RoHM: Robust Human Motion Reconstruction via Diffusion
Supplementary Material

A. Architecture Details

The detailed architecture of our model is illustrated in Fig. S1.

TrajNet adopts a U-Net structure built upon [4, 9], with a series of 1D temporal convolutional blocks ('ConvBlock') to downsample and upsample the input root trajectory $\boldsymbol{R}_t$ at diffusion denoising step t, and predict the clean trajectory $\hat{\boldsymbol{R}}_0$. The U-Net encoder and decoder are connected via skip connections. At each inference iteration i (Sec. 4.3 in the main paper), an extra conditioning encoder, structured similarly to the U-Net encoder, encodes the trajectory signal ($\hat{\boldsymbol{R}}_0^{i-1}$ for inference iteration $i > 1$, yellow arrow, and $\boldsymbol{M}_R \odot \tilde{\boldsymbol{R}}$ for $i = 1$, green arrow) into multi-layer features. These features are concatenated with the intermediate U-Net encoder features at each convolutional block. These two parts constitute the "vanilla" TrajNet.

TrajControl models pose-trajectory correlations and further refines root trajectory (Sec. 4.2 in the main paper), based on the denoised and infilled local body pose $\hat{\boldsymbol{P}}_0^{i-1}$ from the previous inference iteration $i-1$. Namely, upon completing the training of the vanilla TrajNet, the U-Net encoder, along with its weights, is duplicated to serve as the TrajControl encoder, to encode pose information. The intermediate pose features are added to the U-Net decoder via zero convolution layers (1x1 convolution with weights and bias initialized from zero). The TrajControl module is fine-tuned while keeping other TrajNet components frozen. This ensures that the vanilla TrajNet can process input even when only a corrupted trajectory is provided.

PoseNet builds on the transformer encoder architecture from [12]. At each diffusion denoising step t during inference iteration i, the input local pose $\boldsymbol{P}_t$ is concatenated with the estimated trajectory from TrajNet, $\hat{\boldsymbol{R}}_0^i$, and then fed into the transformer encoder. Regarding the conditioning signal, body pose (corresponding to the corrupted pose $\boldsymbol{M}_P \odot \tilde{\boldsymbol{P}}$ for iteration $i = 1$, green arrow, and the estimated body pose $\hat{\boldsymbol{P}}_0^{i-1}$ for iteration $i > 1$, yellow arrow) is combined with the estimated trajectory $\hat{\boldsymbol{R}}_0^i$ and processed through a linear embedding layer. This conditioning feature, along with the embedding of the diffusion step t, serves as the input to the transformer encoder.

