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

Georgios Albanis [1,2]   Nikolaos Zioulis [1]   Spyridon Thermos [1]
Anargyros Chatzitofis [1]   Kostas Kolomvatsos [2]

[1]Moverse {giorgos,nick,spiros,argyris}@moverse.ai
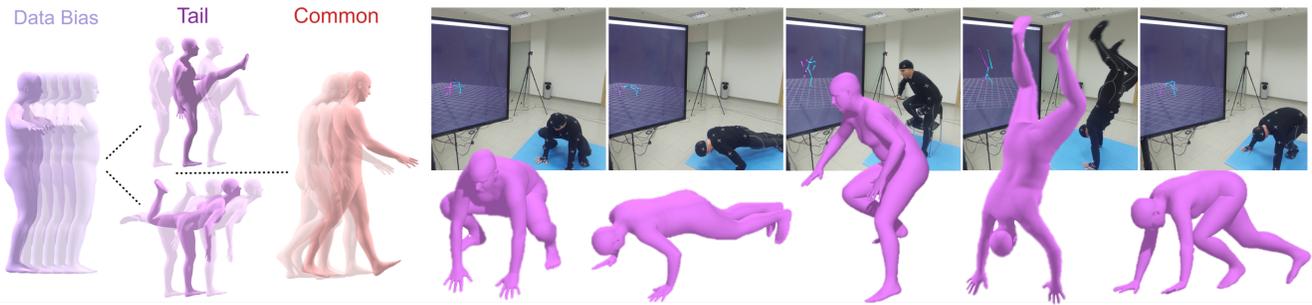[2]Dept. of Informatics and Telecommunications, University of Thessaly kostasks@uth.gr

Figure 1: Effective use of human motion data needs to overcome their inherent temporal bias and long-tailed distribution (*left*). Our model uses a novel balanced regression technique to improve robustness and accuracy to challenging poses, denoise markers and solve joints using raw unstructured marker positions as input. It runs in real-time and can handle higher noise levels (*right*), producing high-quality body fits even when deployed in a system using just 3 consumer-grade sensors.

Abstract

Real-time optical Motion Capture (MoCap) systems have not benefited from the advances in modern data-driven modeling. In this work we apply machine learning to solve noisy unstructured marker estimates in real-time and deliver robust marker-based MoCap even when using sparse affordable sensors. To achieve this we focus on a number of challenges related to model training, namely the sourcing of training data and their long-tailed distribution. Leveraging representation learning we design a technique for imbalanced regression that requires no additional data or labels and improves the performance of our model in rare and challenging poses. By relying on a unified representation, we show that training such a model is not bound to high-end MoCap training data acquisition, and exploit the advances in marker-less MoCap to acquire the necessary data. Finally, we take a step towards richer and affordable MoCap by adapting a body model-based inverse kinematics solution to account for measurement and inference uncertainty, further improving performance and robustness. Project page: moverseai.github.io/noise-tail.

1. Introduction

Human Motion Capture (MoCap) technology has benefited from the last decade's data-driven breakthroughs mostly due to significant research on the human-centric visual understanding that focuses on unencumbered capture using raw color inputs. The golden standard of MoCap technology – referred to as "optical" – still uses markers attached to the body, often through suits, for robust and accurate captures, and has received little attention in the literature. These scarce works [25, 21, 20, 14, 29, 13] mainly focus on processing (raw) archival MoCap data for direct marker labeling [21, 20] or labeling through regression [25], solving the skeleton's joints [14, 13] or transforms [29], while [13] also addressing the case of commodity sensor captures and the noise levels associated with it.

As even high-end systems produce output with varying noise levels, be it either information- (swaps, occlusions, and ghosting), or measurement-related (jitter, positional shifts), these works exploit the plain nature of raw marker representation to add synthetic noise during training. Still, for data-driven systems, the variability of marker placements comprises another challenge that needs to be addressed. Some works [13, 29] address this implicitly,

relying on the learning process, while others [14] address this quasi-explicitly, considering them as input to the model. Another way to overcome this involves fitting the raw data to a parametric model after manually [25, 44, 49], or automatically [21, 20] labeling and/or annotating correspondences, standardizing the underlying representation.

In this work, we explore the next logical step stemming from prior work, bridging standardized representations and consumer-grade sensing, and delivering real-time data-driven MoCap that is robust to tracking errors. Most works [20, 14, 21, 29, 13] leverage high-end MoCap to acquire training data, a process that is expensive, laborious and difficult to scale, apart from [25] that used low-cost sensor acquired data, but nonetheless, applied the model to a high-end capturing system.

Instead, by relying on a standardized representation using a parametric human body model, we benefit from modern markerless capture technology, greatly increasing data acquisition rates at a fraction of the costs and labor. Still, there are certain challenges that need to be addressed, such as the distribution of MoCap data and the input optical sensing noise.

The nature of human motion, albeit high-dimensional, instills a significant level of data redundancy in MoCap datasets. Indeed, standing still or walking poses dominate most captures and affect the training data distribution in two ways. First by introducing bias in the learning process, and second, by further skewing the long-tailed distribution. The latter is an important problem [67] that data-driven methods need to overcome as rare poses exist, not only due to their reduced appearance frequency, but also due to biomechanical limitations of the captured subjects in fast movements, body balancing, and striking challenging poses. Prior work crucially neglects this, resorting to uniform temporal downsampling, which only helps in reducing data samples, yet not redundancy nor long-tailed distribution.

Another typical assumption is that the raw marker data are relatively high quality, most common to labeling works [20, 21] that solve using the raw positions. Even though synthetic noise is added during training, this is mostly to regularize training as the noisy nature of inputs is not taken into account post-labeling. Those works that directly infer solved estimates [13, 14, 29] solely rely on the model's capacity to simultaneously denoise the inputs and solve for the joints' positions. Nonetheless, even the models' outputs are uncertain, a situation that will be increasingly magnified when the raw marker input is affected by higher noise levels, as common when relying on consumer-grade sensors. This lack of solutions that increase noise robustness hinders the adoption of more accessible sensing options.

To that end, we present techniques to address MoCap dataset challenges as well as noisy inputs, resulting in a MoCap framework that ● <u>does not</u> necessarily require data from high-end MoCap systems, ● <u>does not</u> require additional data to boost long-tail performance, and ● <u>does not</u> require specialised hardware. More specifically we:

➔ Leverage representation learning to jointly oversample and perform utility-based regression, addressing the redundancy and long-tailed MoCap data distribution.

➔ Introduce a noise-aware body shape and pose solver that models the measurement uncertainty region during optimization.

➔ Demonstrate a real-time inference capable and artifact-free MoCap solving model, running at $60Hz$ on a system comprising just 3 consumer-grade sensors.

➔ Harness a human parametric representation to cold-start data-driven optical MoCap models using data through markerless acquisition methods.

2. Related Work

2.1. MoCap Solving

Solving the joints' positions or transforms from marker data is a cascade of numerous (sometimes optional) steps. The markers need to be labeled, ghost markers need to be removed, occluded markers should be predicted and then an articulated body structure needs to be fit to the observed marker data. Various works address errors at different stages of MoCap solving, with contemporary ones relying on smoothness and bone-related (angles, offsets and lengths) constraints [27, 66, 31, 6, 18, 53, 73]. Recent approaches started resorting to existing data for initialization [69] or marker cleaning [5]. MoSh [44] moved one step ahead and instead of relying on plain structures employed a parametric human body to solve labeled marker data and estimate pose articulation and joint positions, even accounting for marker layout inconsistencies and/or soft tissue motion.

Nonetheless the advent of modern – deep – data-driven technologies have stimulated new approaches for MoCap solving. A label-via-regression approach was employed in [25] where a deep model was used to regress marker positions and then perform maximum assignment matching for labeling the input. Labeling was also formulated as permutation learning problem [21], albeit with constraints on the input, which were then relaxed in [20] by adding a ghost category. However, labeling assumes that the raw data are of a certain quality as the raw measurements are then used to solve for the joints' transforms or extra processing steps are required to denoise the input.

Consequently, end-to-end data-driven approaches that can simultaneously denoise and solve have been a parallel line of research. While end-to-end cleaning and solving is possible using solely a single feed-forward network [29], the process naturally benefits from using two cascaded

autoencoders [62], the first operating on marker data and cleaning them for the subsequent joint regressor. The staging from markers to joints was also shown to be important from a performance perspective in [13] which trained a convolutional network with coupled noisy and clean data captures to address noisy inputs. Recently, graph convolutional models were employed in [14] allowing for the explicit encoding of marker layout and skeleton hierarchy, two crucial factors of variation that were only implicitly handled in prior end-to-end solvers.

