

Noise-in, Bias-out: Balanced and Real-time MoCap Solving

Supplementary Material

Georgios Albanis ^{1,2}

Nikolaos Zioulis ¹

Spyridon Thermos ¹

Anargyros Chatzitofis ¹

Kostas Kolomvatsos ²

¹Moverse {giorgos,nick,spiros,argyris}@moverse.ai

²Dept. of Informatics and Telecommunications, University of Thessaly kostasks@uth.gr

Contents

Section	Page
1. Intro	1
2. MoCap Solving Model	2
3. Pre-processing	2
3.1. Augmentations	2
3.2. Corruption	2
4. Datasets	3
4.1. Marker-based	3
4.2. Markerless	3
4.3. Long-Tail	4
4.4. Qualitative Distribution	4
5. Performance Metrics & Indicators	4
5.1. MoCap Metrics	4
5.2. Synthesis Metrics	4
5.3. Performance Indicators	4
6. Training Data Sourcing	5
7. Balancing Regression	5
7.1. Robust VPoser	5
7.2. Relevance Function	6
7.3. Orthogonality Investigation	7
7.4. Sampling Ablation	8
8. Extra Solving Experiments	8
9. Landmarks and fitting ablation	8
10 Additional Qualitative Results	9
11 System Details	9

1. Intro

In this supplementary material we provide additional quantitative and qualitative results to accompany the main paper. In addition, a set of ablation studies are presented to offer extra insights into the inner workings of the methods and techniques presented in the main paper. Finally, due to the lack of space in the main paper, we provide more details with respect to the implementation of the proposed models, the experimental protocol with respect to the datasets and metrics that were used, visualizations of related data points, and details regarding the experiments comparing to the state-of-the-art. It should be noted that no additional training or optimization was performed in any of these experiments with respect to that presented in the main paper.

Along with this supplementary material, we share a [short video](#) that showcases the real-time performance of our MoCap system in a challenging input context, captured with only 3 Microsoft Kinect for Azure sensors.

Sec. 2 provides the implementation details of the UNet model used to predict landmarks ℓ_{est} . Sec. 3 clarifies the augmentation and corruptions used when training and when experimenting with noisy fits. Sec. 4 presents the different datasets that were used for the main paper’s experiments and the accompanying experiments found in this supplementary material. Sec. 5 defines the metrics used to evaluate performance and the performance indicators used to select the best performing models.

Following the experimental results structure of the main paper, the remaining sections supplement the already presented analysis with additional experiments, results and insights. Sec. 6 provides visualizations comparing the distribution of the markerless and marker-based data used to assess the efficacy of the former as a training corpus. Complementary experiments are also presented to support the main paper claims. Sec. 7 provides further analysis with respect to the inner workings of the balanced regression approach presented in the main paper, specifically, the VAE model’s

details (Sec. 7.1), a relevance function ablation (Sec. 7.2), an investigation of the orthogonality between the different techniques (Sec. 7.3), and an ablation of the different sampling components (Sec. 7.4). Sec. 8 presents an extra experiment supplementing the solving comparison experiment conducted in the main paper. Sec. 9 offers extra insights with respect to the landmarks regressed by our model, by ablating the fitting process across various noise levels and input landmark types. Finally, Sec. 10 includes additional qualitative results, while Sec. 11 describes the implementation details related to the real-time MoCap system used to capture and provide in-the-wild results.

2. MoCap Solving Model

Our proposed model is designed to work with any method capable of inferring markers and joints from an input markers' point cloud. However, for the presented study, we utilized a light-weight convolutional model that can preserve high resolution outputs, exploiting the quasi-autoencoding nature of regressing pre-defined markers (and, when applicable, joints) from unstructured marker position inputs. Specifically, a modified version of the UNet [26] architecture was used to simultaneously predict 53 markers and 18 joints landmarks. It should be noted that since MoCap-Solver [8] was trained with 56 markers and 24 joints on the CMU data, for the experiment comparing direct solving performance, our model was adapted to the same outputs. The model consists of 5 convolutional blocks, with each block consisting of 32, 64, 128, 256, and 512 features, respectively. Each encoder block comprises 2 convolution layers, with a kernel size of 3, a stride and padding of 1, followed by ReLU activations and batch normalization [14]. When downscaling anti-aliased max pooling [31] is used, while upscaling uses bilinear interpolation. The bottleneck of the model consists of a single convolution block, utilizing the same parameters as the encoder blocks. The decoder includes the same convolution blocks, and the output of each block is concatenated with the corresponding encoder's output. Finally, the prediction layer consists of a convolution block with a kernel size of 1, a stride of 1, and padding of 0, activated by the ReLU function. Training runs for 30 epochs with a batch size of 16, a learning rate of 2×10^{-4} accompanied by a step-wise schedule reducing it to 95% every 4 epochs.

As mentioned in the main paper the model is supervised by the following loss summed over all landmarks (batch notation is omitted for brevity):

$$\mathcal{L} = \sum_{i=1}^L (\lambda_{JS} \mathcal{L}_{JS}(\mathbf{H}_{gt}, \mathbf{H}_{est}) + \lambda_w \mathcal{L}_w^\nu(\tilde{\ell}_{gt}, \tilde{\ell}_{est})). \quad (1)$$

\mathcal{L}_{JS} is the Jensen-Shannon divergence defined in Eq. (2):

$$\mathcal{L}_{JS}(\mathbf{H}_{gt}, \mathbf{H}_{est}) = \frac{1}{2} D_{KL}(\mathbf{H}_{gt}, M) + \frac{1}{2} D_{KL}(\mathbf{H}_{est}, M), \quad (2)$$

where D_{KL} is the Kullback-Leibler divergence, $M = \frac{1}{2}(\mathbf{H}_{gt} + \mathbf{H}_{est})$ is the average of \mathbf{H}_{gt} and \mathbf{H}_{est} .

\mathcal{L}_w^ν is the robust Welsch penalty function, applied to the normalized ℓ coordinates, defined by Eq. (3), with $\nu > 0$ being a user-specified parameter set to 0.05:

$$\mathcal{L}_w^\nu(\tilde{\ell}_{gt}, \tilde{\ell}_{est}) = 1 - \exp\left(-\frac{|\tilde{\ell}_{gt} - \tilde{\ell}_{est}|^2}{2\nu^2}\right) \quad (3)$$

3. Pre-processing

We use a pre-processing pipeline to augment and then corrupt the input training data. Augmentations exploit the parametric nature of the data to increase their variance. Similar to [13, 11, 8], corruption exploits the simple and synthetic nature of motion capture (MoCap) to closely approximate real-world MoCap settings with noisy inputs and marker-/viewpoint- related artifacts like ghost markers, occluded markers, and varying levels of measurement noise.