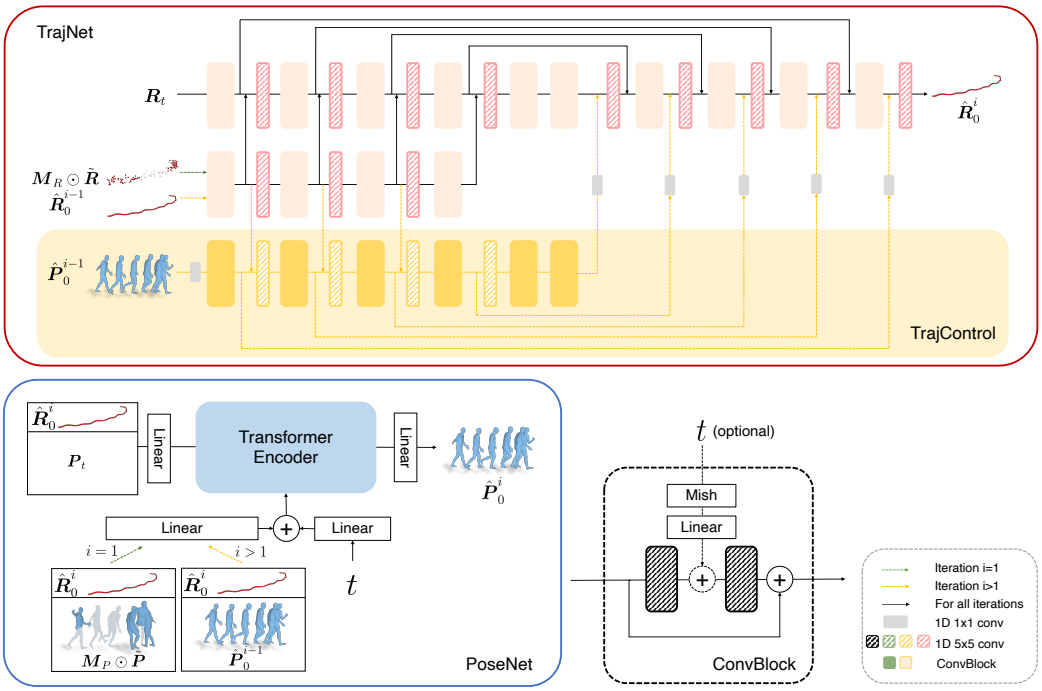

Figure S1. **Model architecture for TrajNet, TrajControl, and PoseNet.**

B. Implementation Details

B.1. Training Details

Data augmentation. During training, we apply Gaussian noise and synthetic occlusion masks to ground-truth motion sequences from AMASS [7], to simulate noisy and occluded input motion $\tilde{\boldsymbol{X}}$. We add Gaussian noise with a noise level k to the ground-truth SMPL-X parameters, with zero mean and standard deviation of $(k^\circ, k^\circ, k\ cm, 0.01k)$ for $(\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\beta})$; we then obtain noisy 3D joint positions via forward kinematics. During the initial training phases, the model is trained on easier tasks, with lower noise levels and smaller occlusion ratios for $\tilde{\boldsymbol{X}}$. As the training progresses, we gradually expose the model to harder cases, with higher noise levels and heavier occlusions, as detailed below.

Training schemes. TrajNet undergoes training in four stages. In the initial two stages, the model is trained with noise level $k = 1$ and $k = 2$, respectively, without occlusions. In the third stage, the noise level is raised to $k = 3$, with and 10% of the frames entirely masked out. Upon completion of these stages, the training for the vanilla TrajNet is concluded. In the last stage, TrajControl is fine-tuned to incorporate additional control from body pose, with a noise level of $k = 2$ and no occlusion masks. PoseNet follows a two-stage training process. In the first stage, the model is trained with a noise level of $k = 1$. To synthesize occlusion masks, we randomly mask out 1-6 joints in the initial 500 epochs. Afterwards, a mixed occlusion scheme is applied: with 0.5 probability, occlusion masks from PROX pseudo ground truth are used; with 0.3 probability, all lower body parts are masked out; with 0.2 probability, all upper body parts are masked out; with 0.1 probability, the full body is masked out in 30% of the frames. In the second stage, we continue the mixed occlusion scheme and increase the noise level to $k = 2$.

Noise assumptions. We experiment with various training noise scales, and Tab. S1 confirms the synthetic noise distribution on AMASS given by the selected scale aligns with real-world input noise (given by VPoser-t initialization on EgoBody). Synthetic noise applied to SMPL-X parameters accumulates along the kinematic tree, which mirrors real-world scenarios: running VPoser-t on EgoBody yields larger MPJPEs for wrists (0.23m) than shoulders (0.12m).

MPJPE	0-0.2m	0.2-0.4m	0.4-0.6m	>0.6m
AMASS (synthetic noise)	78.1%	18.9%	2.7%	0.3%
EgoBody (real-world noise)	82.0%	13.7%	3.5%	0.8%

Table S1. **The synthetic noise distribution aligns with real-world noise:** percentage of joints whose GMPJPE (between noisy / GT joints) falls into each range.