2.2. MoCap Data

Learning to solve MoCap marker data requires supervision provided by collecting data using professional high-end MoCap systems [29, 20, 14, 13]. SOMA [20] standardized the representation using the AMASS dataset [49] which, in turn, relied on an extension of MoSh [44] to fit a parametric human body model to markers. All other works suffer from inconsistent marker layouts which is a problem that was either implicitly addressed [29, 13] or quasi-explicitly [14] using the layouts as inputs. Marker data can be (re-)synthesized in different layouts when higher-level information is available (*e.g.* marker-to-joint offsets, meshes) [29, 20]. Yet, it has been also shown that fitting a synthetic hand model to depth data acquired by consumer-grade sensors can also produce usable training data [25] for deploying a model to a high-end marker capturing system for data-driven MoCap. Compared to [25], we experimentally demonstrate this feasibility and even extend it to noisy inputs at run-time, something not considered in [25] as it relied on a high-end system for live capture.

Statistical parametric models [45, 61, 58, 59, 85, 87, 4, 88] are more expressive alternatives than the skinned mesh [83] used in [25] as, apart from realistic shape variations, deformation corrective factors can also be employed. They have been used to synthesize standardized training data before [82, 28, 38] but crucially rely on preceeding high-end MoCap acquisition. We also explore this path using multi-view markerless capture [33, 15, 92] to produce parametric model fits and synthesize marker positions as a solution to the cold-start problem of data-driven MoCap solving. Even though such data can be fit to marker data as done in AMASS [49] and Fit3D [19], the potential of acquiring them using less expensive capture solutions is very important, as long as it is feasible to train high quality models.

Still, one also needs to take into account the nature of human performance data and their collection processes. As seen in AMASS [49] and Fit3D [19], both contain significant redundancies and suffer from the long-tail distribution effect. Rare poses are challenging for regression models to predict, mainly stemming from the combined effect of the selected estimators and stochastic optimization with mini-batches. Various solutions have been surfacing in the litera-

ture, some tailored to the nature of the problem [67], leveraging a prototype classifier branch to initialize the learned iterative refinement, and others adapting works from imbalanced classification to the regression domain. Traditional approaches fall into either the re-sampling or re-weighting category, with the former focusing on balancing the frequency of samples and the latter on properly adjusting the parameter optimization process. Re-sampling strategies involve common sample under-sampling [79], rare sample over-sampling by synthesizing new samples via interpolation [81], re-sampling after perturbing with noise [9], and hybrid approaches that simultaneously under- and over-sample [10]. Yet interpolating high-dimensional samples like human pose is non-trivial or even defining the rare samples that need to be re-sampled.

Utility-based [80] – or otherwise, cost-sensitive – regression assigns different weights – or relevance – to different samples. Defining a utility function is also essential to re-sampling strategies for regression [79]. Recent approaches employ kernel density estimation [74], adapt evaluation metrics as losses [72], or resort to label/feature smoothing and binning [89]. Another family of methods that are now explored can be categorized as contrastive, with [22] regularizing training to enforce feature and output space proximity. BalancedMSE [64] is also a contrastive-like objective that employs intra-batch minimum error sample classification using a cross-entropy term that corresponds to an L2 error from a likelihood perspective. However, most approaches rely on stratified binning of the output space using distance measures that lose significance in higher dimensions. Further, binning can only be used with specific networks/architectures (proper feature representations for classifying bins or feature-based losses). It has not been shown to be applicable in high-performing dense networks relying on heatmap representations. Instead, we introduce a novel technique that can jointly over-sample and assign higher relevance to rare samples by leveraging representation learning and its synthesis and auto-encoding traits.

3. Approach

The MoCap representation we use is a parametric human body model $\mathcal{B}$. Different variants exist, all data-driven, some relying on stochastic representations [87], others on explicit ones [45, 58], with a notable exception using an artist-made one [88] and all typically employ linear blend skinning [34] and pose corrective factors [45, 87] to overcome its artifacts. Generally, we consider it as a function $(\mathbf{v}, \mathbf{f}) = \mathcal{B}(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{T})$, where $(\mathbf{v}, \mathbf{f})$ are the vertices $\mathbf{v} \in \mathbb{R}^{V \times 3}$ and faces $\mathbf{f} \in \mathbb{N}^{F \times 3}$ of a triangular mesh surface that is defined by S blendshape coefficients $\boldsymbol{\beta} \in \mathbb{R}^S$, articulated by P pose parameters $\boldsymbol{\theta} \in \mathbb{SO}(3)^P$, and globally positioned by the transform $\mathbf{T} \in \mathbb{SE}(3)$. Using linear functions r expressed as matrices $\mathbf{R}$ it is possible to extract L differ-
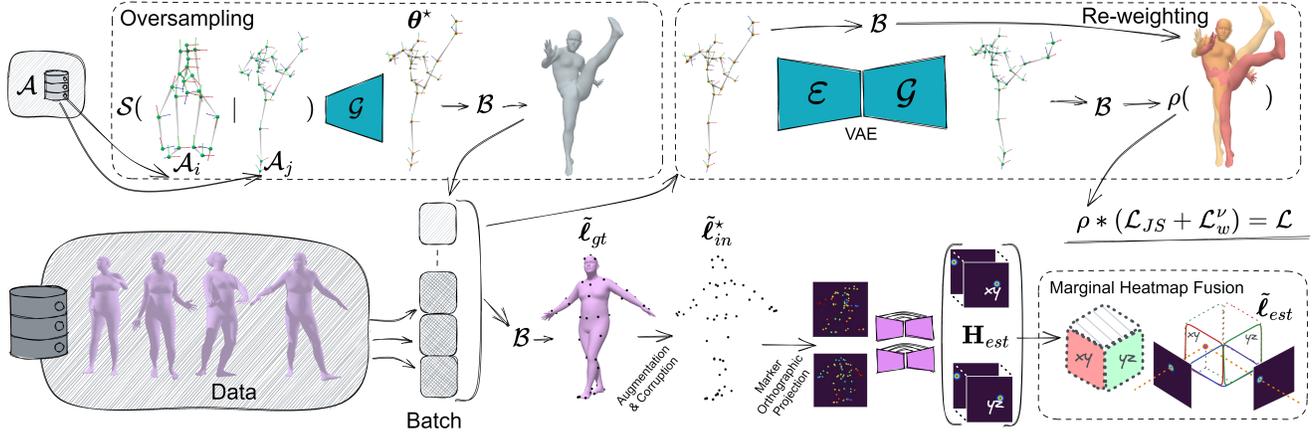
Figure 2: Overview of the balanced and real-time MoCap solving training model. Starting from an existing data corpus (*bottom left*), a set of encoded tail anchor poses $\mathcal{A}$ are selected (Sec. 3.1 - *top left*) and randomly blended via $\mathcal{S}$ and a generator $\mathcal{G}$. This oversamples the tail, adding extra synthetic rare samples during training. A UNet model (Sec. 3.2 - bottom middle) receives two orthographic depth map renders (xy and yz planes) of augmented and corrupted marker 3D positions $\ell_{in}^\star$ extracted from the body's $\mathcal{B}$ surface, producing 2 orthogonal heatmaps which are marginally fused along the y coordinate, producing 3D positions $\tilde{\ell}_{est}$ (Sec. 3.2 - *bottom right*). The loss for each batch item is re-weighted by its relevance ρ, computed after calculating the joint reconstruction error of its pose's $\boldsymbol{\theta}$ generative autoencoder reconstruction (Sec. 3.1 - *top right*).

ent body landmarks $\ell := r(\mathbf{v}) = \mathbf{R} \times \mathbf{v}$, with $\ell \in \mathbb{R}^{L \times 3}$ and $\mathbf{R} \in \mathbb{R}^{L \times V}$. This way, surface points ℓ^v can be extracted using delta (vertex picking) or barycentric (triangle interpolation) functions and joints ℓ^j using weighted average functions. Since markers are extruded by the marker radius d they correspond to $\ell^m = \ell^v + d(\mathbf{R} \times \mathbf{n})$, with $\mathbf{n}$ being the vertices' normals.