3.1. Augmentations

First, we perform an augmentation to account for subject body shape variations. A two-step process is employed that starts with a controlled shifting of the shape coefficients, with random values u sampled from a uniform distribution $u \sim \mathcal{U}(-1, 1)$:

$$\beta' = \beta + u \quad (4)$$

Then, a small random subset of the shape coefficients are randomly sampled from a normal distribution:

$$\beta'_i = \begin{cases} \beta_i, & \text{if } i \notin S \\ \mathcal{N}(0, 1), & \text{if } i \in S \end{cases} \quad (5)$$

where S is a set of n' indices sampled uniformly from the set of indices, with our experiments randomly shifting between $[0, 2]$ coefficients.

Then, using the rotation symmetry of the body, we randomly perform a handedness flipping augmentation by flipping the parameters of the left/right arms/legs.

3.2. Corruption

We simulate marker occlusions with the following process. Let $\mathbf{p} = (p_1, p_2, \dots, p_n)$ be the vector of marker positions, where p_i is the position of the i -th marker. We randomly select a subset of markers for occlusion by determining the number of markers to be occluded, denoted as k . We draw a random sample from a discrete uniform distribution to determine k , $k \sim \mathcal{U}(m, n')$, $m \leq k \leq n' \leq n$, where $\mathcal{U}(m, n')$ is the uniform distribution over the range



Figure 1: A set of random samples from the THuman2.0 [30] dataset. The darker meshes indicate more challenging poses.

of integers $\{m_1, m_2, \dots, n'\}$, and n' defines the maximum number of markers to be occluded. Next, we draw another random sample from a uniform distribution to determine the indices of the markers to be occluded, *i.e.* $\mathbf{m} = (m_1, m_2, \dots, m_k) \sim \mathcal{U}(1, n)$, $k \leq n$ where $\mathcal{U}(1, n)$ is the uniform distribution over the markers' set of indices. The resulting vector m contains the indices of the markers to be occluded and is used to exclude these markers from \mathbf{p} .

As a next step, the ghosting of markers is emulated by extracting samples from a Gaussian distribution with mean and standard deviation values equivalent to the original marker positions, following [11]. In more detail, we first compute the median position for each spatial dimension of the marker positions, μ_j , (*i.e.* the median value for the j -th spatial dimension of the marker positions), and the sample covariance matrix Σ . We then draw samples $g \sim \mathcal{G}(\mu, \Sigma)$, which are appended to the original markers' positions \mathbf{p} .

Finally, to simulate marker noise, we randomly select a set of markers to shift and generate a random offset for each selected marker. Particularly, with N being the number of markers to shift, and M being the maximum allowable shift distance, we randomly sample from a uniform distribution to determine the indices of the markers to which the noise will be added $I \sim \mathcal{U}(1, N)$. For each index $i_j \in I$, we generate a random offset vector $o \sim \mathcal{U}(-M, M)$, and add this offset to the original marker position to obtain the noisy position $\mathbf{p}' = \mathbf{p} + \mathbf{o}$.

The proposed preprocessing pipeline is randomly applied in each epoch, with specific probabilities assigned to each of the augmentation and corruption functions. In more detail, we apply the aforementioned augmentation functions with 0.5 probability each, meaning that they will be applied to half of the instances of input data. Similarly, we apply the ghosting and occlusion corruption functions with 0.7 probability, while the shifting one with 0.8.

4. Datasets

4.1. Marker-based

For our experiments we used a variety of MoCap datasets unified within AMASS [20] to body model parameters. The datasets we use for our experiments include the CMU dataset, which is one of the largest motion capture datasets containing a wide variety of motion types, such as walking, running, dancing, and more. We also use the Transitions dataset, which focuses on the transitions between different activities, such as sitting down and standing up, or picking up and carrying an object. Additionally, we use the PosePrior dataset developed by [2] to train a statistical model of human pose, the HumanEva dataset [27], which includes various activities performed by multiple subjects, and the ACCAD dataset [1], consisting of more action motion types such as dancing, martial arts, and sports. Moreover, we use the TotalCapture dataset [15], which includes data from 5 different subjects performing 37 motion actions, the DFaust dataset [4] that includes motion data from 10 subjects performing 129 different types of motion, and the CNRS dataset consisting of data from 2 subjects performing 79 different motions.

4.2. Markerless

Apart from these, which were all acquired with high-end marker-based optical MoCap systems, we additionally use a number of datasets that were collected with markerless methods, using body models and fitting them to observations. These include the THuman 2.0 [30] dataset, including 5 subjects in extreme poses, the GeneBody dataset [9] consisting of 50 subjects performing various short duration activities, and the ZJU-MoCap dataset [24] that includes data from 10 sequences of human performances. Fig. 1 depicts an indicative subset from the THuman 2.0 dataset, which consists of both common and challenging-



Figure 2: Exemplar rare and complex poses from our custom tail dataset.

	Subjects	Activities	Minutes
ACCAD	20	14	26.74
CMU	111	25	543.49
CNRS	2	2	9.91
DFaust	10	12	5.72
HumanEva	3	5	8.47
PosePrior	3	10	20.82
TotalCapture	5	12	41.10
Transitions	1	4	15.10
THuman 2.0	10	-	-
Genebody	50	50	8.33
ZJUMoCap	24	10	14.40

Table 1: Datasets overview.

to-understand poses (shown with darker meshes).

4.3. Long-Tail

We have manually curated a small test set comprising 274 challenging poses, including extreme and rare ones, and was used as our “Tail” dataset for assessing long-tail regression performance. These were coarsely grouped into 4 categories, “crossed legs”, “crossed arms”, “kicks” and “crouching”. Indicative examples are shown in Fig. 2.

4.4. Qualitative Distribution

An overview of these datasets in terms of some qualitative variance indicators is presented in Tab. 1. These were used to select by approximately equalizing the datasets used in the markerless vs optical data study.

5. Performance Metrics & Indicators

5.1. MoCap Metrics

For evaluating our model’s performance we resort to common metrics used in previous works as the root mean squared error (RMSE), defined below:

$$RMSE = \frac{1}{N} \sum_{i=1}^N \sqrt{\frac{1}{J} \sum_{j=1}^J \|\ell_{gt}^{(i,j)} - \ell_{est}^{(i,j)}\|_2^2}, \quad (6)$$

with N being the number of samples in the dataset, and J is the number of joints in each sample. We follow the same notation for all the equations below.

Apart from RMSE, we use a PCK-like metric (*i.e.* distance accuracy metric), which measures the percentage of predicted keypoints that fall within a certain distance threshold τ from their ground-truth positions:

$$PCK = \frac{1}{N} \sum_{i=1}^N \frac{1}{J} \sum_{j=1}^J [\|\ell_{gt}^{(i,j)} - \ell_{est}^{(i,j)}\|_2 < \tau]. \quad (7)$$

In our experiments, we used three variants of PCK, namely PCK1, PCK3 and PCK7 with τ set to 10mm, 30mm, and 70mm accordingly.

Finally, we use an angular metric defined in Eq. (8):

$$MAE = \frac{1}{N} \sum_{i=1}^N \frac{1}{J} \sum_{j=1}^J d(R_{gt}^{(i,j)}, R_{est}^{(i,j)}), \quad (8)$$

where d is the geodesic distance between each joint’s rotation matrix R_{gt}^i and R_{est}^i .