Training weights. For both PoseNet and TrajNet, weights λ_{3D} and λ_{vel} are set to 100 and 1000, respectively. λ_{skate} is set to 0 during the first training stage, and 0.1 during the second training stage in PoseNet.

PoseNet Training strategy. PoseNet is trained with the GT trajectory instead of TrajNet output, as the small error between them (pelvis GMPJPE 12.5 mm) ensures minimal impact. Training PoseNet with TrajNet output requires separate trainings with vanilla and fine-tuned TrajNet, thus not optimal for efficiency.

B.2. Motion Representation

In the proposed motion representation, the root linear position $\boldsymbol{r}^l$ and height r^z denote the pelvis xy coordinates projected on the ground, and pelvis z coordinate, respectively. There is a shape-dependent shift between $(\boldsymbol{r}^l, r^z)$ and the SMPL-X body translation $\boldsymbol{\gamma}$. The root angular rotation r^a refers to the body rotation around z-axis. This is similar to SMPL-X global orientation $\boldsymbol{\Phi}$, but not identical. Existing works [5, 12] typically adopt only the joint-based representation, and resort to time-consuming post-processing optimization to obtain the expressive SMPL body meshes. Our over-parameterized representation enables learning of motion dynamics in both the joint space and SMPL-X parameter space, such that the model can directly predict the SMPL-X bodies without any post-processing step. Although the ultimate goal is to output the SMPL-X body mesh, the joint-based representation part is necessary, as the joint space is more straightforward and explicit for the motion learning, which in turn benefits the learning for SMPL-X parameters.

B.3. Motion Initialization

For experiments on PROX [3], we utilize the per-frame body regressor CLIFF [6] to estimate per-frame body poses from RGB input as initialization. In contrast to most existing human mesh regressors, which take as input only an image cropped around the human body, CLIFF incorporates information from the cropped bounding box (scale and location with respect to the original image) and the original image focal length. This approach yields improved predictions for global orientation,

particularly beneficial when the body is positioned at the boundary of the original input image. However, it is worth noting that CLIFF is trained on the SMPL body model. Consequently, we complementarily employ a SMPL-X based human mesh regressor, PIXIE [2], to estimate also SMPL-X body shape parameters $\boldsymbol{\beta}$. We then combine the pose from CLIFF and the shape from PIXIE. Additionally, to enhance global translation estimation, we leverage a skeleton-based 3D human pose regressor, MeTRAbs [10], which provides a better prediction for the absolute global position. We combine the global orientation and body pose obtained from CLIFF, the body shape derived from PIXIE, and the global translation estimated by MeTRAbs and use them as our per-frame initialization $\tilde{\boldsymbol{X}}$ for motion estimation from RGB videos on PROX.

For RGB-D sequences on PROX, we additionally perform a per-frame optimization step to incorporate depth observations. More precisely, for each frame, we optimize the SMPL-X body parameters $(\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\beta})$ by minimizing the following objective function:

$$\mathcal{L} = \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}}. \tag{1}$$

$\mathcal{L}_{\text{2D}}$ penalizes the 2D joint distances between the optimized 2D SMPL-X body joints projected onto the RGB image, and detections from OpenPose [1]. $\mathcal{L}_{\text{depth}}$ penalizes the 3D Chamfer distance between the human point cloud obtained from the depth frame and SMPL-X surface points visible from the camera as in [3, 13]. $\mathcal{L}_{\text{pose}}$ and $\mathcal{L}_{\text{shape}}$ denote priors that regularize SMPL-X body pose and shape. λs denote the corresponding weights. This approach is akin to VPoser-t but excludes the 3D joint smoothness term, working per-frame.

On EgoBody [14], to conduct a quantitative comparison with the baselines while factoring out the influence of various initialization strategies, we employ VPoser-t for initialization as in HuMoR [8]. Regarding the input OpenPose 2D detections for our method and baseline methods, instead of raw detections, we use a manually post-processed version provided by EgoBody, where the detections for most occluded joints are masked out.