Following prior art [20], the input data are the parameters of a body model that synthesize markers, which due to their synthetic nature can be augmented, and corrupted with artifacts and noise [29, 14, 13]. Fig. 2 illustrates our model's training framework which is followingly explained starting with the technique addressing the redundancy and long-tailed nature of the data (Sec. 3.1), the marker denoising and joint solving model's design choices (Sec. 3.2), and finally the noise-aware body parameter solver (Sec. 3.3).

3.1. Balancing Regression

Relevance functions drive utility regression and guide the re-/over-/inter-sample selection/generation [10, 79, 80, 81]. Instead of defining relevance or sample selection based on an explicit formula or set of rules, we employ representation learning to learn it from the data. Autoencoding synthesis models [41, 65] jointly learn a reconstruction model as well as a generative sampler:

$$\boldsymbol{\theta}^\ddagger = \mathcal{G}(\mathcal{E}(\boldsymbol{\theta})), \qquad \boldsymbol{\theta}^\star = \mathcal{G}(\mathcal{S}(\cdot)), \qquad (1)$$

with varying constraints on the input $\boldsymbol{\theta}$ and latent $\mathbf{z} = \mathcal{E}(\boldsymbol{\theta}), \mathbf{z} \in \mathbb{R}^Z$ spaces. An encoder $\mathcal{E}(\boldsymbol{\theta})$ maps input $\boldsymbol{\theta}$ to

a latent space $\mathbf{z}$ which gets reconstructed to $\boldsymbol{\theta}^\ddagger$ by a generator $\mathcal{G}(\mathbf{z})$. Using a sampling function $\mathcal{S}$ to sample the latent space it is also possible to generate novel output samples $\boldsymbol{\theta}^\star$. We exploit the hybrid nature of such models to design a novel imbalanced regression solution that simultaneously over-samples the distribution at the tail and adjusts the optimization by re-weighting rarer samples. Our solution is based on a deep Variational AutoEncoder (VAE) [41].

Relevance via Reconstructability. Autoencoding models are expected to reflect the bias of their training data, with redundant/rare samples being easier/harder to properly reconstruct respectively. This bias in reconstructability can be used to assign relevance to each sample as those more challenging to reconstruct properly are more likely to be tail samples. We define a relevance function ρ (see Fig. 2 re-weighting) using a reconstruction error ϵ:

$$\rho(\theta) = 1 + exp(\epsilon/\sigma), \quad \epsilon = \sqrt{\frac{1}{J} \sum_{i=1}^{J} ||\bar{\ell}_i^j - \bar{\ell}_i^{j\ddagger}||_2}, \quad (2)$$

with $(\bar{\cdot})$ denoting unit normalization using the input joints' bounding box diagonal, ϵ the normalized-RMSE over the reconstructed and original joints, and σ a scaling factor controlling the relevance ρ. Using landmark positions we can preserve interpretable semantics in ρ and σ as they are unidirectionally interchangeable (linear mapping) with the pose $\boldsymbol{\theta}$ given fixed shape $\boldsymbol{\beta}$. Fig. 3 shows exemplary poses as scored by our relevance function.

Balance via Controlled Synthesis. Even though the tail samples are not reconstructed faithfully, the generative
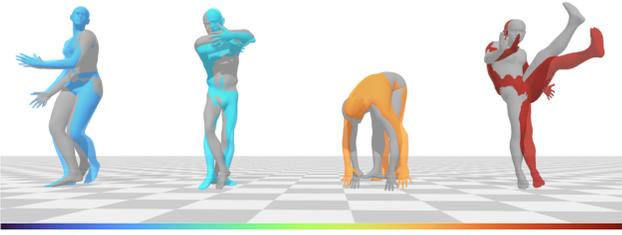
Figure 3: Color-coded (turbo colormap [54] at the bottom) autoencoding relevance ρ of various poses.

Figure 4: Tail oversampling using latent anchors $\mathcal{A}$. Random latent vector blending using **non-linear** interpolation generates diverse and realistic tail samples, compared to the **linear** one which produces less diverse or unrealistic samples, or to **random** sampling which produces more biased samples.

and disentangling nature of modern synthesis models shape manifolds that map inputs to the underlying factors of data variation, effectively mapping similar poses to nearby latent codes which can be traversed across the latent space dimensions. Based on this, we define a controlled sampling scheme for synthesizing new tail samples (see Fig. 2 oversampling). Using the relevance function from Eq. (2), it is possible to identify tail samples $\boldsymbol{\theta}^\dagger$ via statistical thresholding that serve as anchor latent codes $\mathcal{A} = \{\, \mathbf{z}^\dagger \mid \mathbf{z}^\dagger = \mathcal{E}(\boldsymbol{\theta}^\dagger) \,\}$. This process adapts to the training data distribution instead of risking a mismatch via empiric manual picking when using a purely generative model (*e.g.* [78]). We then sample using the following function:

$$\mathcal{S}_{i,j}(\cdot) = \varsigma(\mathcal{N}(\mathbf{a}_i, \mathbf{s}), \mathcal{N}(\mathbf{a}_j, \mathbf{s}), b), \quad \mathbf{a}_{i,j} \in_R \mathcal{A}. \quad (3)$$

Specifically, we sample from a normal distribution centered around two random anchors i and $j, i \neq j$, from $\mathcal{A}$ using a standard deviation $\mathbf{s}$, and blend them using spherical linear interpolation [70] ς with a uniformly sampled blending factor $b \in \mathcal{U}(0, B), B \in [0, 1]$. Non-linear interpolation between samples avoids dead manifold regions as not all directions lead to meaningful samples [35, 37] and increases our samples' plausibility [86], as illustrated in Fig. 4.

3.2. Real-time Landmark Estimation

Compared to pure labeling [20, 21] or pure solving approaches [29, 14] we design our model around simultaneous denoising, solving and hallucination.

While some approaches use the raw marker positions as input, we opt to leverage the maturity of structured heatmap representations and employ a convolutional model, similar to [25, 13] instead of relying on unstructured regression [14, 29] using MLPs. This improves the convergence of the model and by using multi-view fusion we can also improve accuracy via robust regression. First, we augment and corrupt the input markers ℓ_{gt} into $\tilde{\ell}_{in}^\star$. Then, we normalize and render $\tilde{\ell}_{in}^\star$ from two orthographic viewpoints as in [13], but with a notable difference when processing the model's output; instead of predicting the 3rd dimension, we manage to predict normalized 3D coordinates by learning to
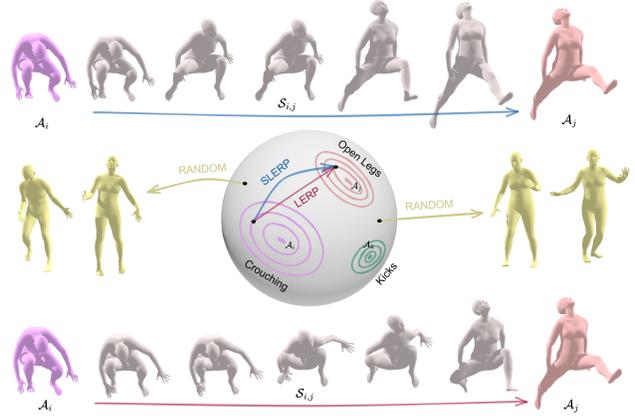
solve a single 2D task. To achieve that, we use the two rendered views as input to the model, predict the corresponding view's heatmaps, and fuse them with a variant of marginal heatmap regression [56, 90] (see Fig. 2 fusion). We assume the gravity direction along the y axis and use the orthogonal and orthographic views denoted as xy and yz which share the y axis. To estimate the landmarks' normalized positions $\tilde{\ell}_{est}$, we employ center-of-mass regression [48, 75, 55, 77] taking the average expectation [56, 90] for y from the two views. The model is supervised by:

$$\mathcal{L} = \rho(\lambda \mathcal{L}_{JS}(\mathbf{H}_{gt}, \mathbf{H}_{est}) + \mathcal{L}_w^\nu(\tilde{\ell}_{gt}, \tilde{\ell}_{est})), \quad (4)$$

where $\mathcal{L}_{JS}$ is the λ−weighted Jensen-Shannon divergence [52] between the normalized ground truth and soft-max normalized predicted heatmaps, while $\mathcal{L}_w^\nu$ is the robust Welsch penalty function [30, 17], with the support parameter ν, between the normalized landmark ground-truth $\tilde{\ell}_{gt}$ and estimated $\tilde{\ell}_{est}$ coordinates. Overall, $\mathcal{L}_{JS}$ accelerates training while $\mathcal{L}_w^\nu$ facilitates higher levels of sub-pixel accuracy since even though we reconstruct the heatmaps $\mathbf{H}$ using the normalized − un-quantized − coordinates [93], discretization artifacts can never be removed entirely.

