: Removing vertex subsampling results in artifacts around the elbow.

by the CAPE dataset for training our model. Note that we approximate raw scans by sampling point clouds with surface normals computed on the registered meshes as the CAPE dataset only provides registered meshes for `03375_longlong`.

Please refer to the supplementary video for qualitative results. While our approach produces plausible short-term clothing deformations, it remains challenging to model dynamically deforming clothing with longer rollouts. Compared to soft-tissue deformations, dynamics on clothed humans involve high-frequency deformations and topology change, making the learning of clothing dynamics more difficult. We leave this for future work.

H Limitation: Lack of hand details

The input raw scans provided by DFaust dataset are often incomplete and noisy around hand regions (see Fig. C and Fig. 1 in the main paper). While the proposed shape learning method allows us to fill missing regions across frames, highly corrupted or unobserved regions throughout the sequences remain challenging to recover. How to hallucinate missing information from a generic body prior could be an interesting direction for future works.

I Limitation: Artifacts around stomach

In the supplemental video, we can occasionally observe relatively large artifacts when the hand is close to the stomach (3:15, 3:56 and 6:27). The artifacts are caused by an infeasible body state of the arm penetrating inside the body. We attribute this to naively transferring pose parameters between source and target, whose body shapes are largely deviated (the source is skinny, whereas the target is plump). How to correct poses based on body shape could be an interesting problem for future works.

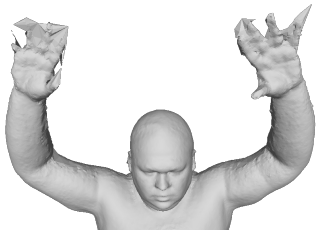

Fig. C: Noisy and incomplete hands in training data.

References

1. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic faust: Registering human bodies in motion. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 6233–6242 (2017) 1
2. Chen, X., Zheng, Y., Black, M.J., Hilliges, O., Geiger, A.: Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In: Proc. of International Conference on Computer Vision (ICCV) (2021) 2, 3
3. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: Neural free-view synthesis of human actors with pose control. ACM Trans. on Graphics (TOG) **40**(6), 1–16 (2021) 1, 3
4. Loper, M., Mahmood, N., Black, M.J.: Mosh: Motion and shape capture from sparse markers. ACM Trans. on Graphics (TOG) **33**(6), 1–13 (2014) 3
5. Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., Black, M.J.: Learning to dress 3d people in generative clothing. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 6469–6478 (2020) 4
6. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proc. of International Conference on Computer Vision (ICCV). pp. 5442–5451 (2019) 2, 3