It is worth highlighting that our approach can be combined with various initialization strategies (both optimization- and regression-based), ensuring flexibility for different applications and inputs.

B.4. Inference Details

Occlusion masks for reconstruction from RGB(-D) videos. To obtain joint occlusion masks for inference on PROX and EgoBody, given the initialized 3D body, we identify a body joint as occluded if it fulfills two conditions: (1) the confidence score of the corresponding 2D joint detection is below 0.2; and (2) the depth of the joint is greater than the depth of the scene vertex which is projected on the same 2D pixel in the image plane as the body joint, from the camera view. The depth of the joint is determined by rendering the 3D body mesh obtained from initialization from the camera view.

Score-guided sampling. In Eq. (14) in the main paper, we set λ_{2D} to 3e5. λ_{skate} is set to 1e5 for experiments on PROX and EgoBody, and to 3e6 for experiments on AMASS. The score-guided sampling is enabled for the last 100 denoising steps for PoseNet. Furthermore, as the modulation variance Σ_t diminishes towards the end of the diffusion denoising steps, we skip the last 20 denoising steps for PoseNet for experiments on PROX and EgoBody; this ensures stronger gradient guidance for 2D alignment with image observations.

Runtime. To assess the runtime difference between our method and HuMoR [8], we omit the initialization stage (as the runtime depends on the setup and the choice of the initialization method), and focus solely on the inference/test-optimization stage for both methods. For RGB-D input, employing an NVIDIA A100 GPU with a batch size of 10, and with a sequence length of 144 frames, our method completes the inference in 59 seconds, while HuMoR requires 30 minutes for the entire test-time optimization. We use the default configurations of the official HuMoR code. Note that the run time we report here differs from the ones in HuMoR paper due to the different hardware and setups (batch size and sequence length). Besides, we also report the inference time under the same setup as above for CLIFF/VPoser-t as 0.5sec, and 2.5min, respectively.

C. Baseline MDM++ Details

For motion infilling and in-between tasks, at each denoising step, MDM [12] and PriorMDM [11] replace denoised joints with visible input joints, when they are available. This assumes clean motion for visible body parts as input, and therefore cannot handle noisy scenarios like the ones we consider. Moreover, we observe that the relative trajectory representation in [11, 12], which only considers trajectory velocities, results in severe global trajectory drifting and deviation from the input, due to accumulated errors in the estimated trajectory velocities. To address these limitations and enable denoising together infilling and in-between tasks, we adapt the original MDM formulation to obtain MDM++, as explained below.

MDM++ shares a similar design with PoseNet (Fig. S1), but with two key distinctions. Firstly, MDM++ takes the initial noisy and incomplete motion $(\boldsymbol{M}_R \odot \tilde{\boldsymbol{R}}, \boldsymbol{M}_P \odot \tilde{\boldsymbol{P}})$ as the condition, and concurrently predicts both root trajectory $\hat{\boldsymbol{R}}_0$ and local body pose $\hat{\boldsymbol{P}}_0$. This means that, differently from [11, 12], MDM++ explicitly conditions on noisy motion by taking

noisy trajectory and local pose as input – thus enabling motion denoising at inference time. We train MDM++ with the same augmentation scheme as RoHM, see Sec. B.1. Secondly, MDM++ adopts the same motion representation as our method, as detailed in Sec. 4 of the main paper, incorporating both the absolute and relative representations for the root trajectory. This design choice significantly mitigates trajectory drifting issues.

However, addressing both denoising and infilling tasks in two different spaces (root trajectory and local pose) within one single model remains very challenging. Indeed, MDM++ still exhibits degraded reconstruction accuracy and motion plausibility compared to RoHM, as shown in Tab. 1 in the main paper.