Note that the fusion outcome $\tilde{\ell}_{est}$ comprises both marker and joint estimations, essentially estimating a complete, labeled, and denoised marker set, as well as solving for the joints' positions.

Finally, we use U-Net [68] as a regression backbone for its runtime performance and its efficiency in high-resolution regression.

3.3. Noise-aware Fitting

Given the denoised and complete set of landmarks $\tilde{\ell}_{est} \in \mathbb{R}^{L \times 3}$, we can fit the body to these estimates and obtain the pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$ which implies an articulated skeleton and mesh surface. This is a non-linear optimization problem with the standard solution being MoSh [44] and its successor MoSh++ [49]. However, MoSh(++) also solves for the marker layout which in our case is known apriori as the model was trained with a standard 53 marker configuration. Compared to prior works that assume the estimates are of high-quality or low signal-to-noise ratios, we seek to relax this assumption to support additional sensing options. The solution to this is robust optimization but typical approaches that involve robust kernels/estimators require confident knowledge about the underlying data distribution. This is not easily available in practice, and moreover, it varies with different sensing options but more importantly, when involving a data-driven model, it is skewed by another challenging-to-model distribution. The Barron loss [7] is a robust variant that also adapts to the underlying distribution and interpolates/generalizes many known variants by adjusting their shape and scale jointly.

Following likelihood-based formulations [39, 24] that have been presented for multi-task/robust stochastic optimization, we formulate a noise-aware fitting objective that is adaptive and optimizes the Gaussian uncertainty region $\boldsymbol{\sigma} \in \mathbb{R}^{L}$ jointly with the data and prior terms:

$$\underset{\boldsymbol{\theta}^*, \boldsymbol{\beta}^*, \mathbf{T}^*, \boldsymbol{\sigma}^*}{\operatorname{argmin}} \quad \mathcal{E}_{data} + \mathcal{E}_{prior}. \tag{5}$$

We use standard prior terms [44, 49, 61] $\mathcal{E}_{prior} = \lambda_{\boldsymbol{\beta}} \sum ||\boldsymbol{\beta}||_2 + \lambda_{\mathbf{z}} \sum ||\mathbf{z}||_2$, and a data term formulated as:

$$\mathcal{E}_{data} = \sum_{i}^{L} \frac{1}{2\sigma_i^2} ||\tilde{\ell}_{est,i} - \tilde{\ell}_i^*||_2 + log\sigma_i. \tag{6}$$

As in MoSh(++) we perform staged annealed optimization but with only 2 stages as there is no marker layout optimization. The first stage optimizes over $\boldsymbol{\beta}^*, \boldsymbol{\theta}^*, \mathbf{T}^*$, while the second stage fixes $\boldsymbol{\beta}$ and $\mathbf{T}$ and optimizes $\boldsymbol{\theta}^*, \boldsymbol{\sigma}^*$.

4. Results

We base our implementation on the SMPL(-X) body model $\mathcal{B}$ [45, 61]. Our models are implemented using PyTorch [60], optimized with Adam [40], initialized with Kaiming init. [26], and trained for a fixed number of epochs and with a fixed seed, with the best parameters selected using the performance indicators presented in Sec. 5 of the supplement. UNet receives 160×160 depth maps and outputs heatmaps of the same resolution for all landmarks (53 markers and 18 joints in all cases apart from the experiments in Tab. 3 where 56 markers and 24 joints are used for consistency). The autoencoding generator is implemented as a

	RMSE↓	PCK1↑	PCK3↑	PCK7↑
Optical#1	50.4 mm	36.14%	**84.89%**	**90.90%**
Optical#2	89.9 mm	**41.11%**	81.18%	86.24%
Optical#3	92.9 mm	39.16%	79.74%	86.08%
Markerless	59.4 mm	21.70%	79.96%	90.08%

Table 1: Markerless vs optical data tested on ACCAD.

robust variant of VPoser [61][1]. To fit the body to the estimated landmarks we use quasi-Newton optimization [57]. For the evaluation, the $\tilde{\ell}_{est}$ are denormalized to ℓ_{est}. Finally, the Tables are color-coded with the best result being visualized in pink and bolded, the second in green, and the third (where it is needed) in yellow.

We use a variety of datasets that provide corresponding parametric body $\mathcal{B}$ parameters from which we can extract input (markers) and ground truth (joints and markers). We additionally curate a custom test set comprising 4 categories of tail samples. Note that all models' performance is validated using *unseen* data comprising entire datasets, thus, ensuring different capturing contexts. For a lack of space, we moved all preprocessing (see supp. Sec. 3), datasets (see supp. Sec. 4), and metrics (see supp. Sec. 5) details in the supplement, as well as an in-the-wild supp. video.

Are high-end MoCap data necessary?

Relying on an intermediate body model $\mathcal{B}$ representation opens up new opportunities for data acquisition. We seek to validate the hypothesis that training an optical MoCap model does not necessarily require data acquired by high-end optical MoCap systems. Recent multi-view datasets [92, 15, 63] rely on markerless capturing technology to fit parametric body models to estimated keypoint observations. We train our model (without the imbalanced regression adaptation) on the combined GeneBody [15] and THuman2.0 (TH2) [92] multi-view marker-less data (*Markerless*), and on 3 high-end MoCap dataset combinations from AMASS [49], specifically, EKUT [50], HumanEva [71], MoSh [44], and SOMA [20] (*Optical #1*); CNRS and HumanEva (*Optical #2*); and, solely HumanEva (*Optical #3*) to progressively reduce the diversity of the samples. We equalize the different markerless and optical training data via temporal downsampling to a total of 9mins of MoCap. By evaluating these models using ACCAD [2] (see Tab. 7), we observe a correlation between pose diversity and performance, and that the markerless data result in comparable performance to the high-end MoCap data. The latter indicates that it is possible to acquire data for optical MoCap without having access to any high-end system.

---

[1]Description and comparison can be found in Sec. 7.1 of the suppl.

		RMSE↓	PCK1↑	PCK3↑	PCK7↑
TH2	Base	21.4 mm	28.69%	92.08%	98.60%
	[64]	22.0 mm	25.51%	91.90%	98.62%
	Ours	**19.1** mm	**32.38%**	**93.55%**	**99.11%**
Tail	Base	35.8 mm	22.04%	80.27%	94.31%
	[64]	32.9 mm	**27.66%**	81.98%	94.92%
	Ours	**29.3** mm	23.42%	**84.70%**	**97.24%**

Table 2: Imbalanced regression results.

	RMSE↓	JPE↓	PCK1↑	PCK3↑	PCK7↑
[14]	21.1 mm	17.4 mm	38.11%	84.70%	**99.17%**
[13]	27.0 mm	17.5 mm	**51.08%**	89.39%	97.24%
Ours	**20.1** mm	**15.9** mm	50.14%	**92.23%**	98.14%

Table 3: Direct joint solving on CMU test set [11].

Figure 5: Fits to our regressed vs SOMA labeled markers. Incorrect labeling results in highly erroneous fits.

Addressing the bias and long-tail

To evaluate our novel imbalanced regression discussed in Sec. 3.1, we design an experiment simulating a progressive data collection process by aggregating the DFaust [8], EYES [47], EKUT, HumanEva, MoSh, PosePrior [3], SFU [91], SOMA, SSM, and Transitions parts from AMASS, captured with varying acquisition protocols and settings. Tab. 2 presents the results compared to a baseline model trained without re-weighting/oversampling, and the BMSE [64] imbalanced regression loss, which is properly adapted to consider joint distances and not scalars.

Tab. 2 (top) presents the results on TH2, a dataset of diverse static poses that also includes challenging poses (*e.g.* extreme bending, inversion, etc.), where our approach improves overall performance compared to BMSE that presents inferior results to the baseline model. Tab. 2 (bottom) presents the results on our "tail" (rare) poses that include "*high kicks*", "*crouching*", "*crossed arms*", and "*crossed legs*". Both imbalanced regression approaches improve the long-tail performance, with our oversampling and re-weighting method achieving the best results almost horizontally. These results highlight that our approach overcomes the known weakness of the BMSE balancing the data distribution at the expense of performance on more common poses. Ablation experiments showcasing the orthogonality of oversampling and re-weighting can be found in the supplementary material (Sec. 7.2, Tab. 4).

