Supplementary Material: PocoLoco

Siddharth Seth[1] Rishabh Dabral[2] Diogo Luvizon[2] Marc Habermann[2]
Ming-Hsuan Yang[1] Christian Theobalt[2] Adam Kortylewski[2,3]
[1]UC Merced [2]Max Planck Institute for Informatics [3]University of Freiburg

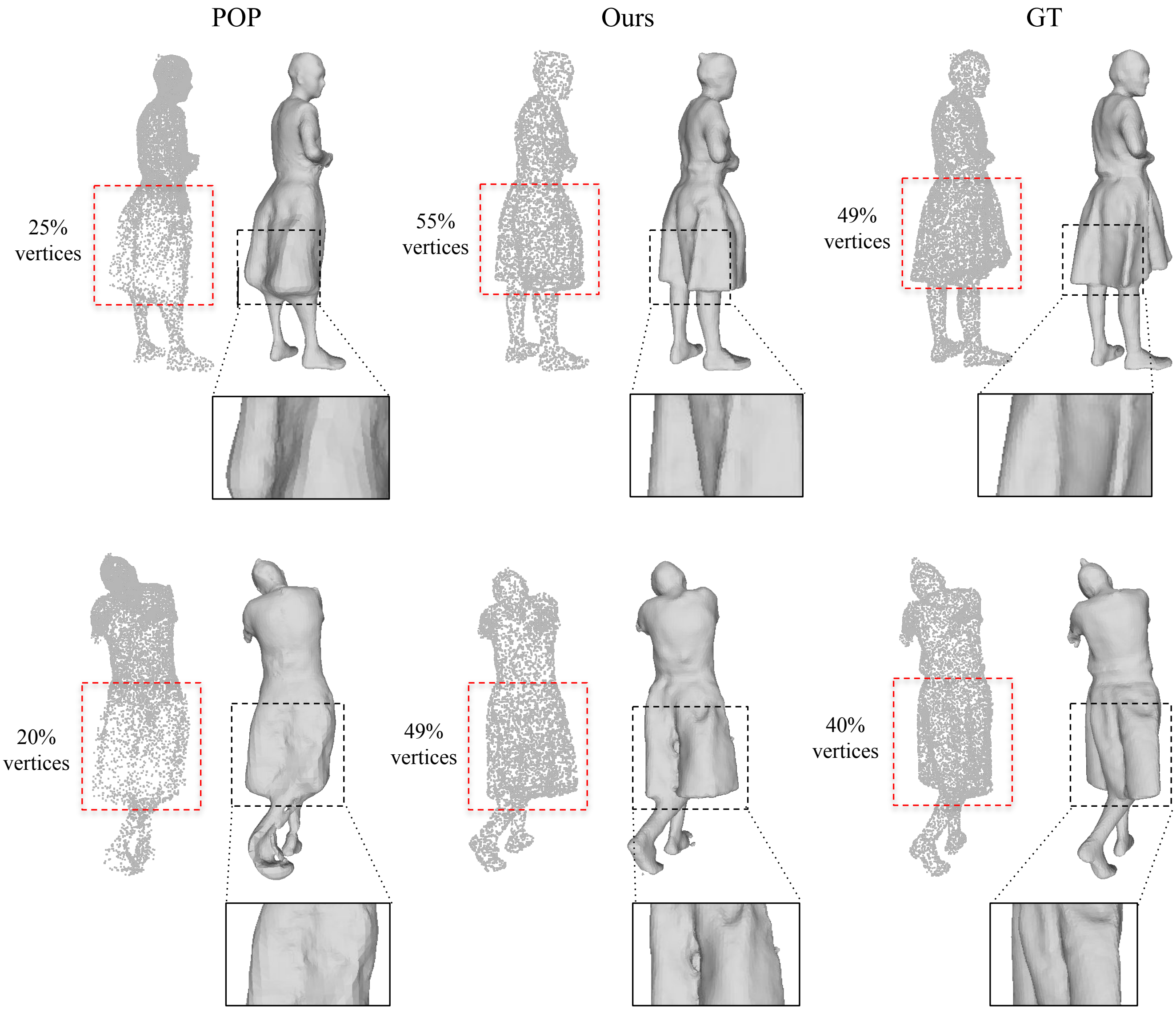

Figure 1. Qualitative comparison to POP showing results for loose clothing on unseen poses. We show both point clouds and their meshified versions for depicting point density and clothing deformation respectively. We additionally show the percentage of vertices occupying the loose clothing region (skirt). Due to modeling the clothing on top of a template model such as SMPL, the points from POP in the skirt region are too sparse to model any significant deformations. This is due to the points having a hard association with the nearest body part. Our method produces points much more consistently distributed across the body and clothing, thereby exhibiting realistic pose-dependent clothing deformations. Zoomed-in regions emphasize the most significant clothing deformations.