5.2. Synthesis Metrics

Inspired by C. Guo *et al.* [12], we use two metrics to choose our best model for tail-pose generation and regression regularization, measuring quality and evaluating diversity. Regarding quality, we extract features from 1052 generated and real samples and compute the Fréchet Inception Distance (FID) between the feature distribution of the generated pose and poses from the THuman 2.0 test set that serve as the “real” poses. To evaluate the diverse generation capability of our generative model, we generate and re-encode 1052 samples which are then split into two subsets of the same size $N = 526$. The diversity (DIV) is defined as the Euclidean norm of the distance between these two subsets as follows:

$$DIV = \frac{1}{N} \sum_{i=1}^N \|v_i - \tilde{v}_i\|, \quad (9)$$

where v and \tilde{v} correspond to re-encoded samples as vectors from a different subset.

5.3. Performance Indicators

The plethora of metrics makes it harder to find the best-performing model. To that end, we introduce a set of performance indicators, which essentially combines an error and an accuracy metric. Specifically, for the MoCap metrics we introduce *rmse3* indicator, defined in Eq. (10):

$$rmse3 = (1 - PCK3) \times RMSE, \quad (10)$$

Regarding the generative model performance, we choose our best-performing model using the indicator defined as:

$$synthesis = \frac{FID}{DIV}. \quad (11)$$

6. Training Data Sourcing

Tab. 2 presents a more extensive set of experiments for the markerless vs marker-based training data study where the models are also evaluated on our “Tail” test set. Extra experiments are also included, namely another variant of the markerless model that was additionally trained with the ZJU-MoCap data apart from GeneBody and THuman2.0 (i.e. Markerless#2), and another variant of the optical data, Optical#4 trained only on the CNRS dataset.

As in the main paper, we observe that even though the best performance is offered by an optical MoCap dataset combination, the markerless alternative is close in performance and surpasses some marker-based dataset combinations. Essentially, the quality of the data acquisition method does not seem to play a big part in the performance of the model, but instead the variance of the samples seems to be the largest performance denominator.

To supplement this point, Fig. 3 offers comparative visualizations of the encoded pose parameters θ vectors’ distribution for each dataset combination.

7. Balancing Regression

7.1. Robust VPoser

G. Pavlakos *et al.* [23] were the first to leverage a Variational Autoencoder (VAE) [16] instead of Gaussian mixture models to learn a pose prior by folding axis-angle embeddings around a Gaussian distribution. Apart from VAEs, pose - and by extension, motion-priors have been learned using other generative models [10] or by mapping the pose space on a surface-like manifold [29]. However, in this paper, we choose to focus on autoencoding generative models, as the trained model operates as a rare pose generator, as well as to reconstruct poses and providing input to the relevance function of our balanced regression model (see Section 3.1 of the main paper).

As noted in the works above, VAEs have certain drawbacks; due to the lack of other constraints. The learned prior tends to be mean-centered while the manifold “folded” around the Gaussian includes several “dead” regions that could lead to non-plausible data generation. These drawbacks would make a fitting process hard as the prior would serve as a regularizer. However, we choose to focus on the controllable generation of tail samples, as well as the use of the VAE for re-weighting each sample’s contribution to the batch loss during training. That is, we focus our experiments on comparing our VPoser variant termed Robust

		RMSE ↓	PCK1 ↑	PCK3 ↑	PCK7 ↑
ACCAD	Optical#1	50.40 mm	36.14%	84.89%	90.90%
	Optical#2	89.99 mm	41.11%	81.18%	86.24%
	Optical#3	92.90 mm	39.16%	79.74%	86.08%
	Optical#4	118.2 mm	26.21%	64.70%	79.64%
	Markerless#1	59.40 mm	21.70%	79.96%	90.08%
	Markerless#2	57.40 mm	24.75%	80.86%	90.40%
Tail#1	Optical#1	23.80 mm	17.04%	86.67%	99.26%
	Optical#2	37.50 mm	19.26%	76.30%	95.56%
	Optical#3	41.30 mm	17.04%	70.74%	94.81%
	Optical#4	116.8 mm	5.55%	44.07%	70.74%
	Markerless#1	33.50 mm	12.59%	82.96%	98.52%
	Markerless#2	28.85 mm	20.00%	87.77%	98.14%
Tail#2	Optical#1	26.70 mm	15.26%	84.33%	97.55%
	Optical#2	57.70 mm	13.89%	71.27%	89.84%
	Optical#3	72.80 mm	14.64%	67.16%	86.48%
	Optical#4	123.8 mm	5.16%	44.63%	71.54%
	Markerless#1	29.50 mm	13.43%	82.34%	97.68%
	Markerless#2	33.70 mm	18.19%	82.11%	95.11%
Tail#3	Optical#1	71.40 mm	13.89%	57.78%	82.22%
	Optical#2	300.0 mm	3.33%	10.56%	19.44%
	Optical#3	300.1 mm	0.5%	10.56%	17.22%
	Optical#4	309.1 mm	0.5%	6.67%	12.78%
	Markerless#1	222.0 mm	2.22%	22.78%	40.56%
	Markerless#2	248.0 mm	2.22%	16.11%	30.33%
Tail#4	Optical#1	68.30 mm	11.30%	59.90%	88.36%
	Optical#2	280.2 mm	7.00%	37.87%	60.58%
	Optical#3	343.5 mm	6.43%	36.91%	60.77%
	Optical#4	374.4 mm	4.07%	20.25%	36.33%
	Markerless#1	76.60 mm	10.68%	58.65%	86.71%
	Markerless#2	77.56 mm	13.10%	62.90%	89.23%

Table 2: Markerless vs optical data tested on ACCAD and tail test sets. Models trained on data sourced from a multi-view markerless fitting process perform on par with models trained on high-quality Optical data.

VPoser (RVPoser) with the model from [23] for tail-sample generation.