D. More Results

Results on unoccluded data. Here we report RoHM reconstruction accuracy (MPJPE) for unoccluded frames (frames with all body joints visible) on the EgoBody subset (the same subset as used in Sec. 5 in the main paper) as 60.1mm, while HuMoR yields an MPJPE of 73.1mm for unoccluded data. This indicates that, apart from cases with both noise and occlusions, our reconstruction also outperforms the baseline in accuracy specifically for scenarios with noise only but without occlusions.
Cross-view results. Fig. S2 illustrates that the cross-view results are also plausible.

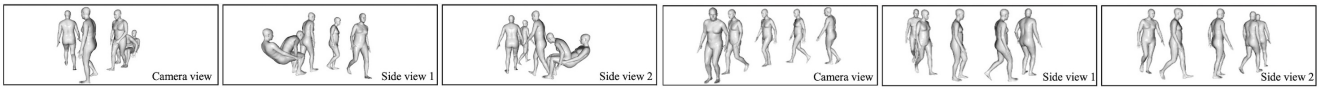

Figure S2. **Cross view results for two sample sequences.**

Thresholds for foot skating. To investigate the impact of different thresholds in the Skating metric, we evaluate the foot skating ratio on EgoBody with different velocity thresholds, and RoHM consistently outperforms baselines: VPoser-t / HuMoR / Ours: 0.19/0.17/**0.04** with threshold 15cm/s, 0.22/0.23/**0.08** with 10cm/s, 0.28/0.39/**0.23** with 5cm/s.
Non-determinism. Despite the stochastic nature of generative models, our reconstruction accuracy is not impacted by the non-determinism due to constraints posed by visible joints and the temporal span (144 frames). For the setup in Tab. 1 (Occ-L., noise level 3) in the main paper, different random seeds yield similar results: GMPJPE-*occ*/-*vis* are within 57.3∼57.7mm / constantly 21.8mm.

E. Limitations and Failure Cases

We show example failure cases in Fig. S3. As it is common for learning-based approaches, our method can struggle to generalize to out-of-distribution test cases – such as shapes and poses that are rarely seen in the training data. For instance, the first two columns of Fig. S3 show subjects that are relatively tall, and the last two columns show rare poses.

Another limitation lies in the model's dependence on both the 3D scene mesh and 2D joint detections to determine if a joint is occluded. This reliance becomes problematic when the 3D scene mesh is unavailable or when 2D joint detections are unreliable. A potential solution could involve learning an occlusion classifier based on the initial 3D body pose and image inputs to identify joint occlusions. We consider this avenue a promising direction for future exploration.

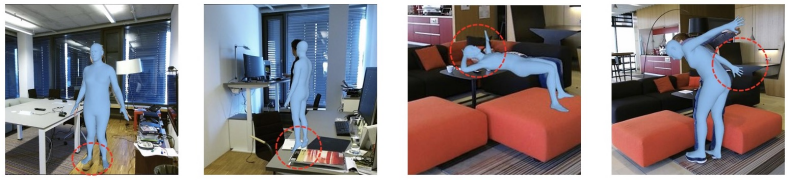

Figure S3. **Failure cases** with inaccurate estimations for out-of-distribution shapes (column 1, 2) and poses (column 3, 4).

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3

[2] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Collaborative regression of expressive bodies using moderation. In *2021 International Conference on 3D Vision (3DV)*, 2021. 3

[3] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*, 2019. 2, 3

[4] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1

[5] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *ICCV*, 2023. 2

[6] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 2

[7] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 2

[8] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. HuMoR: 3D human motion model for robust pose estimation. In *ICCV*, 2021. 3

[9] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, 2023. 1

[10] István Sárándi, Timm Linder, Kai O. Arras, and Bastian Leibe. MeTRAbs: metric-scale truncation-robust heatmaps for absolute 3D human pose estimation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):16–30, 2021. 3

[11] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023. 3

[12] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023. 1, 2, 3

[13] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *ICCV*, 2021. 3

[14] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In *ECCV*, 2022. 3