Supplementary Material for the paper: "Score-Guided Diffusion for 3D Human Recovery"

Anastasis Stathopoulos
Rutgers University

Ligong Han
Rutgers University

Dimitris Metaxas
Rutgers University

A. ScoreHMR Pseudo-code

In Section 4.1 of the main paper, we introduce Score-Guided Human Mesh Recovery (ScoreHMR). Here, we provide a pseudo-code implementation of ScoreHMR in Algorithm 1.

B. Implementation Details

In Section 4.2 of the main paper, we provide information about the design, architecture and training of the denoising model ϵ_ϕ. Here, we describe them in more detail. We also provide the hyper-parameters used for guidance. Our code and pre-trained model weights are released at https://github.com/statho/ScoreHMR.

Denoising model architecture. The architecture of the denoising model ϵ_ϕ is depicted in Figure 5. For each trainable layer we include the number of input and output features as $d_{in} \rightarrow d_{out}$. We use image features $\mathbf{c}$ from frozen HMR regression networks as discussed in the main paper. ProHMR [13] uses the standard ResNet-50 [7] backbone, and we use the features after the global average pooling layer, *i.e.* the dimension of $\mathbf{c}$ is 2048. PARE [12] learns disentangled features for the pose and shape SMPL parameters. We only used the pose features of PARE, so $\mathbf{c}$ is a 3072-dimensional vector.

Training details. The total number of timesteps in the diffusion model is set to $T = 1,000$ following prior work [5, 8]. We use cosine variance schedule [18]. We train with a batch size of 128, learning rate 10^{-4} and Adam optimizer [11] for 1M iterations. We maintain an exponential moving average (EMA) copy of our model with rate of 0.995. Our implementation is in PyTorch [19]. Training takes only 6 hours on a single NVIDIA A100 GPU.

Guidance details. The gradient step size in Eq. (8) is set to $\rho_{repr} = 0.003$, $\rho_{MV} = 0.005$ and $\rho_{temp} = 30$ for $\mathcal{L}_{repr}$, $\mathcal{L}_{MV}$ and $\mathcal{L}_{temp}$ respectively. The outer refinement loop is set to $S_{max} = 10$. The threshold for the early stopping criterion is set to $\lambda_{thr} = 10^{-5}$. The timestep (noise level) where there refinement process starts is set to $\tau = 50$ and the DDIM step size is set to $\Delta t = 2$. For multi-view refinement experiments we set $\tau = 100$ and $\Delta t = 10$.

C. Ablations

Here, we provide an ablation study of the two core components of score guidance. The ablation study is performed on the 3DPW test set in the model fitting setting, starting from the regression estimate of HMR 2.0b with 54.3 PA-MPJPE. The default setting is marked with gray. All other components are set to their default values during each component's individual ablation.

Noise level τ. The Table below shows the PA-MPJPE error varying τ. ScoreHMR works better for small noise levels t. The one-step denoised result $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ used to compute the guidance loss Eq. (10) is more accurate for small values of $t \in [0, \tau]$.

τ	50	100	200	300
HMR 2.0b + ScoreHMR	51.1	52.3	54.3	54.5

DDIM step size Δt. The Table below shows the PA-MPJPE error varying the DDIM step size Δt. Even though larger DDIM step sizes result in lower PA-MPJPE in 3DPW, we find that ScoreHMR with a small step size is more robust and performs better qualitatively especially for challenging and unusual poses. A similar observation is made in [6], where HMR 2.0b has a higher PA-MPJPE error than HMR 2.0a on 3DPW and Human3.6M, but performs better in practice.

Δt	2	4	6	8	10	12
HMR 2.0b + ScoreHMR	51.1	49.6	48.8	48.4	48.2	48.4

D. Datasets

In this part we offer some information on the datasets used for training and evaluation. The datasets used for training are Human3.6M [9], MPI-INF-3DHP [17], COCO [16] and MPII [1]. The datasets used for evaluation are 3DPW [20], EMDB [10], Human3.6M [9] and Mannequin Challenge [15].

Human3.6M. It contains data for 3D human pose captured in a studio environment. Following standard practices we use subjects S1, S5, S6, S7 and S8 for training, while we

Algorithm 1 Score-Guided Human Mesh Recovery (ScoreHMR)

Input: Given observation $\mathbf{y}$, denoising model ϵ_ϕ, image features $\mathbf{c}_I$, estimate $\mathbf{x}_{reg}$ from a regression network, gradient step size ρ, noise level τ, DDIM step size Δt, threshold λ_{thres}, number of iterations for the outer refinement loop S_{max}.

