Pose-NDF: Modeling Human Pose Manifolds with Neural Distance Fields Supplementary

Garvita Tiwari[1,2], Dimitrije Antic[1], Jan Eric Lenssen[2], Nikolaos Sarafianos[3], Tony Tung[3], and Gerard Pons-Moll[1,2]

[1] University of Tubingen, Germany
{garvita.tiwari, dimirije.antic, gerard.pons-moll}@uni-tuebingen.de
[2] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
jlenssen@mpi-inf.mpg.de
[3] Meta Reality Labs-Research, Sausalito, USA
{nsarafianos, tony.tung}@fb.com

1 Implementation Details

1.1 Data Preparation

Training data: We use the AMASS dataset [4] to train our model. We assume that AMASS represents a dataset of realisitc poses and all the poses, which are not part of AMASS are not plausible, with some confidence value. As mentioned in the experiment section, we prepare a dataset of points (poses) and corresponding distance values using kNN. We implement kNN using FAISS [3] and Pytorch3D [7]. We first approximate k'-NN of a query pose using FAISS and L2 distances, where $k' >> k$. Then we use the geodesic distance to find exact k neighbours from these k' neighbours. Specifically we use $k' = 500$ and $k = 5$ in our case. We use this multi-step approach, because the AMASS dataset is very large and our proposed data preparation steps are efficient. We generate the ground truth distance by taking the average of the k smallest distances. We use approximately 21M poses from the AMASS dataset for data preparation and training. Following VPoser [6], we also sample random frames from the motion sequence data, in order to avoid repetitive poses.

Evaluation and Validation: For validation, we use the validation split of the AMASS dataset and prepare distance values with respect to the train split. For testing the accuracy of the distance field prediction in our model, we use the test-split of AMASS dataset. For downstream tasks, we use existing real world mocap data like [2] and [4]. For comparison on the EHF dataset, we follow the evaluation provided in [6], i.e. we align the predicted SMPL mesh with the ground truth using Procrustes and then calculate the error on SMPL vertices (not on face and hands).

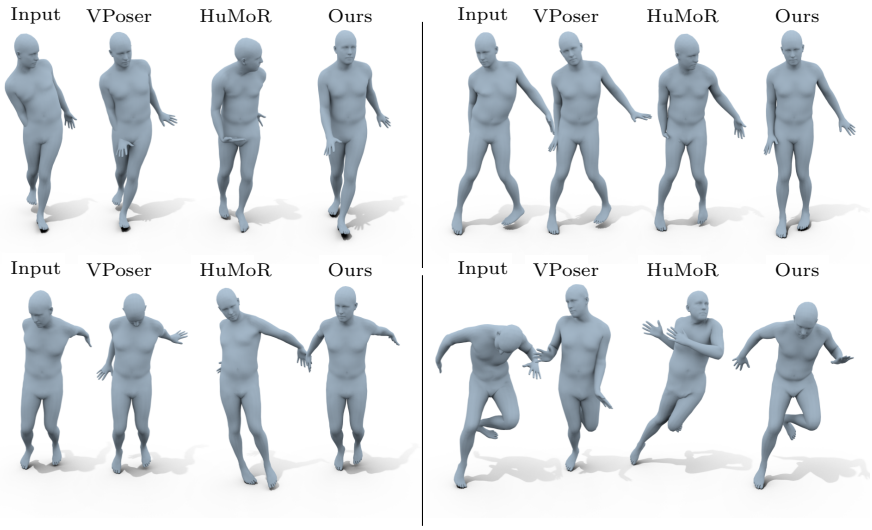

Fig. 1. Motion denoising for noisy mocap data ("Noissy AMASS"). We observe that Pose-NDF based motion denoising makes the pose realistic and also resembles the input observation, while VPoser and HuMoR still result in unrealistic poses and in some cases deviate too much from the input observation (bottom-left)

1.2 Network Architecture

We implement the hierarchical pose encoding network using structural MLPs [5,1], where each MLP consists of 2 layers, followed by a 5 layer MLP for distance field prediction. We use softplus as activation for the hidden layers, with $\beta = 100$.

2 More results

2.1 Motion Denoising

We show more results of motion denoising in Fig 1. We observe that the VPoser based motion denoising does not change the pose much if the noisy observation seems similar to common poses like standing with hands near the body (first row in Fig 1) and changes significantly in a random way when the poses look rare (bottom row in Fig 1). HuMoR changes the pose into m ore realistic poses at first (for initial frames), but results in unrealistic poses later because of accumulation of changes over time, e.g. (bottom-left in Fig 1). Pose-NDF seems to perform better in all such cases, resulting in realistic poses and also is not deviating much from noisy observations.

We evaluate the average % of self-intersecting mesh faces in the motion denoising task to evaluate which method produces more realistic poses. As seen from qualitative examples, HuMoR and VPoser generate results with self-intersecting poses. We evaluated this quantitatively, by counting the number of intersecting faces for all methods in Table 1. Pose-NDF clearly produces less intersections, which we credit to our detailed manifold.