Our RVPoser follows a similar structure to the VPoser’s, with 3 main differences: a) we do not use batch normalization [14] prior to the first fully-connected layer of the encoder, b) we do not use any dropout layers in the decoder, and c) we do not use any activation function after the last fully-connected of the decoder. We train RVPoser using the CMU, Transitions, and PosePrior datasets, while our total training loss can be decomposed into the following losses:

$$\mathcal{L}_{VAE} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{orth} \quad (12)$$

$$\mathcal{L}_{KL} = \Psi(D_{KL}(q_\theta(z|R) || \mathcal{N}(0, I))) \quad (13)$$

$$\mathcal{L}_{rec} = \|v - \hat{v}\|_2, \quad (14)$$

$$\mathcal{L}_{orth} = \frac{Trace(R^T \hat{R}) - 1}{2}, \quad (15)$$

where $z \in R^{32}$ is the 32-dim latent code, $R \in \mathbb{SO}(3)^P$ is the rotation matrix for each pose parameter P , while

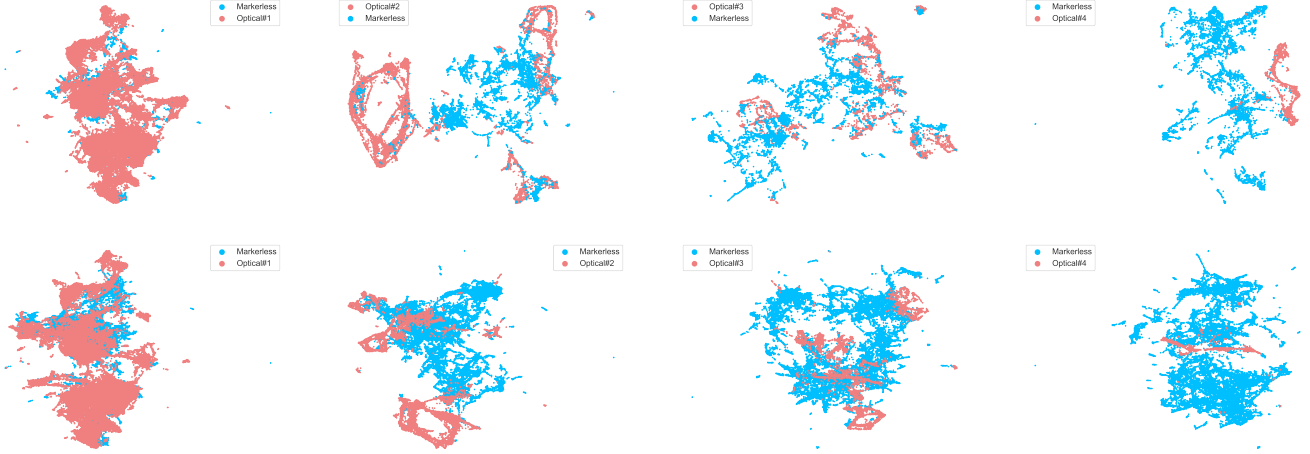


Figure 3: UMAP projections [21] on datasets collected using high-end MoCap systems and others collected from a multiview markerless fitting process. The first row uses the markerless#1 dataset and the second row uses the markerless#2 dataset. It can be seen that the variability of data is independent of the type of acquisition.

\hat{R} is the rotation matrix output of the decoder. v, \hat{v} correspond to the predicted and ground truth vertices, indicating that the reconstruction term incorporates both angular and 3D joint-position errors. Instead of using solely the Kullback-Leibler (KL) divergence, we regularize it (as in [32]) using the Charbonnier penalty function Ψ , with $\Psi(x) = \sqrt{1 + x^2} - 1$ [6] to prevent posterior collapse and learn a more disentangled manifold. Eqs. (13) and (14) follow the VAE training scheme - *e.g.*, trading of reconstruction quality with learning a Gaussian-like manifold, while Eqs. (14) and (15) force the model to construct a valid rotation latent space. We complement RVPoser training with the weight-decaying version of Adam optimization [19], which penalizes large weights and prevents over-fitting.

We choose to evaluate the 2 models on two different settings: a) compare the models in the task of generating realistic and diverse poses, and b) compare the models as priors for the task of fitting human body parameters. We evaluate both tasks on unseen data from the THuman 2.0 dataset which comprises diverse samples with challenging poses. From the results presented in Tab. 3, we observe that RVPoser is able to generate more diverse and faithful poses, while also outperforming VPoser in the fitting task, improving the overall angular error and the pose prediction accuracy (except for PCK7). Apart from the quantitative results, in Fig. 4 we show the UMAP projection [21] of 1200 ground truth pose vectors superposed on 1200 generated ones using VPoser and RVPoser. Based on the depicted result, the samples generated with our VAE variant cover significantly more space spanned by the ground truth embeddings. That is, our prior can generate more diverse - but still plausible - samples compared to VPoser.

	Synthesis		Fitting			
	FID \uparrow	DIV \uparrow	MAE \downarrow	PCK1 \uparrow	PCK3 \uparrow	PCK7 \uparrow
VPoser [23]	7.94	12.11	2.68°	28.83%	89.04%	99.03%
RVPoser (Ours)	8.57	14.24	1.51°	53.72%	94.57%	98.15%

Table 3: Quantitative comparison between the VPoser model from [23] and our robust variant (RVPoser) in synthesis and fitting on the THuman 2.0 test set.

7.2. Relevance Function

As stated in the main paper, bias in sample reconstructability can be used to assign relevance to each sample as more challenging (tail) poses are hard to reconstruct accurately. As relevance ρ , we define the weight used to scale the contribution of each pose to the batch-wide loss. That is, we need to increase the contribution of the tail poses to the batch loss for every iteration to mitigate the regression bias due to the high number of mean-like poses in our training set. We have experimented with 2 different relevance

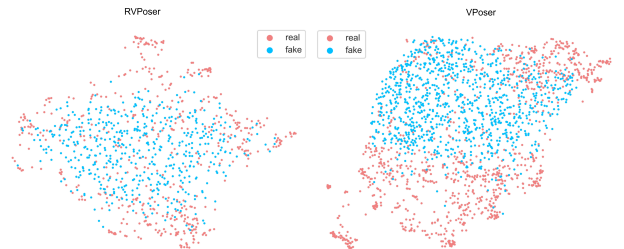


Figure 4: UMAP projections [21] of “real” ground truth samples and of “fake” ones generated by our RVPoser (left) and the VPoser [23] (right) models, respectively.

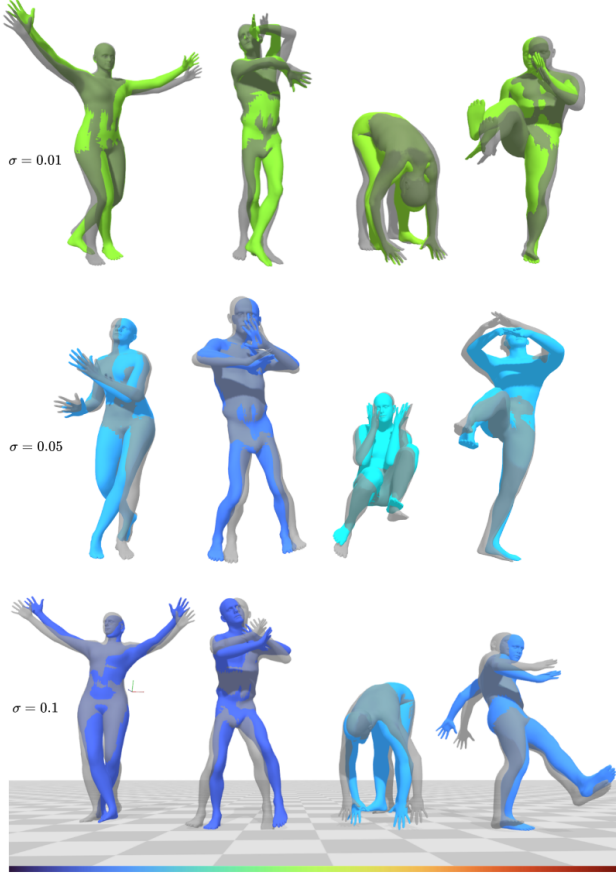


Figure 5: Color-coded (turbo colormap [22] at the bottom) autoencoding ρ of various poses and σ values, using the Sigmoid-based relevance function.