for $s = 1$ to S_{max} **do**
  if $s = 1$ **then**
    $\mathbf{x}_{init} \leftarrow \mathbf{x}_{reg}$ ▷ First iteration starts with estimate from regression
  else
    $\mathbf{x}_{init} \leftarrow \mathbf{x}_0$ ▷ Iteration starts with $\mathbf{x}_0$ from previous iteration
  end if
  $\mathbf{x}_\tau = \text{DDIMInvert}(\mathbf{x}_{init}, \mathbf{c}_I)$ ▷ Run DDIM inversion until noise level τ
  for $t = \tau$ to Δt with step size Δt **do**
    $\tilde{\epsilon} \leftarrow \epsilon_\phi(\mathbf{x}_t, t, \mathbf{c}_I)$ ▷ Predict noise
    Initialize computational graph for $\mathbf{x}_t$
    $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \sqrt{1-\alpha_t})\tilde{\epsilon}$ ▷ Predict one-step denoised result
    $\mathcal{L}_g \leftarrow ||\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)||^2$ ▷ Compute guidance loss
    if $\mathcal{L}_g < \lambda_{thres}$ **then**
      return $\hat{\mathbf{x}}_0$ ▷ Early stopping: return $\mathbf{x}_0$ if the loss is below a threshold
    end if
    $\tilde{\epsilon}^{'} \leftarrow \tilde{\epsilon} + \rho\sqrt{1-\alpha_t}\nabla_{\mathbf{x}_t}\mathcal{L}_g$ ▷ Compute modified noise after score-guidance
    $\hat{\mathbf{x}}_0^{'} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \sqrt{1-\alpha_t})\tilde{\epsilon}^{'}$ ▷ Predict one-step denoised result with modified noise
    $\mathbf{x}_{t-\Delta t} \leftarrow \sqrt{\alpha_{t-\Delta t}}\hat{\mathbf{x}}_0^{'} + \sqrt{1-\alpha_{t-\Delta t}}\tilde{\epsilon}^{'}$ ▷ DDIM sampling step
  end for
end for
return $\hat{\mathbf{x}}_0^{'}$

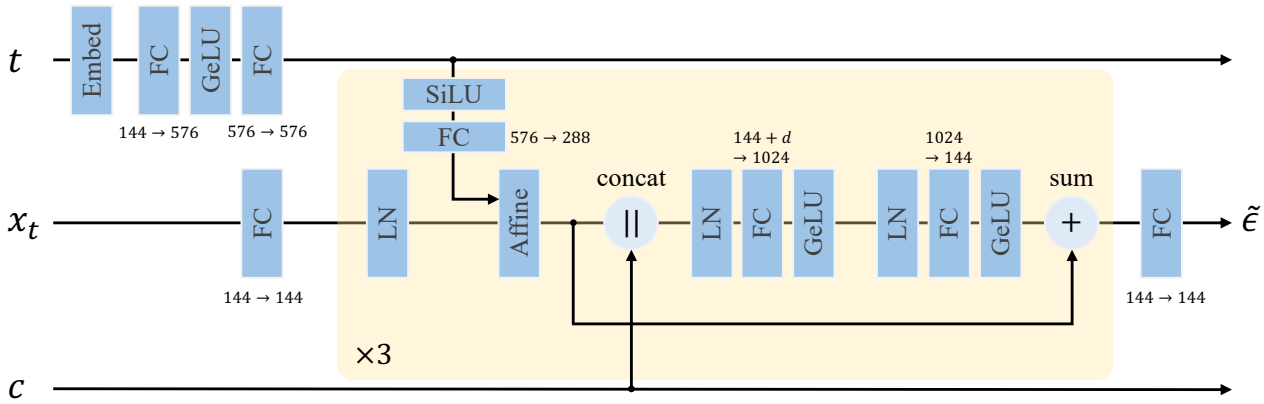

Figure 5. **Diffusion model architecture.** Implementation of $\epsilon_\phi(\mathbf{x}_t, t, \mathbf{c} = g(I))$. *LN* denotes Layer Normalization [3], $\|$ denotes concatenation, and d denotes the dimension of the image features $\mathbf{c}$. Rotations are parameterized with 6D representations, thus $\mathbf{x}_0, \mathbf{x}_t, \tilde{\epsilon}$ are 144-D vectors.

use subjects S9 and S11 for evaluation in the multi-view refinement setting.

MPI-INF-3DHP. It contains data for 3D human pose captured mainly in indoor studio environments with a markerless setup. We use the predefined train split for training.

COCO. It contains images in-the-wild annotated with 2D keypoints. We use this dataset only during training.

MPII. It contains images annotated with 2D keypoints. We use this dataset only during training.

3DPW. It is a dataset captured in indoor and outdoor locations and contains SMPL pose and shape ground-truth. We follow standard practices in the literature and only use the predefined test split for evaluation.

EMDB. It is a new dataset captured in indoor and outdoor locations and contains SMPL pose and shape ground-truth. It includes a split (*i.e.*, EMDB 1) with the most challenging outdoor sequences. We use EMDB 1 for evaluation.

