– Supplemental Document –
SPAMs: Structured Implicit Parametric Models

Pablo Palafox[1*] Nikolaos Sarafianos[2] Tony Tung[2] Angela Dai[1]
[1]Technical University of Munich [2]Meta Reality Labs Research, Sausalito, USA

In this appendix, we provide additional details for our test-time optimization in Sec. 1, and then present an ablation study on the use of our pose encoder for pose code initialization in Sec. 1.1. In Sec. 2 we present a quantitative comparison with IP-Net [1]. Additional qualitative evaluations and results are shown in the supplemental video.

1. Test-time Optimization

In Eq. 6 in the main paper we present the energy term that is minimized at test-time when fitting our SPAMs to a depth sequence, which we rewrite here for completeness:

$$[\tilde{\boldsymbol{s}}_{\mathrm{q}}], \{[\tilde{\boldsymbol{p}}_{\mathrm{q}}^{j}]\}_{j=1}^{L} = \underset{[\boldsymbol{s}_{\mathrm{q}}], \{[\boldsymbol{p}_{\mathrm{q}}^{j}]\}_{j=1}^{L}}{\arg\min} \sum_{j=1}^{L} \sum_{\forall \boldsymbol{x}_k} \mathcal{L}_{\mathrm{r}} + \mathcal{L}_{\mathrm{c}} + \mathcal{L}_{\mathrm{t}} + \mathcal{L}_{\mathrm{icp}}. \quad (1)$$

As mentioned in the paper, $\mathcal{L}_{\mathrm{c}}$ enforces shape and pose code regularization through an ℓ_2 loss on the latent codes:

$$\mathcal{L}_{\mathrm{c}} = \sum_{q=1}^{Q} \frac{\|\boldsymbol{s}_{\mathrm{q}}\|_2^2}{\sigma_{\mathrm{s}}^2} + \frac{\left\|\boldsymbol{p}_{\mathrm{q}}^{j}\right\|_2^2}{\sigma_{\mathrm{p}}^2}, \quad (2)$$

with $\sigma_{\mathrm{s}}^2 = 0.01$, $\sigma_{\mathrm{p}}^2 = 0.001$.

$\mathcal{L}_{\mathrm{t}}$ enforces temporal regularization between the current frame j and its neighboring frames $H = \{j-1, j+1\}$. As in [3], this is enforced with an ℓ_2-loss on the pose MLP flow predictions for points $\boldsymbol{x}_k$, and controlled with a weight of $\lambda_t = 10$:

$$\mathcal{L}_t = \lambda_t \sum_{q=1}^{Q} \sum_{h \in H} \left\| f_{\theta_{\mathrm{p}}^{\mathrm{q}}}(\boldsymbol{s}_{\mathrm{q}}, \boldsymbol{p}_{\mathrm{q}}^{j}, \boldsymbol{x}_k) - f_{\theta_{\mathrm{p}}^{\mathrm{q}}}(\boldsymbol{s}_{\mathrm{q}}, \boldsymbol{p}_{\mathrm{q}}^{h}, \boldsymbol{x}_k) \right\|_2^2. \quad (3)$$

As presented in the Sec. 3.5.1 in the main paper, we employ a part-guided ICP-inspired loss $\mathcal{L}_{\mathrm{icp}}$. We recall that $\mathcal{L}_{\mathrm{icp}}$ is computed by establishing part-driven correspondences between each estimated part q from the predicted input depth map part labels and the part decoder predictions. To this end, every I_{resample} iterations we consider the canonically-posed shape from the current state of the shape codes, re-sample a new set of $N_t = 500$k points around the mesh and keep those within a distance $\epsilon_{\mathrm{icp}} = 0.005$ (in normalized units), denoted by $\boldsymbol{x}_k^{\mathrm{ns}}$, from the implicitly represented surface. We use our part decoder (Sec. 3.5 in the main paper) to estimate part labels for these canonical points, and then warp them into a posed frame j using our pose decoders. Then for every point in the input depth map $\boldsymbol{x}' \in D_j$, and given its predicted part q (obtained by PointNet++), we find its nearest neighbor in the warped set of points belonging to q based on $f_{\theta_{\mathrm{q}}}$, denoted by

$$\mathcal{W}_{\mathrm{q}} = \{\boldsymbol{x}_k^{\mathrm{ns}} + f_{\theta_{\mathrm{p}}^{\mathrm{q}}}(\boldsymbol{s}_{\mathrm{q}}, \boldsymbol{p}_{\mathrm{q}}^{j}, \boldsymbol{x}_k^{\mathrm{ns}})\} \quad (4)$$

to establish correspondences, and minimize the distance between these points:

$$\mathcal{L}_{\mathrm{icp}} = \lambda_{\mathrm{icp}} \sum_{q=1}^{Q} \sum_{\boldsymbol{x}' \in D_j} \left\| \boldsymbol{x}' - \mathrm{NN}_{\mathcal{W}_{\mathrm{q}}}(\boldsymbol{x}') \right\|_2. \quad (5)$$

