DuoMo: Dual Motion Diffusion for World-Space Human Reconstruction

Supplementary Material

A. Sparse mesh representation

A.1. Topology

We use a sparse mesh $\mathbf{X} \in \mathbb{R}^{V\times 3}$ to represent the human body and its motion. The sparse mesh has $V = 595$ vertices [3], and is designed to balance between representation quality and efficiency. Figure 1a shows the sparse mesh topology. Specifically, the vertices are positioned to ensure that body pose, shape, and hand gestures are accurately represented despite the sparsity. Consistent with prior methods [9, 12], skeletal joints are regressed from the posed mesh using linear combinations of local vertices.

A.2. SMPLX to sparse mesh

To leverage existing datasets for training our motion models, we create regression matrices to convert SMPL or SMPLX meshes to the sparse mesh, using a strategy similar to learning the joint regressor for SMPL [9]. Each vertex on the sparse mesh can be computed as a weighted average of a sparse set of vertices from the SMPL/SMPLX models.

With the regressor matrices, we convert training datasets [1, 10, 18] to the sparse mesh format. Figure 1b visualizes an example from the BEDLAM dataset [1] with its annotation converted to our sparse mesh format.

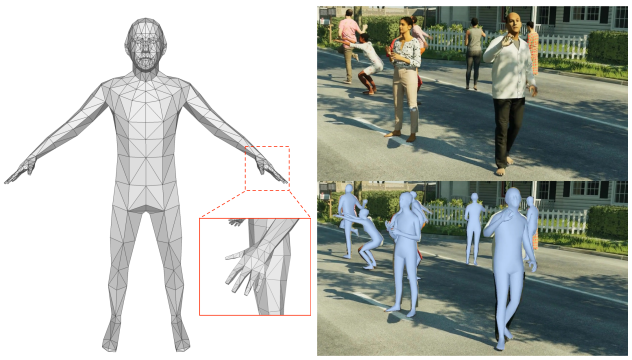
A.3. Sparse mesh to SMPLX

Our sparse mesh can represent detailed motion, but converting it to SMPLX provides a more compact representation and is necessary to compare reconstruction accuracy with prior works.

The conversion to SMPLX parameters can be achieved through optimization [12, 15], but it is slow and prone to local minima. In contrast, we train an optimization-inspired iterative network [2, 16, 23] for this task.

Our network performs iterative refinement using a cascade of three MLPs, as shown in Figure 2. The process begins with zero-initialized SMPLX parameters. At each stage, we generate a predicted sparse mesh from the current SMPLX parameters, using the SMPLX layer followed by our sparse regressor A.2. We then rigidly align the prediction to the target sparse mesh using the head vertices, and compute the per-vertex errors. The MLP takes the errors as input and predicts an update to the parameters. We repeat this process for two more stages. This cascaded design allows each MLP to specialize its refinement.

We tested the network on 3DPW [18] and find the MPJPE error to be under 5mm. We visualize some qualitative results in Figure 3.



(a) Topology (b) BEDLAM with sparse mesh

Figure 1. **Sparse mesh**. (a) Topology of our sparse mesh, with the red box showing the details of the hands. (b) An example from the BEDLAM [1] dataset with its annotation converted to the sparse mesh format.

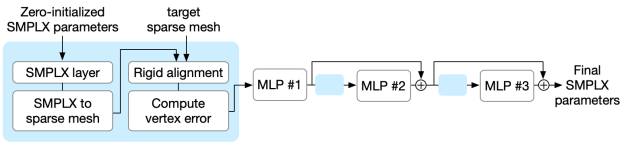


Figure 2. **Architecture (sparse mesh to SMPLX)**. This network performs iterative refinement to predict SMPLX parameters from a target sparse mesh.

B. Dense keypoint detection

The dense 2D keypoints are defined to semantically correspond to the 3D vertices of our sparse mesh. We used the conditional dense body keypoint detection model from SAM 3D Body [20] to infer these 595 body surface keypoints. This model takes a human–centered image and a set of 23 sparse human body keypoints as condition to estimate the dense surface points. We leverage the synthetic dataset BEDLAM [1] to train the network, where the backbone weights were initialized with ViTPose [19] to maintain the generalizability to the real images. During inference, we first run ViTPose to extract the sparse 23 keypoints from the input images, which are then fed into the network as the conditioning signal to predict the final 595 dense body surface keypoints. When training the diffusion models, the dense keypoint detection model is frozon.