Method \ Data	HPS	AMASS	Noisy AMASS	Partial Observation
Input	3.13	1.14	2.54	-
VPoser [6]	3.16	1.36	2.54	2.98
HuMoR [8]	2.81	**1.04**	1.43	2.13
Pose-NDF	**2.28**	1.05	**1.40**	**2.01**

Table 1. Average % of self-intersecting mesh faces in motion denoising. It can be seen that in most cases, Pose-NDF produces less intersections than VPoser and HuMoR.

2.2 Fitting to partial data

We show qualitative results of estimating 3D poses from partial observations. As discussed we perform this experiment on three different kinds of occlusions, namely 1) occluded left leg, 2) occluded left arm and 3) occluded right shoulder and upper arm. We observe that VPoser is biased towards mean poses, (e.g. as seen in Fig. 2), VPoser tends to produce nearly straight legs for occluded leg cases, which is more commonly seen in training data. On the other hand HuMoR and PoseNDF produce more variety, but in some cases of HuMoR, the resulting pose looks unrealistic due to accumulation in correction in input pose. For our method, it highly depends on initialization. Since we initialise randomly near the mean pose in our experiments, it looks similar to VPoser in most cases.

2.3 Pose Interpolation

The Pose-NDF manifold can be used to interpolate between two distinct poses by traversing the manifold. We provide more such examples of pose interpolation using Pose-NDF in Fig 3.

2.4 Runtime

We compare the runtime of each method used in optimization based pose recovery methods. For motion denoising and motion recovery from partial observations, we conduct our experiments on a V100 GPU whereas for image based reconstruction, we used an RTX 3080Ti. HuMoR for motion denoising and pose recovery from partial observation takes approximately 10.83 sec. and 10.23 sec., respectively, for one frame. On the other hand, VPoser based optimization is way faster and takes only 0.96 sec. per frame. Pose-NDF based optimization takes even less: only 0.56 sec for motion denoising and partial observation experiments. This is because Pose-NDF is fast per step and has less optimization steps, as compared to VPoser and HuMoR. For the image based pose and shape reconstruction task, VPoser and Pose-NDF, take 3.63 sec. and 4.59 sec. respectively.

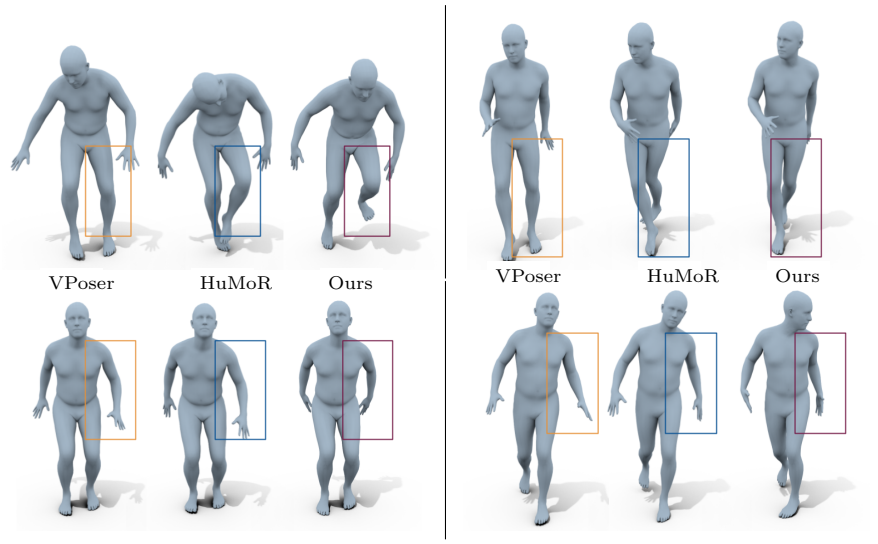

Fig. 2. Fitting to partial observation: We compare VPoser, HuMoR and Pose-NDF based prior on the task of recovering 3D pose from partial observation. (Top): In case of occluded left leg, we observe that VPoser produces nearly straight legs(close to mean position), while HuMoR and PoseNDF produce much more diverse poses. HuMoR results in extreme unrealistic poses in some cases. We observe similar behavior for occluded arm case (bottom).

Fig. 3. Pose interpolation using Pose-NDF.

References

1. Aksan, E., Kaufmann, M., Hilliges, O.: Structured prediction helps 3D human motion modelling. In: ICCV (2019) 2
2. Guzov, V., Mir, A., Sattler, T., Pons-Moll, G.: Human POSEitioning System (HPS): 3D human pose estimation and self-localization in large scenes from body-mounted sensors. In: CVPR (2021) 1
3. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019) 1
4. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: ICCV (2019) 1
5. Mihajlovic, M., Zhang, Y., Black, M.J., Tang, S.: LEAP: Learning articulated occupancy of people. In: CVPR (2021) 2
6. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR (2019) 1, 3
7. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3D deep learning with PyTorch3D. arXiv:2007.08501 (2020) 1
8. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: HuMoR: 3D human motion model for robust pose estimation. In: ICCV (2021) 3