functions, omitting linear weighting as our goal is to boost the contribution of the poses with higher reconstruction error non-linearly. First, we experimented with the Sigmoid function, focusing on the part that corresponds to the positive input values:

$$\rho(\theta) = 1 + 2\left(\frac{e^x}{e^x + 1} - 0.5\right), \quad x = \frac{\epsilon}{\sigma}, \quad (16)$$

where ϵ is the normalized-RMSE, σ is a scaling factor, and θ is the given pose parameters as defined in Eq. (2) of the main paper. As shown in Fig. 5, the Sigmoid-based ρ - although non-linear - leads to similar error values (colorized) and thus fails to serve our cause in significantly boosting the contribution of the least faithfully reconstructed samples. To achieve this, we experiment with a relevance function that scales the error contribution exponentially:

$$\rho(\theta) = e^{\epsilon/\sigma}. \quad (17)$$

Note that since the exponential function does not have an upper limit, we clamp the result at $\rho(\theta) = 3$, so the effective

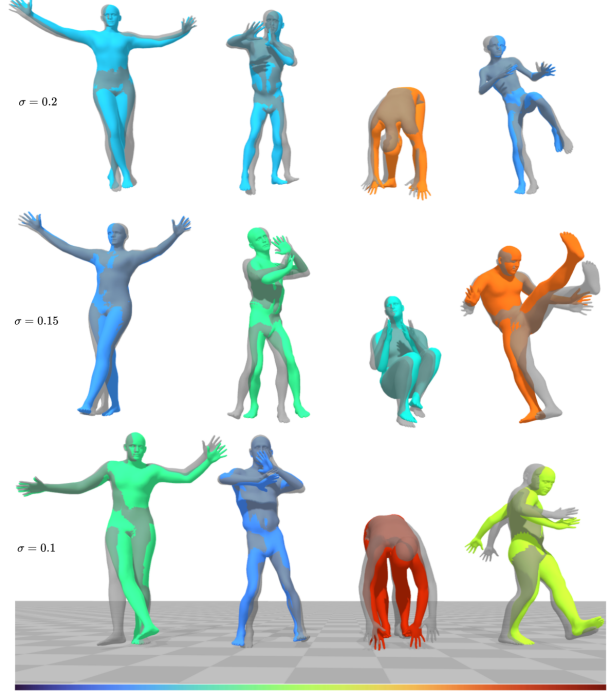


Figure 6: Color-coded (turbo colormap [22] at the bottom) autoencoding ρ of various poses and σ values, using the Exponential-based relevance function.

range of the weighting function is $[1, 3]$, while for the Sigmoid-based relevance function $\rho \in [1, 2]$ range. From the exemplar samples depicted in Fig. 6, it can be observed that the exponential relevance function achieves our original goal as it seems to assign a significantly larger weight to higher reconstruction error (colorized). Note that the performance of each relevance function for different σ values is also depicted in Figs. 5 and 6.

7.3. Orthogonality Investigation

Tab. 2 of the main paper presents the performance of our model against the baseline model (no oversampling or relevance function used) and the same model trained with the Balanced Mean Square Error (BMSE) from [25]. Here, we present further details that help us explore the orthogonality of 2 of the contributions of our paper, namely the oversampling and re-weighting through reconstructability methods, as well as the performance of our best model when trained using the BMSE regression loss.

As shown in Tab. 4, the ‘Ours’ model performs better than the ‘Sampling’ (*i.e.* oversampling synthetic data) and ‘Relevance’ (*i.e.* re-weighting the loss) models for both THuman 2.0 and “tail” test sets. This indicates that there is an underlying synergy between oversampling and re-weighting that is horizontal for simple, challenging, and rare poses. We also observe that both variants improve the

	RMSE ↓	PCK1 ↑	PCK3 ↑	PCK7 ↑
Base	21.4 mm	28.69%	92.08%	98.60%
Sampling	20.4 mm	29.69%	92.78%	98.80%
Relevance	20.6 mm	30.99%	92.79%	98.61%
Ours	19.1 mm	32.38%	93.55%	99.11%
[25]	22.2 mm	25.51%	91.90%	98.62%
Base	35.8 mm	22.04%	80.27%	94.31%
Sampling	31.0 mm	26.34%	83.90%	95.76%
Relevance	33.9 mm	23.61%	81.00%	95.21%
Ours	29.3 mm	23.42%	84.70%	97.24%
[25]	32.9 mm	27.66%	81.98%	94.92%

Table 4: Imbalanced regression ablation. ‘Sampling’ and ‘Relevance’ variants are combined in ‘Ours’ model, while the results of [25] are presented for reference.

	RMSE ↓	PCK1 ↑	PCK3 ↑	PCK7 ↑
Base	21.4 mm	28.69%	92.08%	98.60%
TH2 Random	21.5 mm	31.60%	92.49%	98.60%
TH2 LERP	21.6 mm	29.48%	92.68%	98.58%
TH2 SLERP	20.4 mm	29.69%	92.78%	98.80%
Base	35.8 mm	22.04%	80.27%	94.31%
Tail Random	35.8 mm	23.00%	81.81%	95.70%
Tail LERP	33.5 mm	25.02%	79.82%	95.22%
Tail SLERP	31.0 mm	26.34%	83.90%	95.76%

Table 5: Alternative sampling methods ablation. ‘SLERP’ variant corresponds to the ‘Sampling’ variant in Tab. 4, while ‘Base’ corresponds to the baseline model (*i.e.* no synthetic samples).

baseline, while the oversampling variant seems to perform slightly better than the re-weighting one. This result is in line with the feedback from the prior work in unbalanced regression. For the rest of the orthogonality experiments, we choose the ‘Ours’ model as our best-performing one. Obviously, we have just scratched the surface of the general picture of balancing a regression task and we will keep investigating the complex relationships between different methods that attempt to “unskew” unbalanced distributions.

7.4. Sampling Ablation

Our ‘Sampling’ and ‘Ours’ models consist of a specific strategy for sampling from a learned latent space in order to generate diverse, rare, and plausible poses. As stated in Section 3.1 of the main paper, this strategy is based on non-linear sampling between 2 or more anchor samples. That is, we choose samples using statistical thresholding and use them as anchor samples, avoiding using them in any training or test set. Our sampling strategy is to randomly sample a latent vector and add it to one of the anchor vectors. This

	RMSE ↓	JPE ↓	PCK1 ↑	PCK3 ↑	PCK7 ↑
[8]	18.20 mm	14.80 mm	37.19%	85.38%	99.37%
[7]	22.27 mm	17.08 mm	49.86%	88.98%	97.26%
Ours	17.90 mm	14.20 mm	48.93%	92.55%	98.84%

Table 6: Direct joint solving on CMU [5] test set with a different seed (SEED200 from [8]) than in the main paper.

helps us achieve extra diversity versus (re)using the anchor vector as is. The next step is to pick a latent sample from the intermediate space between 2 anchor neighborhoods. For this purpose, we choose geometric spherical linear interpolation (SLERP) with alternative blending factors in the $[0, 1]$ range and compare it with its linear variant ‘LERP’ and the simple random (*i.e.* no anchors used) sampling (‘Random’).