Direct joint solving

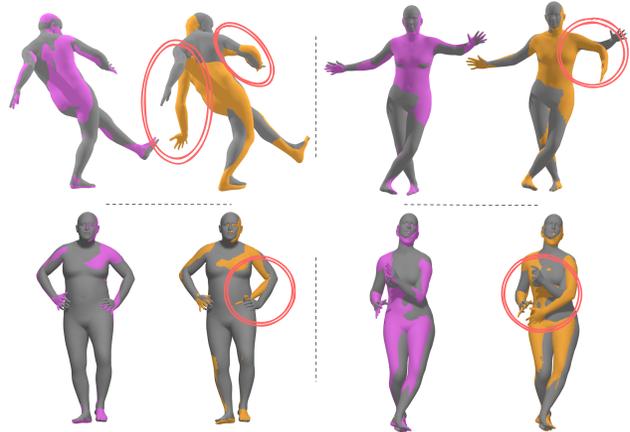We proceed with evaluating our model's ability to accurately estimate the skeleton joints ℓ^j from the input markers (*i.e.* joint-solving). We compare our model against two SotA joint-solving approaches: a) MoCap-Solver [14] that uses graph convolutions and temporal information, and b) DeMoCap [13] that employs an HRNet [84] backbone and frontal-back fusion. All models are trained and evaluated on the CMU [11] dataset as in [14]. For MoCap-Solver we rerun the evaluation without normalizing the markers and the skeletons as this information should be unknown during testing. At the same time, we employ the joint position error (JPE) from [14] for a more fair comparison. From the results in Tab. 3 we observe that our model outperforms the SotA in both positional metrics (RMSE, JPE) while having the best or the second-best accuracy for different PCK.

Explicit vs implicit labeling

Our next experiment aims to showcase the advances of fitting a parametric body model on landmarks estimated with regression instead of explicitly labeling them. We compare our model that de-noises, completes, and implicitly labels landmarks via regression with SOMA, a SotA explicitly labeling method, by fitting the body to the markers similar to [44]. Note that in order to have a fair comparison we solve **only for markers** and not for markers & joints (as discussed in Sec. 3.2). We train our model using the same datasets that SOMA was trained on, and then test on TH2 and our "Tail" test set using the clean body-extracted markers, and the same MoSh-like fitting without uncertainty region optimization and without considering latent markers as the marker layout is fixed to the nominal one. Tab. 4 showcases

		RMSE↓	MAE↓	PCK1↑	PCK3↑	PCK7↑
TH2	[20]	29.7 mm	3.49°	28.33%	87.78%	96.11%
	Ours ($\ell^{m,*}$)	19.1 mm	**2.68°**	26.49%	93.72%	99.26%
	Ours (ℓ)	**17.6 mm**	-	**33.92%**	**95.13%**	**99.35%**
Tail	[20]	68.6 mm	6.76°	11.78%	60.87%	84.84%
	Ours ($\ell^{m,*}$)	30.1 mm	**2.89°**	12.11%	73.13%	**96.87%**
	Ours (ℓ)	**28.3 mm**	-	**27.31%**	**83.12%**	95.35%

Table 4: Explicit (SOMA [20]) vs implicit (Ours) labeled marker fits and direct landmarks' ℓ solving comparison.

	RMSE↓	MAE↓	PCK1↑	PCK3↑	PCK7↑
[44, 49]	30.1 mm	3.49°	11.79%	66.85%	**98.34%**
[7]	30.8 mm	3.10°	12.71%	67.06%	97.71%
Ours (ℓ^m)	**28.9 mm**	**2.98°**	**14.71%**	**69.86%**	98.18%

Table 5: Noisy landmark fits comparison on TH2.

that the fits on our model's markers ℓ^m deliver better performance, a fact that is mainly attributed to the robustness of regression compared to the larger error margin of fitting to incorrectly labeled markers. This is evident in all test sets but more pronounced in the tail (rare) poses. Indicative qualitative examples are depicted in Fig. 14. For completion (not direct comparison with SOMA), we include the results for solving both markers and joints (ℓ) estimated by our model, which clearly achieves the best overall performance.

Addressing input noise

Finally, we design an experiment for showcasing our model's fitting robustness to noisy marker input as discussed in Sec. 3.3. Tab. 12 presents results when fitting to noisy landmarks between the uncertainty optimization method and MoSh(++) like fitting (ignoring the latent marker optimization as the markers are extracted from the body's surface and placed using the nominal layout). The TH2 dataset is used for evaluation, with the body extracted input markers corrupted with high levels of noise (see Sec. 3.2 of the supp. for the applied types of noise) prior to fitting the body model to them. Naturally, optimizing the uncertainty region improves fitting performance to noisy observations. Compared to a more complex optimization objective that also considers the shape of the data distribution [7] we find that the proposed Gaussian uncertainty region optimization delivers improved fits. This can be attributed to the complexity of tuning it, as well as the increased parameter count. Fig. 6 depicts qualitative examples with body fits in the noisy inputs acquired with just 3 viewpoints (same capture session as Fig. 1) and shows that jointly opti-
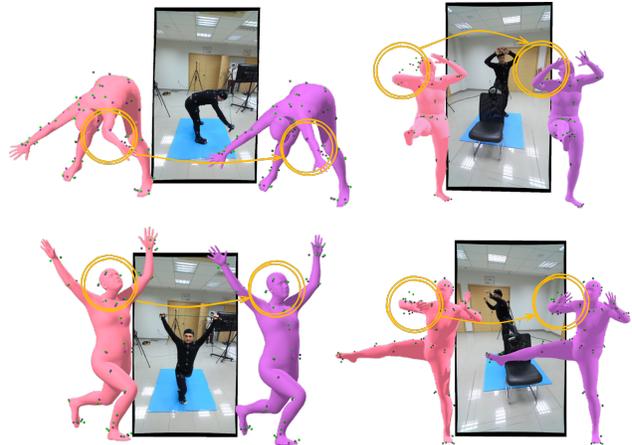

Figure 6: Plain vs uncertainty-based fit. Input markers from the consumer-grade system and the model inferred ones are colored with with green, and violet respectively.

mizing the uncertainty region allows for robustness to input-related measurement noise, as well as model-related information noise. Some interesting noise-aware fitting ablations along with visualizations can be found in Sec. 9 of the supplementary material.

Real-time performance. We validate our end-to-end method by implementing a real-time system using sparse consumer-grade sensors (see details in Sec. 11 of the supp.). Leveraging the orthogonal view two-pass approach we deploy an optimized ONNX [1] model where we flatten the two passes across the batch dimension, performing only the light-weight marginal heatmap fusion in a synchronized manner. Our system achieves under 16ms inference even on a laptop equipped with a mobile-grade RTX 2080. Nonetheless, we understand that high-quality MoCap requires greater efficiency to achieve processing rates of at least 120Hz and we set this rate as the next goal.

5. Conclusion

MoCap data are highly imbalanced and in this work we have presented a novel technique for imbalanced regression. Still we believe we have but scratched the surface of exploiting representation learning for addressing the long-tail and bias, as different architectures, samplers and relevance functions can be explored. At the same time, this work contributes to integrating machine learning in real-time optical MoCap, while also making it more accessible. However, there is room for improvements, as temporal information is not integrated in our approach, and a single, fixed marker layout is only supported.