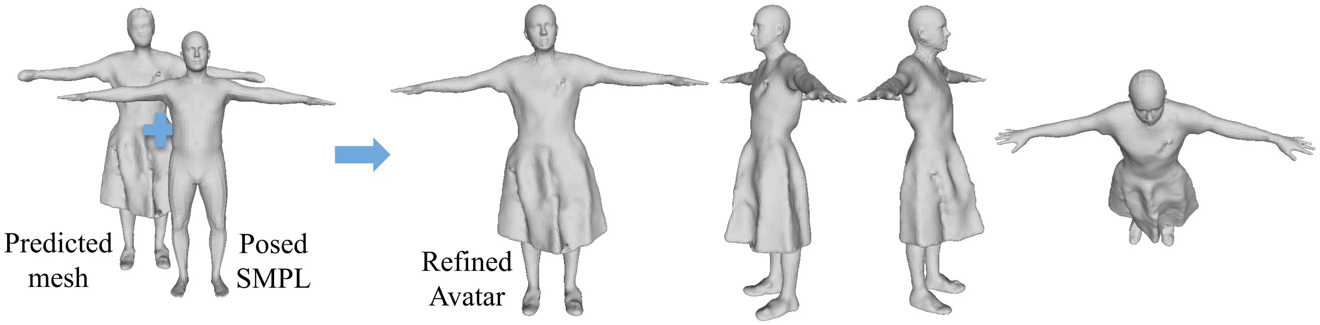

Figure 2. Using SMPL as a post-processing step helps in recovering face and the more difficult to obtain hand details.

1. Extended Results and Discussion

Here, we conduct further analysis of our results in the main paper. We compare with SoTA point-based method POP and also show how a posed SMPL mesh can be used to refine the obtained avatar from PocoLoco. Finally, we also provide more ablation studies to give an insight into the selection of our architecture.

1.1. Comparison with POP

POP shows quantitative results for multi-subject training. For a fair comparison, we train and evaluate POP in a subject-specific manner, similar to how we report results for PocoLoco.

Points sparsity. We show an in-depth comparison with POP here. The illustration in Fig. 1 reveals that points from POP exhibit sparsity in the skirt region, detecting fewer clothing deformations. In contrast, PocoLoco generates points evenly distributed across the body and clothing, enabling the recognition of clothing deformations. We quantify this by counting the number of vertices in the skirt region. While POP allocates approximately 25% of its points to the skirt region, PocoLoco assigns approximately 50% of its points to this area, contributing to the detection of significant deformations in the skirt region.

Performance on most difficult poses. Fig. 3 shows a performance comparison on the top 10% most difficult poses in the test set of the loose clothing subject in DynaCap. For each scan in the test set, we find the closest appearing sample in the train set and measure the CD. The samples in the test set with the highest CD are considered the most difficult poses as they do not appear in the train set. We pick the top 10% of such samples and show a quantitative comparison in Fig. 3.

Performance on LOOSE dataset. Lastly, we evaluate POP on our LOOSE dataset. POP achieves a CD of 1.95 cm (vs Ours 2.87 cm) on Subject 1 and 2.86 cm (vs Ours 3.63 cm) on Subject 2.

Reproducibility. Prior arts like POP [4] require the registration of an SMPL [2] body model to the 3D reconstructions before the training process, but unfortunately do not provide the code to obtain these in their public repositories. As our model is purely learning-based and does not require any prior registration of a scanned template or human body model, it will be completely reproducible on any other data given our released codebase.