Tab. 5 presents the performance of our ‘Sampling’ model using each of the 3 different sampling methods on the THuman 2.0 and custom tail test sets, as well as the performance of the ‘Baseline’ for reference. From the results, we can verify that the geometric SLERP helps allows for a safer traversing of the hypersphere-shaped manifold avoiding the dead regions between anchors. This conclusion is supported especially by the performance of SLERP on the “Tail” set, where the sampling neighborhood can be truly “away” from the mean of the manifold. Another interesting feedback from the presented results is the performance drop of the ‘Random’ variant when tested on the tail set compared with the results for THuman 2.0. This result demonstrates the difference between having to operate on diverse - but possibly still close to the mean - poses and having to estimate rare and complex poses. A visual representation of the 3 sampling methods is depicted in Figure 4 of the main paper.

8. Extra Solving Experiments

In the following Tab. 6 we compare the performance of our model to a dataset generated with a different seed following [8] (denoted as SEED200). We observe that the results do not significant vary from those presented in the main paper.

9. Landmarks and fitting ablation

As demonstrated, our noise-aware fitting method is more robust to various types of noise, whether originating from the data, n_d , the model’s inference, n_m , or both. The results in Tab. 7 show that our approach maintains its performance across different noise sources, while the method proposed in [3] may require hyperparameters tuning.

In addition, we present results that are optimized using both ℓ^m and ℓ^j , which further improves performance. Our method also has the advantage of adapting the influence of markers and joints on the fit dynamically, which reduces

	n_d	n_m	RMSE ↓	MAE ↓	PCK1 ↑	PCK3 ↑	PCK7 ↑
[18, 20]	✓	✗	30.10 mm	3.49°	11.79%	66.85%	98.34%
[3]			30.80 mm	3.10°	12.71%	67.06%	97.71%
Ours (ℓ^m)			28.90 mm	2.98°	14.71%	69.86%	98.18%
Ours ($\ell^m \ell^j$)			23.40 mm	2.29°	19.66%	81.06%	99.11%
[18, 20]	✗	✓	20.60 mm	1.93°	28.71%	89.03%	99.05%
[3]			21.71 mm	1.91°	36.38%	87.75%	98.22%
Ours (ℓ^m)			18.70 mm	1.85°	41.99%	90.95%	98.81%
Ours ($\ell^m \ell^j$)			18.50 mm	1.49°	42.18%	91.44%	98.56%
[18, 20]	✓	✓	23.80 mm	2.03°	24.26%	85.63%	98.22%
[3]			24.87 mm	1.94°	31.99%	84.05%	97.00%
Ours (ℓ^m)			22.40 mm	1.79°	36.01%	87.14%	97.53%
Ours ($\ell^m \ell^j$)			21.90 mm	1.52°	36.67%	88.09%	97.69%

Table 7: Noisy landmark fitting on THuman 2.0.

the burden of hyperparameter tuning. In Fig. 9, we qualitatively compare the performance of our method with that of [11], colorised each mesh based on its distance error from the ground truth. Finally, for a fair comparison with [3] we conducted several experiments to find the best range of α values, as well as their initial values. Fig. 7 reports the values of $rmse3$ with different values of α . Interestingly, we found that the best results are obtained with an α range of $[-7, 4]$ and an initial α value of -4.5 .

10. Additional Qualitative Results

We present additional qualitative results comparing our direct regression approach to labeling [11] in the THuman 2.0 and “Tail” sets. These additional results further reinforce the case that a labeling method’s errors are more detrimental to fitting performance, even in cases with no noise, as is evident in the Fig. 8. Finally, Fig. 10 presents qualitative results using real-world data acquired from the developed system presented in Sec. 11, including both model predictions and post-fitting body results, showcasing the benefits of the noise-aware fitting process.

11. System Details

We develop a multi-sensor acquisition system, equipped with 3 Microsoft Kinect for Azure depth sensors, to demonstrate our model’s results in real-time. The system connects K hardware synchronized time-of-flight (ToF) sensors k , $k \in \{1, \dots, K\}$, spatially aligns them by performing extrinsic parameter calibration, and fuses the marker measurements in real-time, producing an unstructured point cloud $\mathbf{m} \in \mathbb{R}^{M \times 3}$, with M being the number of marker estimates.

This process crucially relies on first acquiring 3D position marker measurements from a ToF sensor. The sensor k produces a stream of an infrared image $\mathbf{I}(\mathbf{p}) \in \mathbb{R}$ as well as a pixel-registered depth map $\mathbf{D}(\mathbf{p}) \in \mathbb{R}$, where each pixel $\mathbf{p} \in \mathbb{N}^2$ is defined in the image domain $\Omega := W \times H$ of width W and height H (the subscript k is omitted for the sake of notational simplicity). Using the factory calibrated intrinsic parameters of the sensor, the depth map is straightforwardly transformed to a structured point cloud $\mathbf{P} \in \mathbb{R}^3$,

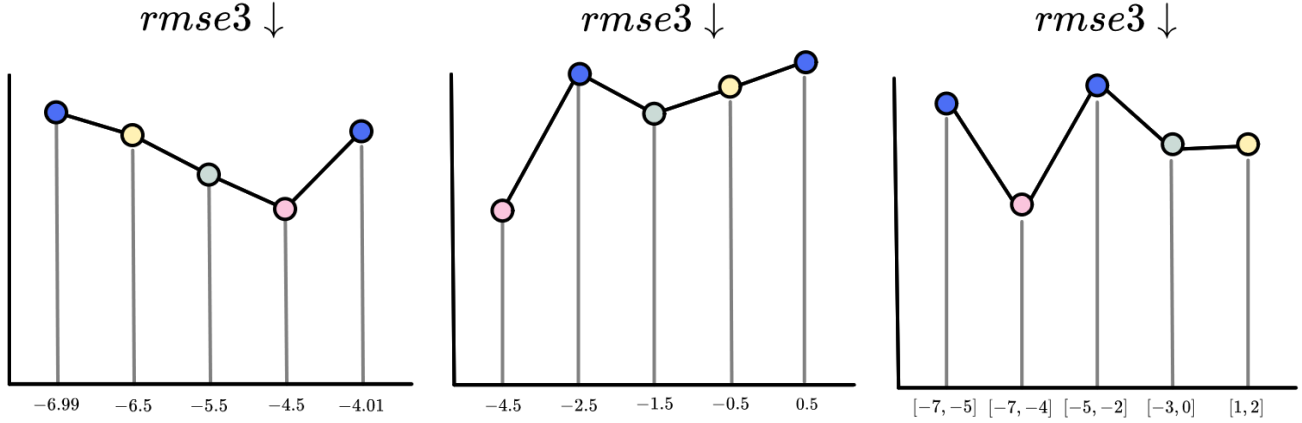
with $\mathbf{P}(\mathbf{p}) = \mathbf{K}\mathbf{G}(\mathbf{p})\mathbf{D}(\mathbf{p})$, with $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ being the intrinsic camera parameters matrix, and $\mathbf{G} \in \mathbb{N}^3$ the homogeneous coordinates image grid.