References

[1] Open Neural Network Exchange (ONNX). https://github.com/onnx/onnx. 8

[2] Advanced Computing Center for the Arts and Design. AC-CAD MoCap Dataset. 6, 14

[3] 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. 7, 14

[4] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imGHUM: Implicit generative models of 3d human shape and articulated pose. In *Proc. IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 5461–5470, 2021. 3

[5] Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, and Ariel Shamir. Self-similarity analysis for motion capture cleaning. In *Computer Graphics forum*, volume 37, pages 297–309. Wiley Online Library, 2018. 2

[6] Andreas Aristidou and Joan Lasenby. Real-time marker prediction and cor estimation in optical motion capture. *The Visual Computer*, 29:7–26, 2013. 2

[7] 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. 6, 8, 19

[8] 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. 7, 14

[9] Paula Branco, Rita P Ribeiro, and Luis Torgo. Ubl: an r package for utility-based learning. *arXiv preprint arXiv:1604.08079*, 2016. 3

[10] Paula Branco, Luís Torgo, and Rita P Ribeiro. SMOGN: A pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017. 3, 4

[11] Carnegie Mellon University. CMU MoCap Dataset. 7, 19

[12] 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. 17

[13] 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. 1, 2, 3, 4, 5, 7, 19

[14] 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. 1, 2, 3, 4, 5, 7, 12, 13, 19

[15] Wei Cheng, Su Xu, Jingtan 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, 6, 14

[16] 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. 16

[17] John E Dennis Jr and Roy E Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-simulation and Computation*, 7(4):345–359, 1978. 5

[18] Yinfu Feng, Mingming Ji, Jun Xiao, Xiaosong Yang, Jian J Zhang, Yueting Zhuang, and Xuelong Li. Mining spatial-temporal patterns and structural sparsity for human motion data denoising. *IEEE Transactions on Cybernetics*, 45(12):2693–2706, 2014. 2

[19] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9919–9928, 2021. 3

[20] 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. 1, 2, 3, 4, 5, 6, 8, 13, 19

[21] Saeed Ghorbani, Ali Etemad, and Nikolaus F Troje. Auto-labelling of markers in optical motion capture by permutation learning. In *Advances in Computer Graphics: 36th Computer Graphics International Conference, CGI 2019, Calgary, AB, Canada, June 17–20, 2019, Proceedings 36*, pages 167–178. Springer, 2019. 1, 2, 5

[22] Yu Gong, Greg Mori, and Frederick Tung. RankSim: Ranking similarity regularization for deep imbalanced regression. *arXiv preprint arXiv:2205.15236*, 2022. 3

[23] 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. 15

[24] Mark Hamilton, Evan Shelhamer, and William T Freeman. It is likely that your loss should be a likelihood. *arXiv preprint arXiv:2007.06059*, 2020. 6

[25] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D Twigg, and Kenrick Kin. Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018. 1, 2, 3, 5

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNET classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 6

[27] Lorna Herda, Pascal Fua, Ralf Plänkers, Ronan Boulic, and Daniel Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human movement science*, 20(3):313–341, 2001. 2

[28] David T Hoffmann, Dimitrios Tzionas, Michael J Black, and Siyu Tang. Learning to train with synthetic humans. In *Pattern Recognition: 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10–13, 2019, Proceedings 41*, pages 609–623. Springer, 2019. 3

9

[29] Daniel Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 1, 2, 3, 4, 5, 13

[30] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. 5

[31] Alexander Hornung, Sandip Sar-Dessai, and Leif Kobbelt. Self-calibrating optical motion tracking for articulated bodies. In *Proc. IEEE Virtual Reality (IEEEVR)*, pages 75–82. IEEE, 2005. 2

[32] 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. 13, 16

[33] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proc. ICCV*, pages 7717–7726, 2019. 3

[34] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. Skinning: Real-time shape deformation (full text not available). In *ACM SIGGRAPH 2014 Courses*, pages 1–1. 2014. 3

[35] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020. 5

[36] 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. 14

[37] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 5

[38] Manuel Kaufmann, Yi Zhao, Chengcheng Tang, Lingling Tao, Christopher Twigg, Jie Song, Robert Wang, and Otmar Hilliges. Em-pose: 3d human pose estimation from sparse electromagnetic trackers. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11510–11520, 2021. 3

[39] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018. 6

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[41] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Proc. International Conference on Learning Representations, ICLR*, 2014. 4

[42] Dieederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2015. 16

[43] 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. 20

[44] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 2, 3, 6, 7, 8, 19, 22

[45] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 3, 6

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations, (ICLR)*, 2019. 17

[47] EYES JAPAN Co. Ltd. Eyes. *http://mocapdata.com*, 2018. 7

[48] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019. 5

[49] 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. 2, 3, 6, 8, 14, 19

[50] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *Proc. International Conference on Advanced Robotics (ICAR)*, pages 329–336. IEEE, 2015. 6

[51] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 16, 17

[52] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. 5

[53] Johannes Meyer, Markus Kuderer, Jörg Müller, and Wolfram Burgard. Online marker labeling for fully automatic skeleton tracking in optical motion capture. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 5652–5657, 2014. 2

[54] Anton Mikhailov. Turbo, An Improved Rainbow Colormap for Visualization. https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html, 2019. 5, 18

[55] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018. 5

[56] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1477–1485. IEEE, 2019. 5

[57] Jorge Nocedal and Stephen J Wright. Nonlinear equations. *Numerical Optimization*, pages 270–302, 2006. 6

[58] Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *Proc. European Conference on Computer Vision (ECCV)*, pages 598–613. Springer, 2020. 3

[59] Ahmed AA Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Supr: A sparse unified part-based human representation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 568–585, 2022. 3

[60] 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. *Advances in neural information processing systems*, 32, 2019. 6

[61] 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. 3, 6, 16, 17

[62] Dario Pavllo, Mathias Delahaye, Thibault Porssut, Bruno Herbelin, and Ronan Boulic. Real-time neural network prediction for handling two-hands mutual occlusions. *Computers & Graphics: X*, 2:100011, 2019. 3

[63] 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. 6, 14

[64] 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. 3, 7, 18

[65] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proc. International Conference on Machine Learning (ICML)*, pages 1530–1538. PMLR, 2015. 4

[66] Maurice Ringer and Joan Lasenby. A procedure for automatically estimating model parameters in optical motion capture. *Image and Vision Computing*, 22(10):843–850, 2004. 2

[67] Yu Rong, Ziwei Liu, and Chen Change Loy. Chasing the tail in monocular 3d human reconstruction with prototype memory. *IEEE Transactions on Image Processing (TIP)*, 31:2907–2919, 2022. 2, 3

[68] 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. 5, 12

[69] Tobias Schubert, Alexis Gkogkidis, Tonio Ball, and Wolfram Burgard. Automatic initialization for skeleton tracking in optical motion capture. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 734–739. IEEE, 2015. 2

[70] Ken Shoemake. Animating rotation with quaternion curves. In *Proc. Conference on Computer Graphics and Interactive Techniques*, page 245–254, 1985. 5

[71] 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. 6, 14

[72] Aníbal Silva, Rita P Ribeiro, and Nuno Moniz. Model optimization in imbalanced regression. In *Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings*, pages 3–21. Springer, 2022. 3

[73] Jannik Steinbring, Christian Mandery, Florian Pfaff, Florian Faion, Tamim Asfour, and Uwe D Hanebeck. Real-time whole-body human motion tracking based on unlabeled markers. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 583–590. IEEE, 2016. 2

[74] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110:2187–2211, 2021. 3

[75] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proc. European conference on computer vision (ECCV)*, pages 529–545, 2018. 5

[76] 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. 20

[77] Christopher Tensmeyer and Tony Martinez. Robust keypoint detection. In *Proc. International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 1–7, 2019. 5

[78] 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, 16

[79] Luís Torgo, Paula Branco, Rita P Ribeiro, and Bernhard Pfahringer. Resampling strategies for regression. *Expert Systems*, 32(3):465–476, 2015. 3, 4

[80] Luis Torgo and Rita Ribeiro. Utility-based regression. In *PKDD*, volume 7, pages 597–604. Springer, 2007. 3, 4

[81] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. Smote for regression. In *Progress in Artificial Intelligence: 16th Portuguese Conference on Artificial Intelligence, EPIA 2013, Angra do Heroísmo, Azores, Portugal, September 9-12, 2013. Proceedings 16*, pages 378–389. Springer, 2013. 3, 4

[82] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 109–117, 2017. 3

[83] Marin Šarić. Libhand: A library for hand articulation, 2011. Version 0.9. 3

[84] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(10):3349–3364, 2020. 7

[85] Yupan Wang, Guiqing Li, Huiqian Zhang, Xinyi Zou, Yuxin Liu, and Yongwei Nie. Panoman: Sparse localized components–based model for full human motions. *ACM Transactions on Graphics (TOG)*, 40(2):1–17, 2021. 3

[86] Tom White. Sampling generative networks: Notes on a few effective techniques. *arXiv:1609.04468*, 2016. 5

[87] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & Ghuml: Generative 3d human shape and articulated pose models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6184–6193, 2020. 3