Mannequin Challenge. It contains videos of people staying frozen in various poses. We use the SMPL annotations from [14] for evaluation in this dataset.

Figure 6. **Model fitting results.** We compare our approach (green) with ProHMR-fitting (blue) and SMPLify (grey). All model fitting algorithms are initialized with regression from ProHMR (pink) or HMR 2.0b (white).

	3DPW (14)	EMDB 1 (24)
HMR 2.0b [6]	54.3	78.7
+ ScoreHMR w/o β	**51.1**	76.6
+ ScoreHMR w/ β	**51.1**	**76.5**

Table 5. ScoreHMR with and without the inclusion of SMPL shape parameters β. Numbers are PA-MPJPE in mm. Parenthesis denotes the number of body joints used to compute PA-MPJPE.

E. Evaluation Metrics

Depending on the setting, we evaluate using the MPJPE, PA-MPJPE and Acc Err metrics following standard practices in the literature. The Mean Per Joint Position Error (MPJPE) computes the Euclidean error between the predicted and ground-truth 3D joints, after aligning them at the pelvis. The PA-MPJPE compute the same error after aligning the predicting the ground-truth 3D joints with Procrustes alignment. Both metrics are used for per-frame 3D human pose evaluation. The acceleration error (Acc Err) is a temporal metric that measures the average difference between ground truth 3D acceleration and predicted 3D acceleration of joints in mm/s^2.

F. Additional Quantitative Evaluation

Diffusion Model for SMPL β. As we discuss in Section 4.2 of the main paper, ScoreHMR can also accommodate the SMPL shape parameters β. In Table 5 we present results in single-frame model fitting to 2D keypoint detections, comparing ScoreHMR with and without the inclusion of SMPL β. We observe that modeling and optimizing β with our proposed approach works well, but does not bring any significant performance improvement compared to modeling only the SMPL pose parameters θ.

Refinement from HMR 2.0a. The Table below shows the PA-MPJPE of model fitting on 3DPW test set, starting from HMR 2.0a regression. Only ScoreHMR can quantitatively improve the performance of HMR 2.0a (by 4.5%).

HMR 2.0a	+ScoreHMR	+ProHMR-fitting	+SMPLify
44.5	**42.5**	54.9	52.5

G. Additional Qualitative Results

In Figure 6 we include additional qualitative examples of model fitting, comparing our proposed approach with ProHMR-fitting [13] and SMPLify [4]. Our approach

achieves more faithful reconstructions than the baselines. We observe that in the case of missing keypoint detections (*e.g.*, example with truncation in last row) SMPLify results in body orientation errors.

In Figure 8 we illustrate the effectiveness of our approach in consolidating information from multiple views in order to improve the 3D pose of a human. The initial view (first row of Figure 8) presents challenges with occluded hands, resulting in inaccurate pose estimate for the hands. Multiple view fusion with our proposed approach results in a more faithful estimation of the true pose.

We present some failure cases of our method in Figure 7. Our approach can fail when there are wrong keypoint detections. Optimization-based methods fail in that case too as we show in Figure 7.

Finally, we demonstrate our approach on video sequences from the validation split of PoseTrack [2] and others. We use predicted tracks from 4DHumans [6]. We encourage viewing video results on the project page.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 1

[2] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 4

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[4] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016. 3

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 1

[6] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *ICCV*, 2023. 1, 3, 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[9] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 2014. 1

[10] 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 *ICCV*, 2023. 1

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[12] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *ICCV*, 2021. 1

[13] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *ICCV*, 2021. 1, 3

[14] Vincent Leroy, Philippe Weinzaepfel, Romain Brégier, Hadrien Combaluzier, and Grégory Rogez. Smply benchmarking 3d human pose estimation in the wild. In *3DV*, 2020. 2

[15] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019. 1

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[17] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017. 1