Figure 4 visualizes keypoint detection results on the EMDB dataset [6]. Edges from the sparse mesh are added to improve visualization clarity.

C. Experiment details

C.1. Training

We train our camera-space model and world-space model with similar setups. We train each model with 8 H200 GPUs, with a batch size of 32 per GPU leading to an effective batch size of 256. The sequence length T is 120 frames.

Architecture. We use the same diffusion transformer (DiT) [13] architecture for the two motion models. It has 8 self-attention layers with $\mathrm{d\_model} = 512$, each with 8 heads for multi-head attention, and a hidden dimension of 2048 for the feedforward layer.

Data. To train our camerea-space motion model, we use AMASS [10], Goliath [11], BEDLAM [1] and 3DPW [18]. For AMASS and Goliath, we generate the 2D dense keypoints from the ground truth annotation and do not use images. For BEDLAM and 3DPW, we generate the 2D dense keypoints and use PromptHMR to extract image features.

To train our world-space motion model, we use AMASS [10], BEDLAM [1] and 3DPW [18]. We use the same procedure as training the camera-space model to generate the inputs. We then use the frozen camera-space model and lifting to generate the inputs for the world-space model training.

We train the dense keypoint detection model with BEDLAM by generating 2D dense keypoints from the ground truth annotation. We train the Sparse-mesh-to-SMPLX iterative network with BEDLAM and 3DPW.

Augmentation. We employ data augmentation and an augmentation in diffusion noise sampling during training.

We apply augmentation to the generated 2D dense keypoints for training the motion models. We apply noise, perturbation, and masking to the keypoints to simulate detection errors. We apply these augmentation at the point level (e.g. by sampling a set of keypoints) and at the part level (e.g. by applying the same perturbation or masking to all keypoints in a body part).

For training the diffusion models, we apply an augmentation to the sampling of diffusion time step k. For 50% of the samples, we uniformly sample time steps from $\{1, ..., 1000\}$ similar to the standard diffusion training [5, 17]. For 50% of the samples, we set $k = 1000$, corresponding to the highest level of corruption. This is equivalent to the "estimation mode" in GENMO [7], which finds this strategy encourages the diffusion model to produce good estimation during early diffusion steps.

C.2. Evaluation metrics

Camera-space reconstruction. We evaluate our model's camera-space reconstruction accuracy using three metrics: MPJPE, PA-MPJPE, and PVE. The MPJPE (mean per-joint position error) aligns the prediction with the ground truth at the pelvis location (removing translation) and measures the mean squared error (MSE) on the 3D joints. The PA-MPJPE (Procrustes-aligned MPJPE) rigidly aligns the prediction and ground truth 3D joints (removing rotation, translation and scale) before calculating the MSE on 3D joints. Finally, the PVE (per-vertex error) aligns the predicted and ground truth meshes at the pelvis (removing translation) and computes MSE on the vertices.

Figure 3. **Sparse mesh to SMPLX estimation**. Examples of the predicted SMPLX mesh (black) overlaid on the target sparse mesh (white).

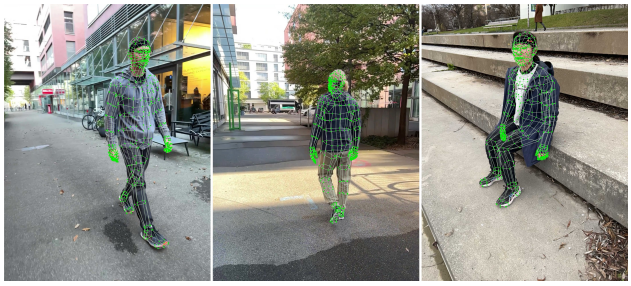


Figure 4. **Dense keypoint detection** on the EMDB dataset [6]. Edges added for visualization only.

World-space reconstruction. We evaluate world-space reconstruction accuracy with WA-MPJPE, W-MPJPE, RTE, Jitter and Foot sliding. WA-MPJPE and W-MPJPE both measure 3D joints MSE on 100-frame segments of prediction with the ground truth, but differ in how they align the prediction with the ground truth. WA-MPJPE aligns the whole segment (e.g. 3D joints across 100 frames) while W-MPJPE aligns the first two frame (e.g. 3D joints in the first two frames) [21]. Intuitively, WA-MPJPE measures the accuracy and coherence of the 100-frame motion snippet, while W-MPJPE additionally measures drift.