[88] Haonan Yan, Jiaqi Chen, Xujie Zhang, Shengkai Zhang, Nianhong Jiao, Xiaodan Liang, and Tianxiang Zheng. Ultrapose: Synthesizing dense pose with 1 billion points by human-body decoupling 3d model. In *Proc. IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 10891–10900, 2021. 3

[89] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *Proc. International Conference on Machine Learning (ICML)*, pages 11842–11851. PMLR, 2021. 3

[90] Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster voxelpose: Real-time 3d human pose estimation by orthographic projection. In *Proc. European Conference on Computer Vision (ECCV)*, pages 142–159. Springer, 2022. 5

[91] KangKang Yin and Goh Jing Ying. SFU motion capture database. *http://mocap.cs.sfu.ca*. 7

[92] 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, 6, 14

[93] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7093–7102, 2020. 5

[94] Richard Zhang. Making convolutional networks shift-invariant again. In *Proc. International Conference on Machine Learning (ICML)*, 2019. 13

[95] Yan Zhang, Michael Black, and Siyu Tang. Perpetual motion: Generating unbounded human motion. *arXiv:2007.13886*, 2020. 17

A. 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.*

Appendix B provides the implementation details of the UNet model used to predict landmarks ℓ_{est}. Appendix C clarifies the augmentation and corruptions used when training and when experimenting with noisy fits. Appendix D presents the different datasets that were used for the main paper's experiments and the accompanying experiments found in this supplementary material. Appendix E 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. Appendix F 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. Appendix G 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 (Appendix G.1), a relevance function ablation (Appendix G.2), an investigation of the orthogonality between the different techniques (Appendix G.3), and an ablation of the different sampling components (Appendix G.4). Appendix H presents an extra experiment supplementing the solving comparison experiment conducted in the main paper. Appendix I 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, Appendix J includes additional qualitative results, while Appendix K describes the implementation details related to the real-time MoCap system used to capture and provide in-the-wild results.

B. 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 [68] architecture was used to simultaneously predict 53 markers and 18 joints landmarks. It should be noted that since MoCap-Solver [14] 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 [32]. When downscaling anti-aliased max pooling [94] 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):

$$\mathcal{L} = \sum_{i=1}^{L} (\lambda_{JS} \mathcal{L}_{JS}(\mathbf{H}_{gt}, \mathbf{H}_{est}) + \lambda_w \mathcal{L}_w^{\nu}(\tilde{\ell}_{gt}, \tilde{\ell}_{est})). \quad (7)$$

$\mathcal{L}_{JS}$ is the Jensen-Shannon divergence defined in Eq. (8):

$$\mathcal{L}_{JS}(\mathbf{H}_{gt}, \mathbf{H}_{est}) = \frac{1}{2} D_{KL}(\mathbf{H}_{gt}, M) + \frac{1}{2} D_{KL}(\mathbf{H}_{est}, M), \quad (8)$$

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 $\mathbf{H}_{est}$.

$\mathcal{L}_w^{\nu}$ is the robust Welsch penalty function, applied to the normalized ℓ coordinates, defined by Eq. (9), with $\nu > 0$ being a user-specified parameter set to 0.05:

$$\mathcal{L}_w^{\nu}(\tilde{\ell}_{gt}, \tilde{\ell}_{est}) = 1 - \exp\left(-\frac{|\tilde{\ell}_{gt} - \tilde{\ell}_{est}|^2}{2v^2}\right) \quad (9)$$

C. 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 [29, 20, 14], 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.

C.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 $u \sim \mathcal{U}(-1, 1)$:

$$\beta' = \beta + u \quad (10)$$

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

$$\beta_i' = \begin{cases} \beta_i, & \text{if } i \notin S \\ \mathcal{N}(0, 1), & \text{if } i \in S \end{cases} \quad (11)$$

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.

C.2. Corruption

We simulate marker occlusions with the following process. Let $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ 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, $k \sim \mathcal{U}(m, n')$, $\quad m \leq k \leq n' \leq n$, where $\mathcal{U}(m, n')$ is the uniform distribution over the range of integers $\{m_1, m_2, \ldots, n'\}$, 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. $\mathbf{m} = (m_1, m_2, \ldots, m_k) \sim \mathcal{U}(1, n)$, $\quad k \leq n$ 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 [20]. 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 prepossessing pipeline is randomly applied in each epoch, with specific probabilities assigned to

Figure 7: A set of random samples from the THuman2.0 [92] dataset. The darker meshes indicate more challenging poses.

Figure 8: Exemplar rare and complex poses from our custom tail dataset.

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.

D. Datasets

D.1. Marker-based

For our experiments we used a variety of MoCap datasets unified within AMASS [49] 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 [3] to train a statistical model of human pose, the HumanEva dataset [71], which includes various activities performed by multiple subjects, and the ACCAD dataset [2], consisting of more action mo-

tion types such as dancing, martial arts, and sports. Moreover, we use the TotalCapture dataset [36], which includes data from 5 different subjects performing 37 motion actions, the DFaust dataset [8] 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.

D.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 [92] dataset, including 5 subjects in extreme poses, the GeneBody dataset [15] consisting of 50 subjects performing various short duration activities, and the ZJU-MoCap dataset [63] that includes data from 10 sequences of human performances. Fig. 7 depicts an indicative subset from the THuman 2.0 dataset, which consists of both common and challenging-to-understand poses (shown with darker meshes).

D.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. 8.

D.4. Qualitative Distribution

An overview of these datasets in terms of some qualitative variance indicators is presented in Tab. 6. These were used to select by approximately equalizing the datasets used in the markerless vs optical data study.

14

	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 6: Datasets overview.

E. Performance Metrics & Indicators

E.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:

$$RMSE = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{J} \sum_{j=1}^{J} ||\ell_{gt}^{(i,j)} - \ell_{est}^{(i,j)}||_2}, \quad (12)$$

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:

$$PCK = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{J} \sum_{j=1}^{J} [||\ell_{gt}^{(i,j)} - \ell_{est}^{(i,j)}||_2 < \tau]. \quad (13)$$

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. (14):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{J} \sum_{j=1}^{J} d(R_{gt}^{(i,j)}, R_{est}^{(i,j)}), \quad (14)$$

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

E.2. Synthesis Metrics

Inspired by C. Guo *et al.* [23], we use two metrics to choose our best model for tail-pose generation and regres-

sion 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:

$$DIV = \frac{1}{N} \sum_{i=1}^{N} ||v_i - \tilde{v}_i||, \quad (15)$$

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

E.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. (16):

$$rmse3 = (1 - PCK3) \times RMSE, \quad (16)$$

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

$$synthesis = \frac{FID}{DIV}. \quad (17)$$

F. Training Data Sourcing

Tab. 7 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. 9 offers comparative visualizations of the encoded pose parameters θ vectors' distribution for each dataset combination.
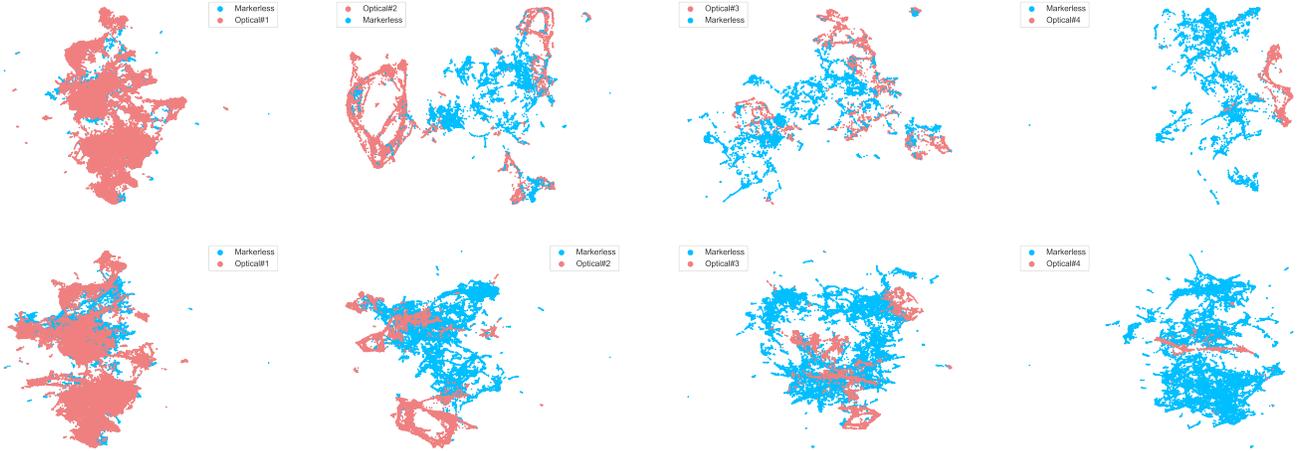
Figure 9: UMAP projections [51] on datasets collected using high-end MoCap systems and others collected from a multiview markerless fitting process. The first row uses the markerless#1 dataset and the second row uses the markerless#2 dataset. It can be seen that the variability of data is independent of the type of acquisition.