In the above equation, $\mathrm{NN}_{\mathcal{W}_{\mathrm{q}}}(\cdot)$ denotes a function that queries the nearest neighbor of a 3D point in a set of points $\mathcal{W}_{\mathrm{q}}$. We control the importance of this loss with $\lambda_{\mathrm{icp}} = 20$ in our experiments.

Finally, we control $\mathcal{L}_{\mathrm{r}}$ (Eq. 7 in the main paper) with $\lambda_{\mathrm{r}} = 1$.

Optimizing over an input sequence of 90 frames until convergence (for 200 optimization steps) takes approximately 1.5 hours on a GeForce RTX 3090 with our highly unoptimized implementation.

1.1. Effect of Pose Code Initialization

We study the effect of pose code initialization in Fig. 1. For a given frame, we study how optimization evolves across different optimization steps for NPMs* [3] (with and without pose encoder initialization) and our SPAMs (with and without pose encoder initialization). Our part basis helps to establish global correspondences that provide robustness against lack of good pose initialization.

2. Additional Comparisons to State of the Art

In Fig. 3 we show a qualitative comparison with IP-Net and NPMs* on one of our test sequences; we show superior

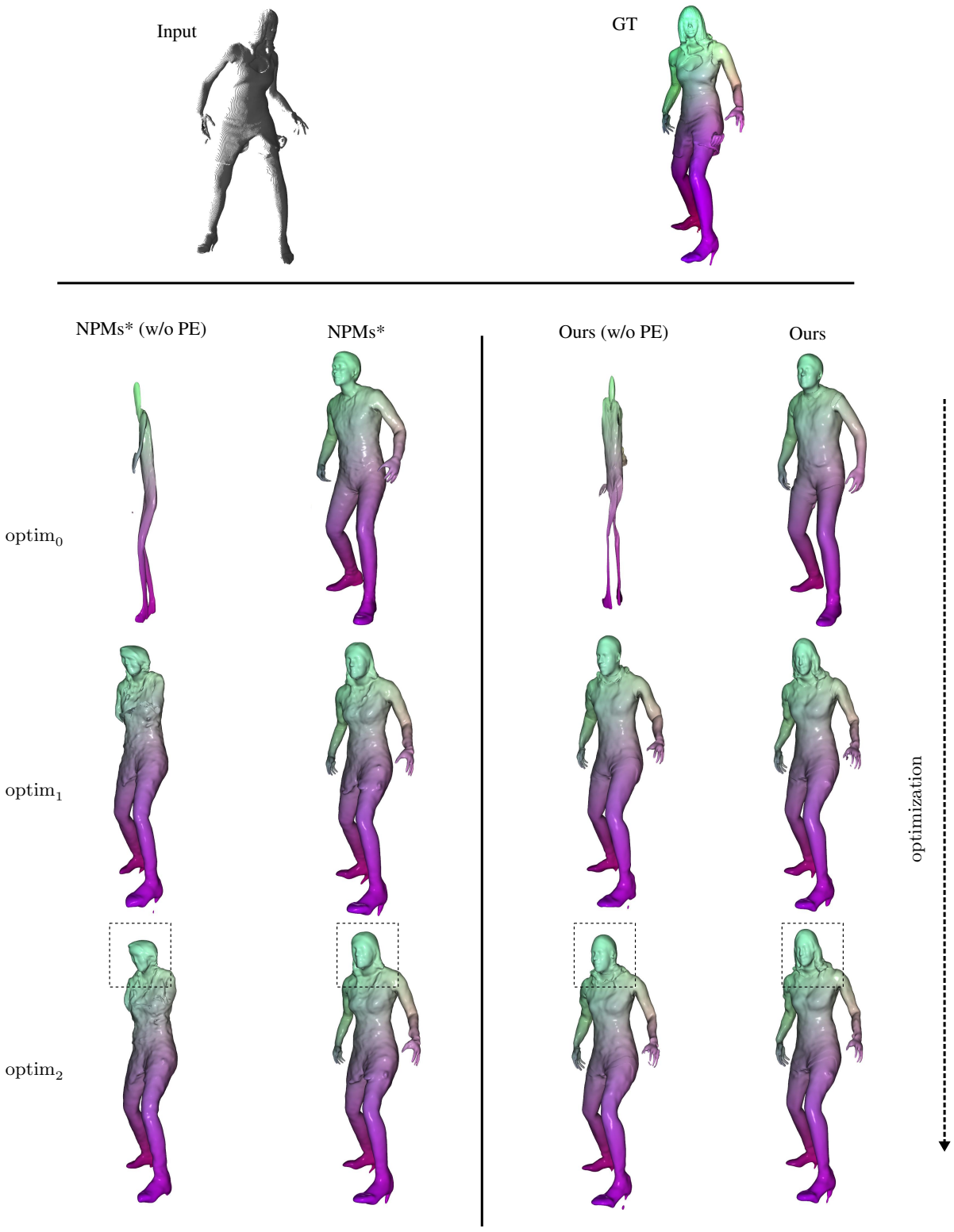

Figure 1. For a given frame, we study how optimization evolves across different optimization steps for NPMs* (with and without pose encoder initialization) and our SPAMs (with and without pose encoder initialization). Note that SPAMs are robust to pose code initialization, and can recover tracking even when starting from randomly initialized pose codes.