RTE (root trajectory error) measures the accuracy of the whole trajectory. It rigidly aligns the trajectories of the root and computes the mean square error in the unit of %. Jitter uses finite difference to compute the jerk on the 3D joints to access motion smoothness. Finally, foot sliding calculates the displacement on the predicted foot vertices on contact frames to measure erroneous sliding.

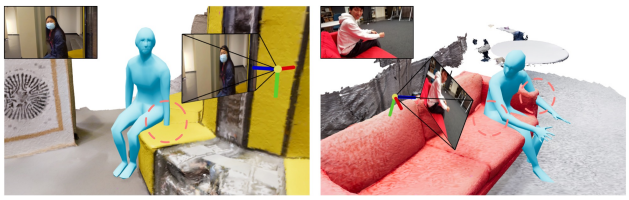


Figure 5. **Limitation 1**. Because our models do not incorporate 3D scene information, the results exhibit inconsistencies in fine-grained details.

Figure 6. **Limitation 2**. In challenging poses, the generated sparse meshes sometime exhibit unrealistic deformation. Converting them to SMPL meshes (sec A.3) partially improves the results.

C.3. Limitations

While DuoMo achieves a new state-of-the-art in world-space human motion reconstruction, it has several limitations that we want to address in future works.

Scene awareness. While we demonstrated that DuoMo can generate scene-consistent motion with guided sampling in the Experiment section, the 3D scene information is not explicitly used. Figure 5 illustrates some failures in terms of human-scene inconsistency. Future works could incorporate scene information or other physics-based objectives in the guided sampling process [22] to improve such scenarios. Another direction is to train a world-space motion model that can take 3D scene information as auxiliary conditioning.

Uncertainty awareness. We have modeled visibility at the keypoint level and the frame level, by replacing the embeddings of occluded keypoints or frames with null tokens (Method section). However, the keypoint visibility is determined by thresholding detection confidence, a procedure that is not always accurate. Turning detection confidence into a binary visibility label also discards information. Future work could improve accuracy of the motion model by integrating uncertainty reasoning with detection confidence.

Mesh generation robustness. We demonstrate that our architecture can generate the motion of mesh vertices, but in difficult cases the generated meshes could have unrealistic deformation. Figure 6 shows examples of deformed meshes. One interesting direction is to integrate our architecture with a tokenized or latent representation of the sparse mesh [4, 8] in the form of latent diffusion [14].

References

[1] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 1, 2

[2] Vasileios Choutas, Federica Bogo, Jingjing Shen, and Julien Valentin. Learning to fit morphable models. In *European Conference on Computer Vision*, pages 160–179. Springer, 2022. 1

[3] Aaron Ferguson, Ahmed AA Osman, Berta Bescos, Carsten Stoll, Chris Twigg, Christoph Lassner, David Otte, Eric Vignola, Fabian Prada, Federica Bogo, et al. Mhr: Momentum human rig. *arXiv preprint arXiv:2511.15586*, 2025. 1

[4] Guénolé Fiche, Simon Leglaive, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Mega: Masked generative autoencoder for human mesh recovery. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5366–5378, 2025. 3

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[6] Manuel Kaufmann, Jie Song, Chen Guo, Kaiyue Shen, Tianjian Jiang, Chengcheng Tang, Juan José Zárate, and Otmar Hilliges. EMDB: The electromagnetic database of global 3d human pose and shape in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14632–14643, 2023. 1, 2

[7] Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. *arXiv preprint arXiv:2505.01425*, 2025. 2

[8] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025. 3

[9] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 34(6):1–16, 2015. 1

[10] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019. 1, 2

[11] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad, Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu,

Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars. *NeurIPS Track on Datasets and Benchmarks*, 2024. 2

[12] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019. 1

[13] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[15] István Sárándi and Gerard Pons-Moll. Neural localizer fields for continuous 3d human pose and shape estimation. *Advances in Neural Information Processing Systems*, 37: 140032–140065, 2024. 1

[16] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *European Conference on Computer Vision*. Springer, 2020. 1

[17] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[18] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision*, pages 601–617, 2018. 1, 2

[19] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. VITPose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022. 1

[20] Xitong Yang, Devansh Kukreja, Don Pinkus, Anushka Sagar, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jiawei Liu, Nicolas Ugrinovic, et al. Sam 3d body: Robust full-body human mesh recovery. *arXiv preprint arXiv:2602.15989*, 2026. 1

[21] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21222–21232, 2023. 2

[22] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023. 3

[23] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3D human pose and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14484–14493, 2021. 1