We exploit this one-to-one mapping between the infrared image \mathbf{I} and the structured point cloud \mathbf{P} to extract the marker positions \mathbf{m}_k . Relying on the retro-reflective properties of markers that return the light emitted by the ToF projector, we identify the marker pixels after applying binary thresholding and contour detection [28] on the infrared image. While measurements are undefined on the actual marker position due to the ToF depth estimation principles, we observe that the measurements around the actual marker position are well-defined. Thus, for each contour we sample the structured point cloud to extract a point measurement, aggregating them into a vector $\mathbf{v} \in \mathbb{R}^{V \times 3}$, with V being the number of the contour points. As spurious outliers can be included in this vector due to fore/background issues and imperfect pixel sampling, we perform Median Absolute Deviation (MAD) outlier rejection [17] using the z -coordinate (depth) of each point, and the average the remaining points to extract the final marker position estimates \mathbf{m}_k .

Using \mathbf{m}_k , the system calibrates the sensors by running bundle adjustment using a simple calibration wand with a marker attached to a stick. Then, gravity alignment is achieved by placing 3 markers in a Γ shape on the floor and extracting the long and short edge cross product as the up vector, transforming all extrinsic transforms to align with it. With the sensors spatially aligned, all marker estimates are fused in a single unstructured point cloud \mathbf{m} . To account for slight calibration errors, we perform point cloud clustering with a radius of 1cm, which results in the actual model input. Evidently, this process is a cascade of numerous estimation errors, the inherent measurement noise that influences the calibration process, and the clustering itself which also adjusts the final estimates. Additionally, we only use $K = 3$ sensors, which accentuates the problem since information fusion is not that effective with such a sparse number of viewpoints.

References

- [1] Advanced Computing Center for the Arts and Design. AC-CAD MoCap Dataset. 3
- [2] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, 2015. 3
- [3] Jonathan T Barron. A general and adaptive robust loss function. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4331–4339, 2019. 8, 9
- [4] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3



(a) With $\alpha_{range} \in [-7, -4]$, we search for the best α_{init} value. (b) With $\alpha_{range} \in [-7, 2]$, we search for the best α_{init} value. (c) We initialize α to the mean value of α_{range} , and search for its best range.

Figure 7: Ablation on α values.



Figure 8: Fits to our regressed versus SOMA labeled markers. The fitting process is more sensitive to labeling errors.

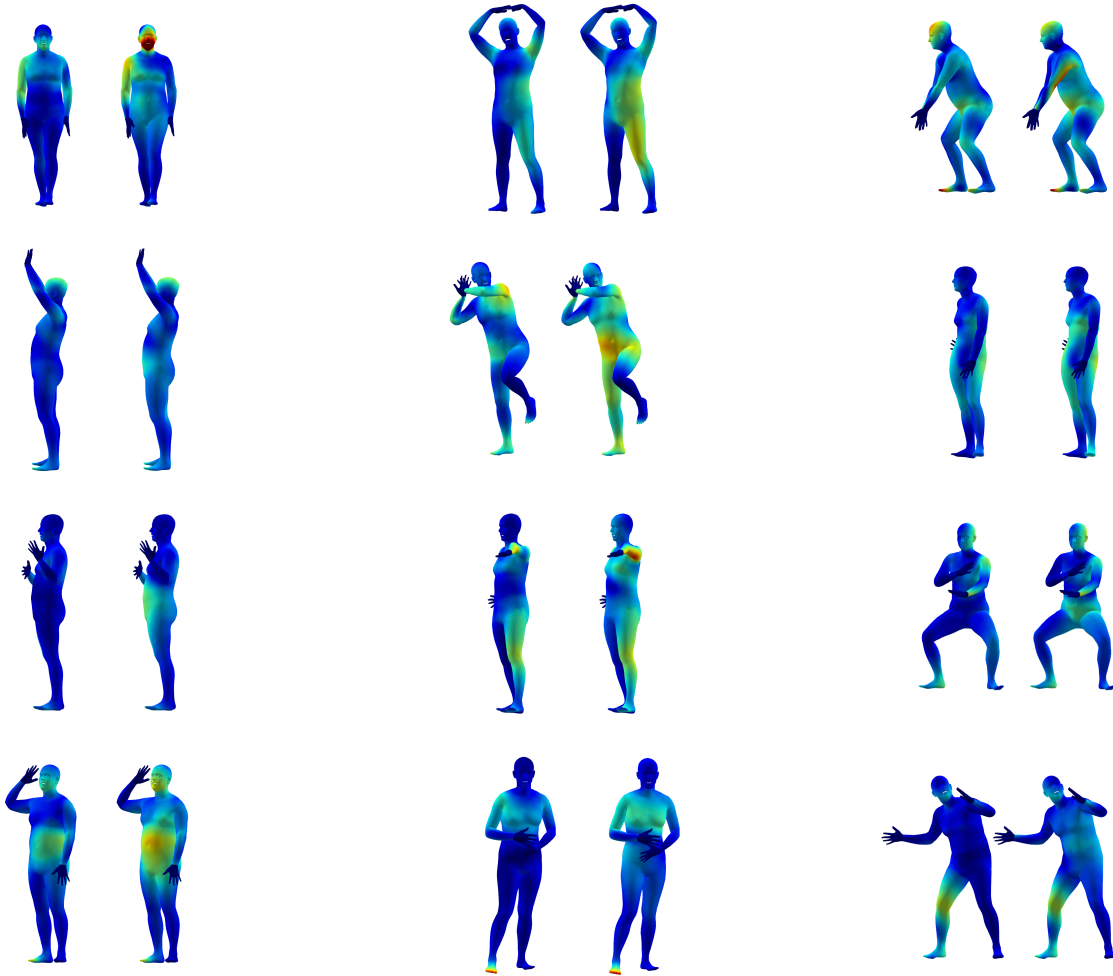


Figure 9: The figure shows the qualitative results of our noise-aware fitting method on the left and the method proposed in [18] on the right. Each mesh is colored using a Jet color map based on the Euclidean distance error metric from the ground truth mesh.