1.2. SMPL based refinement

While previous approaches [4] utilize a template such as SMPL to constrain the space of deformations using Linear Blend Skinning, the resulting clothed meshes suffer from artifacts such as a tear in the skirt region due to points sparsity in modeling loose clothes. Our diffusion-based architecture attends to the uniform distribution of points in the loose clothing region thereby modeling clothing deformations. However, it may lose out on prior information such as facial and hand geometry available to template-based methods. We propose to mitigate this problem via a post-refinement step. Once we obtain the pose-conditioned point clouds from the inference pipeline, we fit a SMPL template to this test-time predicted unseen point cloud. Following this, we extract the head and hand regions from the SMPL template and replace them with our predictions to obtain a higher-quality posed avatar. Fig. 2 illustrates the results obtained using this approach. This is similar to how ECON [5] proposes an optional stage to obtain the final mesh.

1.3. Ablation studies

Importance of the number of points. We perform an ablation study to measure the performance as a function of the number of points used for training our method. Fig. 4 shows that more points yield better performance. In the top row, the model starts to lose out on details in the shirt's sleeve region as the number of points reduces. We see a similar effect in the skirt region as well. In the bottom row, points on the left hand become sparse with respect to the overall points. As the hand is the thinnest part of the body, we see a part of it does not have enough points to get the faces.

Scheduling policy. We propose the Quartic scheduling policy for our diffusion-based architecture which helps to re-