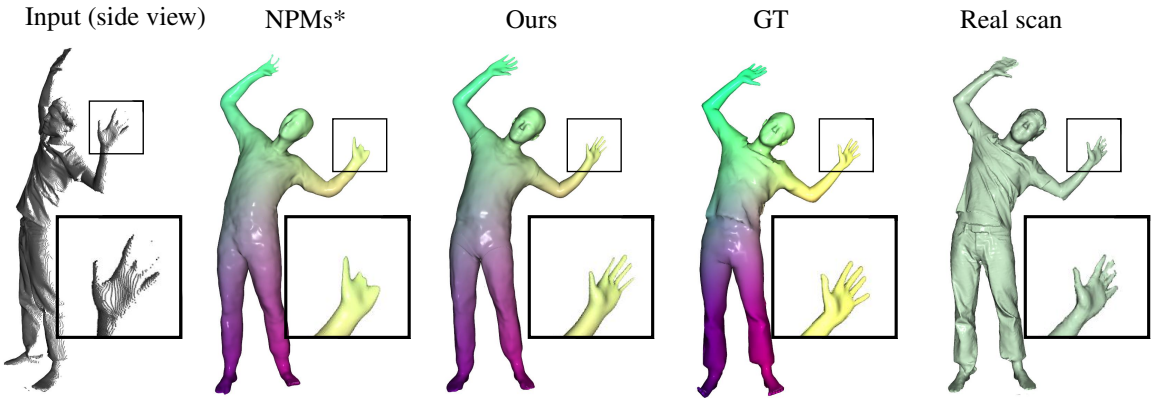

Figure 2. Comparison to the state-of-the-art NPMs* [Palafox et al. 21] on the task of model fitting to a monocular depth sequence from real CAPE scans [2], demonstrating the applicability of SPAMs to real data, capturing finer-scale details in the hands and face.

Method	**IoU** ↑	**C-ℓ_2** ↓	**NC** ↑	**EPE** ↓
NPMs*	0.808	0.0000347	0.895	0.0090
Ours (w/o ICP, w/o PE)	0.675	0.0010797	0.828	0.0387
Ours (w/o PE)	0.777	0.0000769	0.881	0.0138
Ours (w/o ICP)	**0.815**	0.0000371	0.895	0.0085
Ours	0.813	**0.0000268**	**0.900**	**0.0079**

Table 1. Quantitative evaluation on real human data from the CAPE [2] dataset.

performance in loop closing, demonstrating our tracking robustness while maintaining detailed geometry.

We also show a quantitative and qualitative comparison on CAPE [2]. In this experiment, we have taken the pre-trained models of our approach and NPMs* on RenderPeople and fine-tuned both methods on CAPE data (unlike the CAPE experiments from NPMs which were trained on a mixture of CAPE, AMASS, and Mixamo). As shown in Table 1 and Figure 2, our approach achieves superior performance both quantitatively and qualitatively in comparison with NPMs* (e.g., finer details of the fingers and faces).