[18] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 1

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 1

[20] 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 *ECCV*, 2018. 1

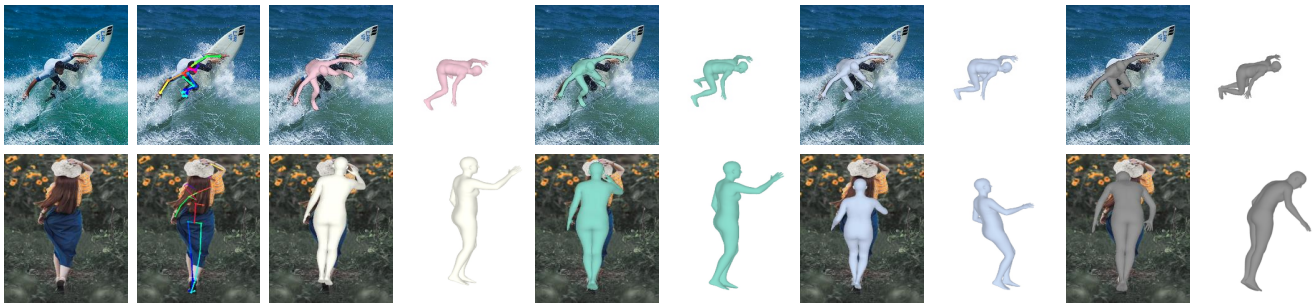

Figure 7. **Failure cases of model fitting.** Pink: ProHMR regression. White: HMR 2.0b regression. Green: Regression + ScoreHMR (ours). Blue: Regression + ProHMR-fitting. Grey: Regression + SMPLify. While all methods encounter challenges when incorrect keypoints are detected, our image-conditioned diffusion model tries to keep the 3D pose aligned with the available image evidence.

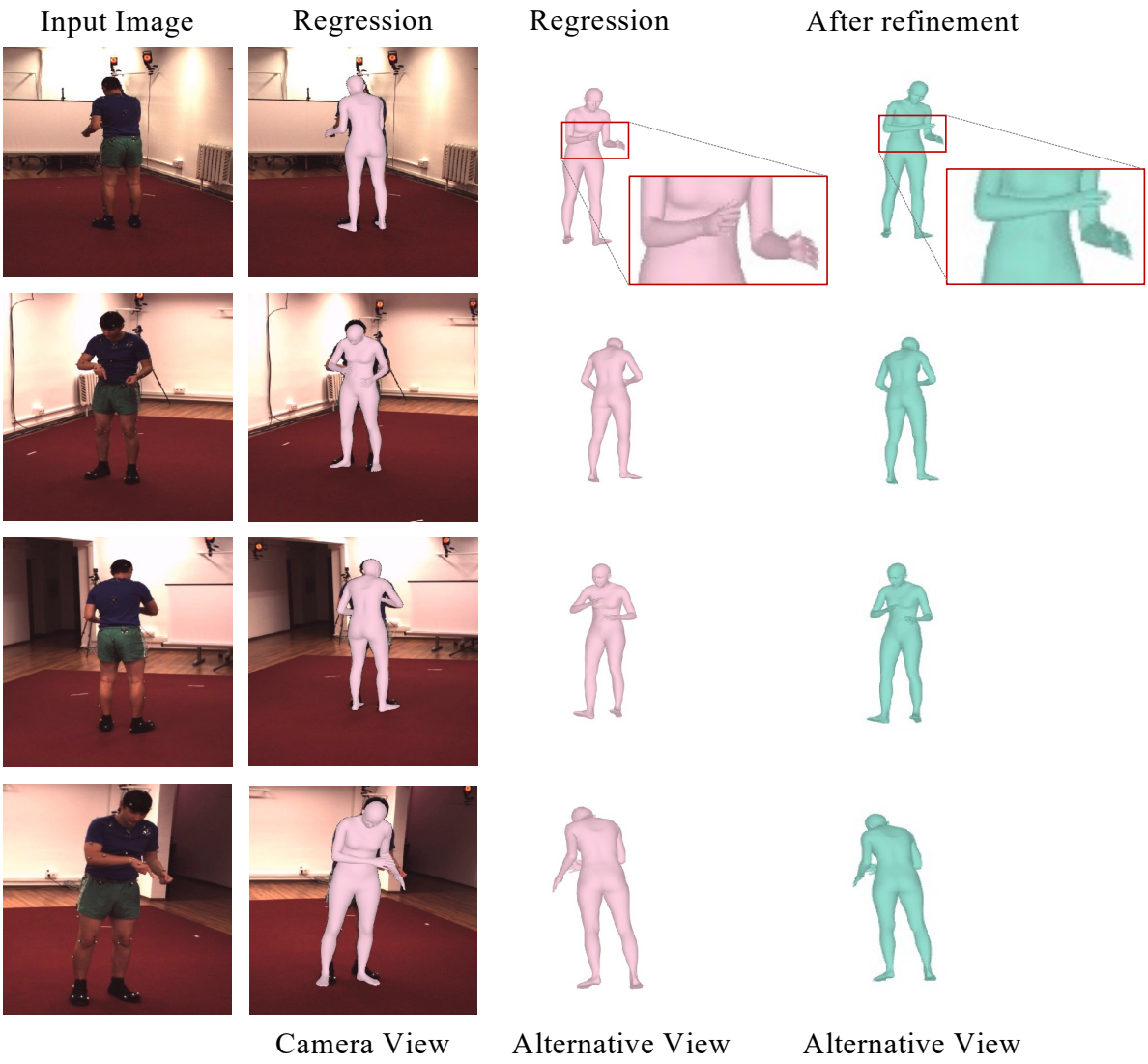

Figure 8. **Multi-view refinement.** Refinement with multiple views fixes the 3D pose of the right hand, which is self-occluded in the first view (first row).