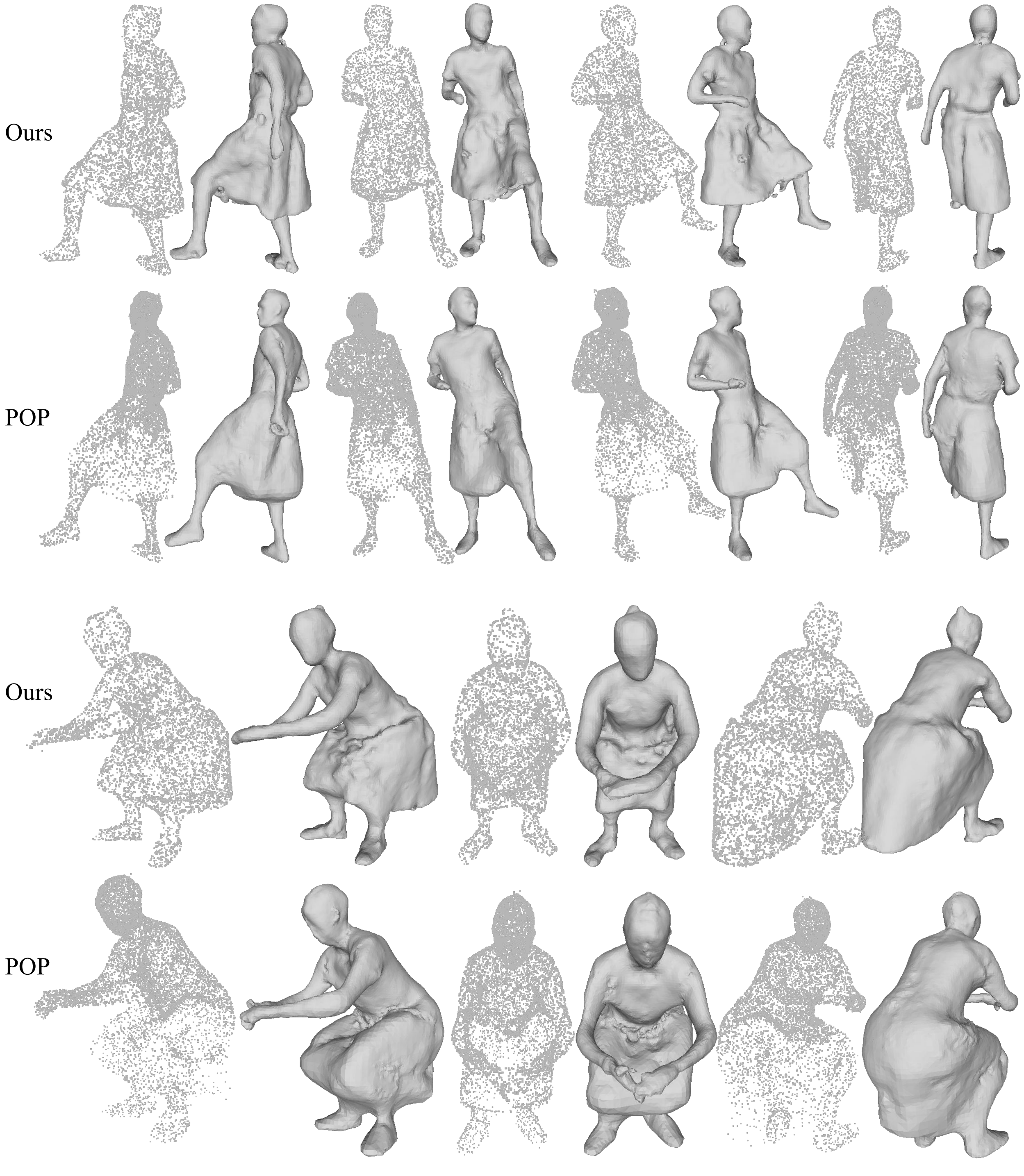

Figure 3. Qualitative comparison on top 10% most difficult poses in the loose subject of DynaCap dataset. POP obtains 7.2 cm CD while PocoLoco obtains 5.5 cm CD.

cover more details compared to the Linear scheduling algorithm. As Fig. 5 illustrates, the Quartic policy uses smaller beta values at the beginning and gradually uses higher values to convert the point clouds to a Gaussian distribution noise. This implies details such as clothing deformations are retained for more time steps during the forward diffusion step, and the coarse body shape is converted to noise in subsequent steps. During the reverse diffusion process,

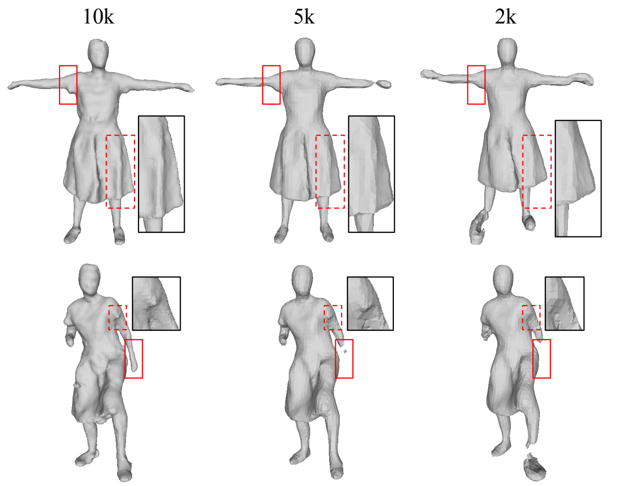

Figure 4. Visualization of modeling clothing deformation as a function of the number of points. More points yield better representation capability. We obtain a CD of 5.44 cm for 2k, 4.62 cm for 5k, and 4.22 cm for 10k points.

we benefit from recovering the coarse body shape early on so that the model can utilize more time steps towards recovering clothing deformations. We also experiment with the Cubic scheduling policy. However, beta values obtained using the quartic schedule work best for us.

Training time. The training time varies for PocoLoco from 4 (19500 frames) to 6 days (33500 frames) with 8 A100 GPUs. However, as we use a Transformer architecture we note that by using FlashAttention, the training time effectively reduces by 4x. Though we do not conduct a quantitative evaluation, we see no visible artifacts in the generated predictions using FlashAttention. Inference time is 80s per sample which can again be reduced using FlashAttention.

Cross, Self, Cross+Self Attention. We show in Fig. 6 the efficacy of using only self, only cross, and a combination of self and cross attention in the proposed Transformer architecture. Though self-attention works reasonably well, it takes a longer time to converge. Cross attention on the other hand does not converge after a long training time but can reasonably model the pose. A combination of self and cross-attention brings the best of both worlds by attending to the conditioned pose and converging to the target point cloud faster.

1.4. Comparison with SkiRT

[3] extend POP to predict blended skinning weights for each point. Besides the data used in training POP, one needs to have some *extra parameters* (as mentioned on the project page) to train SkiRT on a custom dataset that the authors have not yet discussed details about. It is thus not possible to reproduce the results from SkiRT on our dataset. We do not train our model on the ReSynth dataset as it only has about 1000 frames per subject which is insufficient to train a diffusion model. We refer the reader to the supplementary section S3.1 in [3] where the authors note that the points produced by SkiRT can still be visibly sparser than on other body parts. PocoLoco on the other hand does not suffer from such a problem.