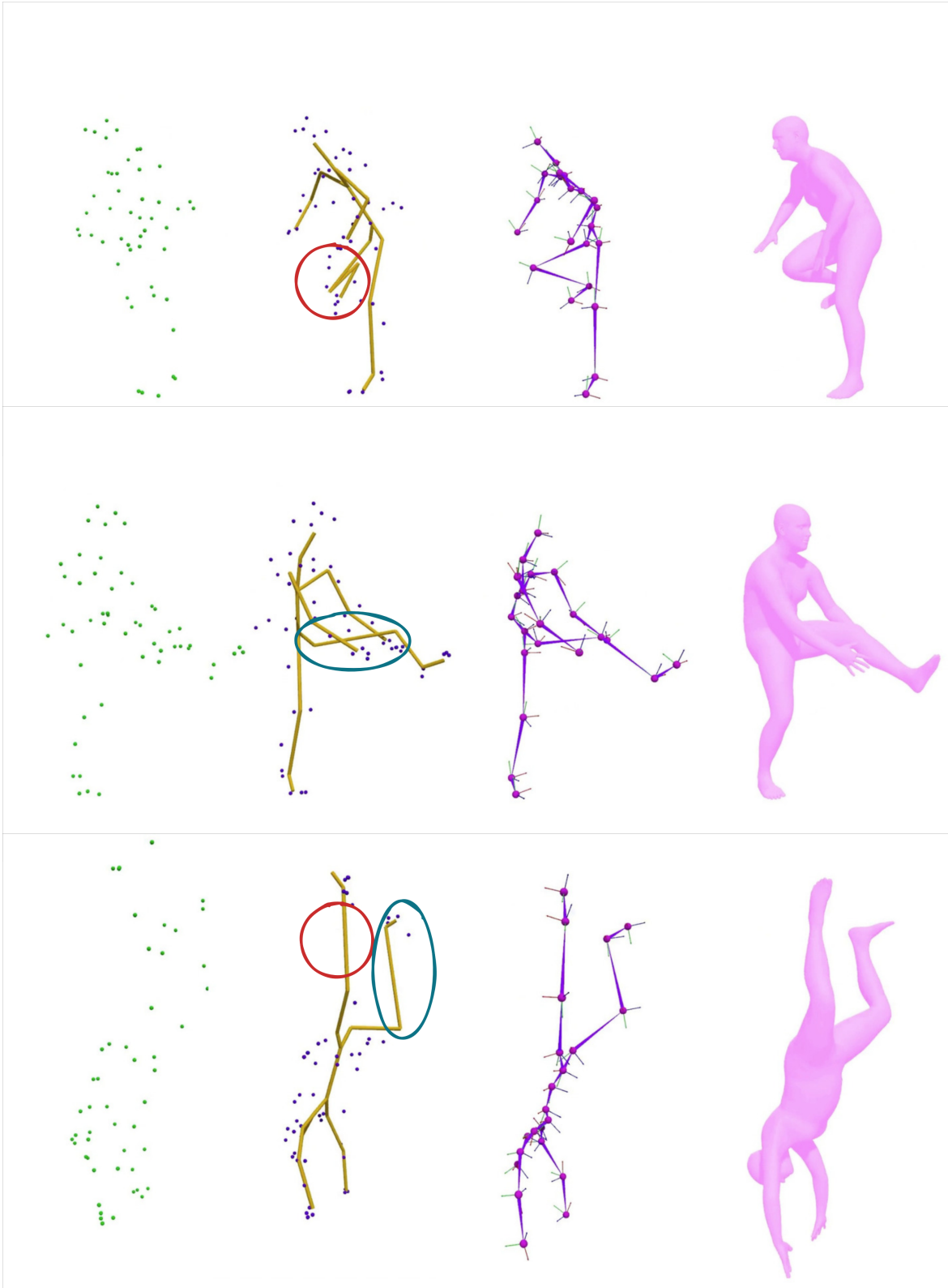


Figure 10: Additional qualitative results of our system in the wild using a setup comprising a very sparse set of low-cost sensors. Starting from the left, we present the raw input collected from our multi-sensor acquisition system (Sec. 11), with the raw (unfiltered) estimated ℓ_{est} from our model following. The last 2 columns present the fitted θ_{est} pose and shape β_{est} parameters. As our real-time model only implicitly learns the human skeleton, this can lead to **unrealistic results**. To address this, the noise-aware fitting approach introduces human body constraints, resulting in more accurate and realistic results. Furthermore, it adequately handles **missing** or **incorrectly** inferred landmarks.

- [5] Carnegie Mellon University. CMU MoCap Dataset. 8
- [6] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 168–172, 1994. 6
- [7] Anargyros Chatzitofis, Dimitrios Zarpalas, Petros Daras, and Stefanos Kollias. Democap: low-cost marker-based motion capture. *International Journal of Computer Vision (IJCV)*, 129(12):3338–3366, 2021. 8
- [8] Kang Chen, Yupan Wang, Song-Hai Zhang, Sen-Zhe Xu, Weidong Zhang, and Shi-Min Hu. Mocap-solver: A neural solver for optical motion capture data. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 2, 8
- [9] Wei Cheng, Su Xu, Jintan Piao, Chen Qian, Wayne Wu, Kwan-Yee Lin, and Hongsheng Li. Generalizable neural performer: Learning robust radiance fields for human novel view synthesis. *arXiv preprint arXiv:2204.11798*, 2022. 3
- [10] Andrey Davydov, Anastasia Remizova, Victor Constantin, Sina Honari, Mathieu Salzmann, and Pascal Fua. Adversarial parametric pose prior. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10987–10995, 2022. 5
- [11] Nima Ghorbani and Michael J Black. Soma: Solving optical marker-based mocap automatically. In *Proc. IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 11117–11126, 2021. 2, 3, 9
- [12] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2Motion: Conditioned generation of 3D human motions. In *Proc. ACM International Conference on Multimedia (MM)*, page 2021–2029, 2020. 4
- [13] Daniel Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 2
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2, 5
- [15] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 8320–8329, 2018. 3
- [16] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2015. 5
- [17] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, 49(4):764–766, 2013. 9
- [18] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 9, 11
- [19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations, (ICLR)*, 2019. 6
- [20] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proc. IEEE/CVF international conference on computer vision (CVPR)*, pages 5442–5451, 2019. 3, 9
- [21] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6
- [22] Anton Mikhailov. Turbo, An Improved Rainbow Colormap for Visualization. <https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html>, 2019. 7
- [23] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 5, 6
- [24] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9054–9063, 2021. 3
- [25] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced mse for imbalanced visual regression. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7926–7935, 2022. 7, 8
- [26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 2
- [27] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4, 2010. 3
- [28] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985. 9
- [29] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *Proc. European Conference on Computer Vision (ECCV)*, pages 572–589. Springer, 2022. 5
- [30] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5746–5756, 2021. 3
- [31] Richard Zhang. Making convolutional networks shift-invariant again. In *Proc. International Conference on Machine Learning (ICML)*, 2019. 2
- [32] Yan Zhang, Michael Black, and Siyu Tang. Perpetual motion: Generating unbounded human motion. *arXiv:2007.13886*, 2020. 6