References

[1] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision*, 2020. 1, 4

[2] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to dress 3d people in generative clothing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[3] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. *IEEE/CVF International Conference on Computer Vision*, 2021. 1, 4

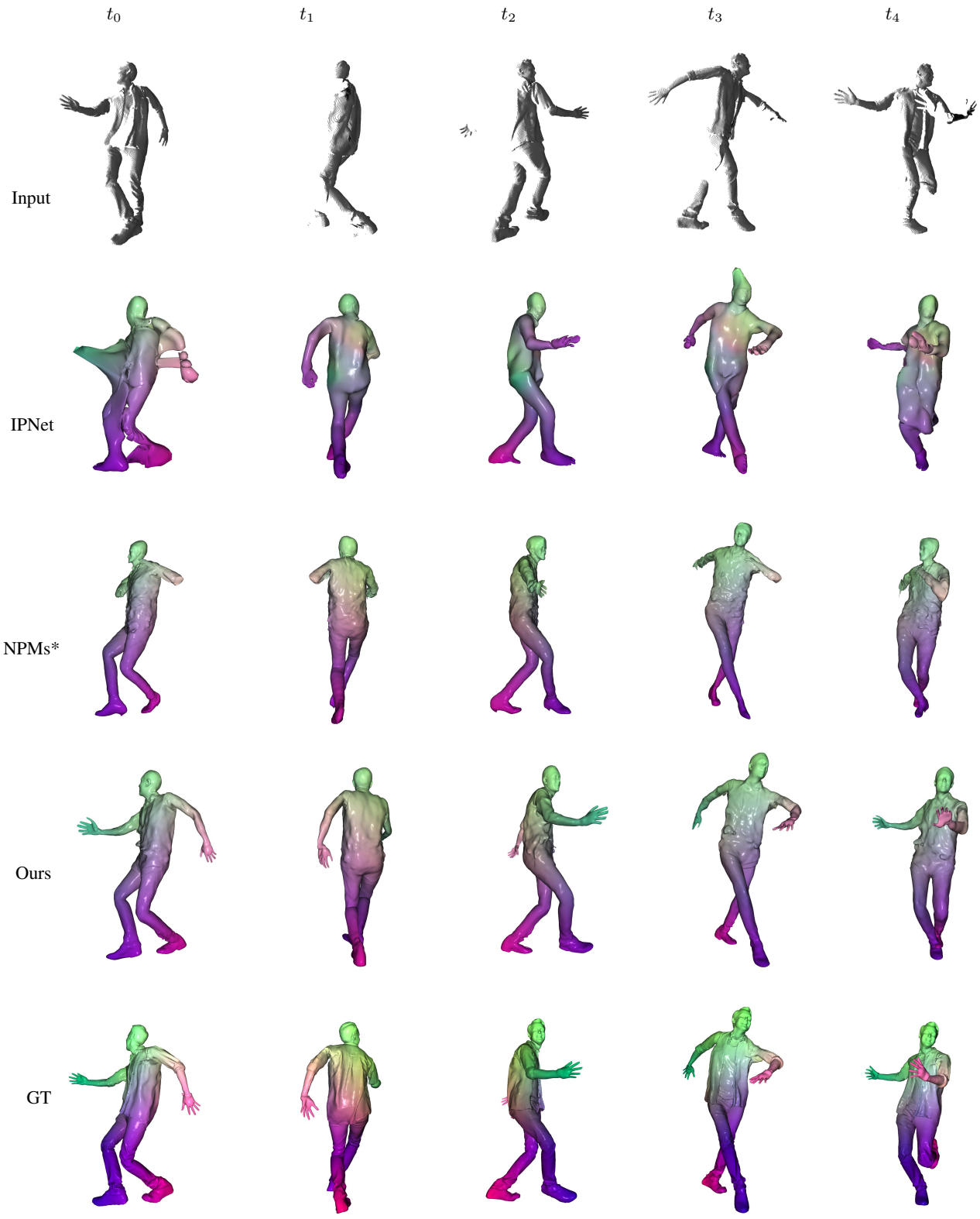

Figure 3. Qualitative comparison to NPMs* [3] and IP-Net [1] on the task of model fitting to a monocular depth sequence. In complex motion scenarios such as loop closures, NPMs* and IP-Net struggle to track the motion, whereas our SPAMs robustly maintains tracking.