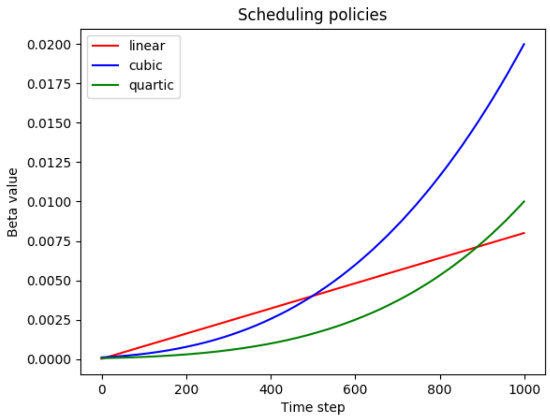

Figure 5. We depict the effect of sampling betas based on different noise scheduling policies used in our diffusion model. We choose quartic as it adds noise slowly at the beginning thereby retaining more details for a longer duration than the linear schedule. Best viewed in color.

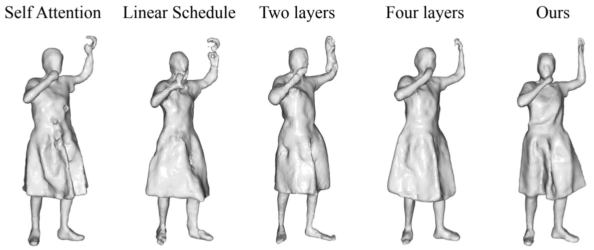

Figure 6. Ablation study to show the effectiveness of proposed components (using only self-attention, a linear noise schedule, or self and cross-attention with two and four layers) against the full model (rightmost). We note the importance of each of our design choices for the quality of the final generation results.

1.5. Application: Pose Editing

We can achieve pose editing in two ways. First, consider the problem of point cloud (PC) completion. For a partial PC $\in R^{(N-K)\times 3}$ with K missing points, we train our model to reconstruct the target PC $\in R^{N\times 3}$ by denoising K points sampled from Gaussian distribution. Similarly, we remove the points in the area of pose difference and proceed to reconstruct w.r.t. the conditioned target pose. The second approach is to add $t = 100$ steps of noise to the source pose PC and condition this with the target pose to get the pose edited result. We note that this works well for minor pose

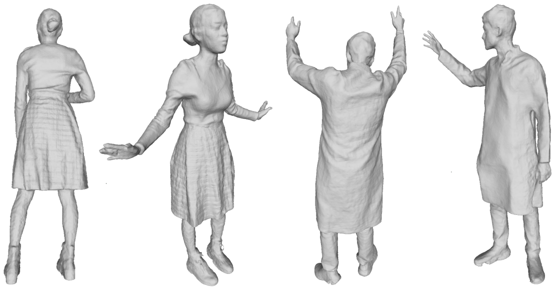

Figure 7. LOOSE dataset with 49K points, showing folds and wrinkles in striped skirts and long shirts. Best viewed zoomed in.

changes and may result in slight shape change for a major change in pose.

1.6. LOOSE Dataset details

We follow a process similar to DynaCap [1] for a convenient benchmarking method. To create the LOOSE dataset, we begin by scanning the actor in a T-pose with a 3D scanner. We then use commercial multi-view stereo reconstruction software, PhotoScan (http://www.agisoft.com) to generate the 3D mesh. This mesh is manually rigged to a skeleton. Additionally, we track human motions using a multi-view markerless motion capture system, TheCaptury (http://www.thecaptury.com/). Fig. 7 shows our dataset with high-quality geometric details using 49K points.

2. Limitations and future work

While our method is adept at recovering loose clothing deformations, modeling fine details such as facial and hand geometry is difficult. Prior arts benefit from this by using a SMPL template as a prior which helps them retain these details. We see this in Fig. 1 where POP models face geometry better than PocoLoco, albeit different from the GT. We propose a way to mitigate this in the post-processing step of our scans where we fit the posed SMPL meshes. On another note, we are limited to using 10k points due to our computational heavy transformer architecture. As we show in Fig. 4, using more helps recover more details. We leave the design of a more efficient architecture as a future work. Furthermore, the method may fail for extreme unseen poses.

Finally, since our method does not consider temporal consistency, a motion sequence extracted as a sequence of poses may exhibit noticeable changes in deformations for similar poses, leading to animation that lacks smoothness. We regard addressing this as a potential avenue for future research.

References

[1] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (ToG)*, 2021. 5

[2] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015. 2

[3] Qianli Ma, Jinlong Yang, Michael J. Black, and Siyu Tang. Neural point-based shape modeling of humans in challenging clothing. In *3DV*, 2022. 4

[4] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *ICCV*, 2021. 2

[5] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *CVPR*, 2023. 2