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

G. Balancing Regression

G.1. Robust VPoser

G. Pavlakos *et al*. [61] were the first to leverage a Variational Autoencoder (VAE) [42] 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 [16] or by mapping the pose space on a surface-like manifold [78]. 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 VPoser (RVPoser) with the model from [61] 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 [32] 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

	Synthesis		Fitting			
	FID↑	DIV↑	MAE↓	PCK1↑	PCK3↑	PCK 7↑
VPoser [61]	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 8: Quantitative comparison between the VPoser model from [61] and our robust variant (RVPoser) in synthesis and fitting on the THuman 2.0 test set.

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:

$$\mathcal{L}_{VAE} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{orth} \quad (18)$$

$$\mathcal{L}_{KL} = \Psi(D_{KL}(q_\theta(z|R)||\mathcal{N}(0,I))) \quad (19)$$

$$\mathcal{L}_{rec} = \|v - \hat{v}\|_2, \quad (20)$$

$$\mathcal{L}_{orth} = \frac{Trace(R^T \hat{R}) - 1}{2}, \quad (21)$$

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 $\hat{R}$ 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 [95]) using the Charbonnier penalty function Ψ, with $\Psi(x) = \sqrt{1 + x^2} - 1$ [12] to prevent posterior collapse and learn a more disentangled manifold. Eqs. (19) and (20) follow the VAE training scheme - *e.g.,* trading of reconstruction quality with learning a Gaussian-like manifold, while Eqs. (20) and (21) force the model to construct a valid rotation latent space. We complement RVPoser training with the weight-decaying version of Adam optimization [46], 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. 8, 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. 10 we show the UMAP projection [51] 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.

G.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 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:

$$\rho(\theta) = 1 + 2\left(\frac{e^x}{e^x + 1} - 0.5\right), \quad x = \frac{\epsilon}{\sigma}, \quad (22)$$

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. 11, 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:

$$\rho(\theta) = e^{\epsilon/\sigma}. \quad (23)$$

Note that since the exponential function does not have an upper limit, we clamp the result at $\rho(\theta) = 3$, so the effective 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. 12, 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. 11 and 12.
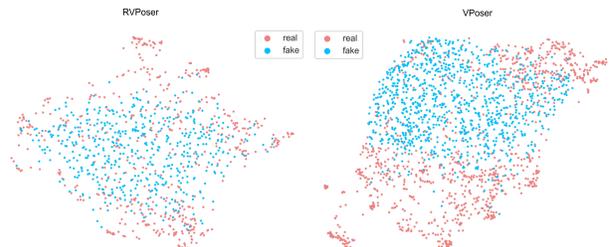

Figure 10: UMAP projections [51] of "real" ground truth samples and of "fake" ones generated by our RVPoser (left) and the VPoser [61] (right) models, respectively.
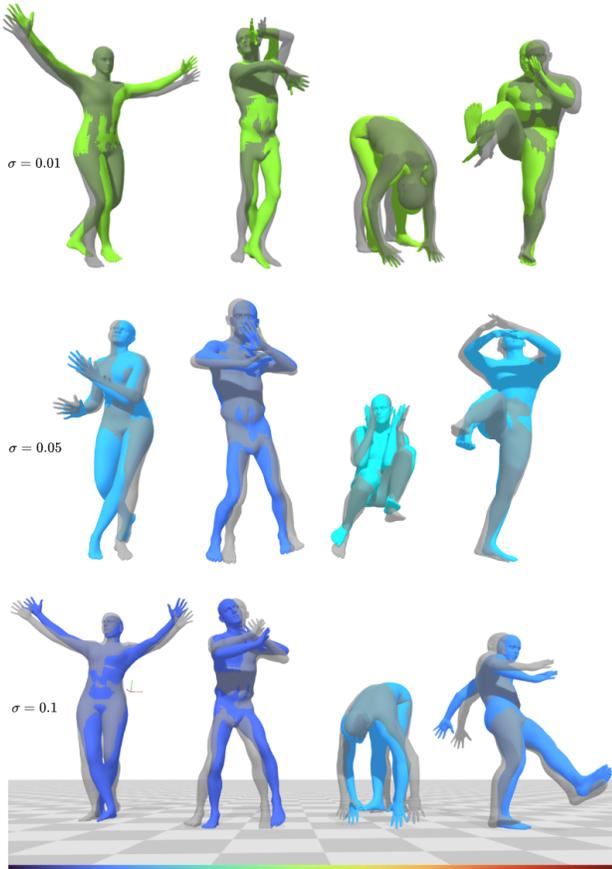
Figure 11: Color-coded (turbo colormap [54] at the bottom) autoencoding ρ of various poses and σ values, using the Sigmoid-based relevance function.

G.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 [64]. 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. 9, 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 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.
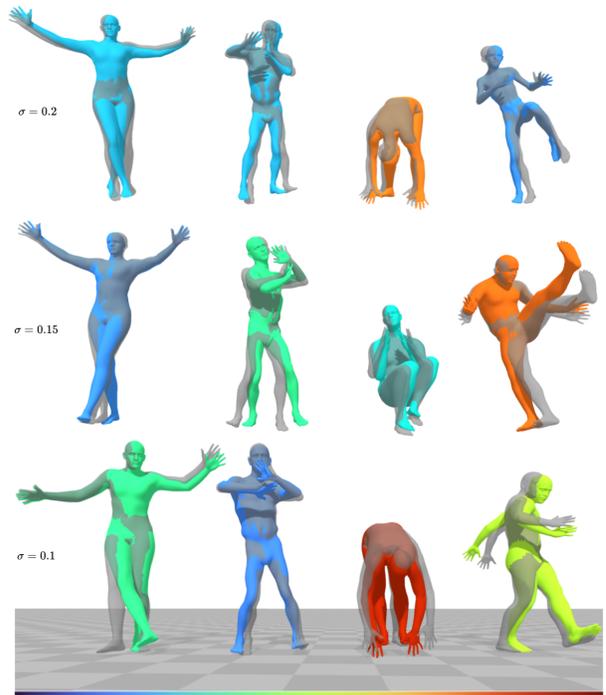

Figure 12: Color-coded (turbo colormap [54] at the bottom) autoencoding ρ of various poses and σ values, using the Exponential-based relevance function.

	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%**
[64]	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%**
[64]	32.9 mm	**27.66%**	81.98%	94.92%

Table 9: Imbalanced regression ablation. 'Sampling' and 'Relevance' variants are combined in 'Ours' model, while the results of [64] are presented for reference.

	RMSE↓	JPE↓	PCK1↑	PCK3↑	PCK7↑
[14]	18.20 mm	14.80 mm	37.19%	85.38%	**99.37%**
[13]	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 11: Direct joint solving on CMU [11] test set with a different seed (SEED200 from [14]) than in the main paper.

	n_d	n_m	RMSE↓	MAE↓	PCK1↑	PCK3↑	PCK7↑
[44, 49]			30.10 mm	3.49°	11.79%	66.85%	98.34%
[7]	✓	✗	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%**
[44, 49]			20.60 mm	1.93°	28.71%	89.03%	**99.05%**
[7]	✗	✓	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%
[44, 49]			23.80 mm	2.03°	24.26%	85.63%	**98.22%**
[7]	✓	✓	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 12: Noisy landmark fitting on THuman 2.0.

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

Table 10: Alternative sampling methods ablation. 'SLERP' variant corresponds to the 'Sampling' variant in Tab. 9, while 'Base' corresponds to the baseline model (*i.e.* no synthetic samples).

G.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 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. 10 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.

H. Extra Solving Experiments

In the following Tab. 11 we compare the performance of our model to a dataset generated with a different seed following [14] (denoted as SEED200). We observe that the results do not significant vary from those presented in the main paper.
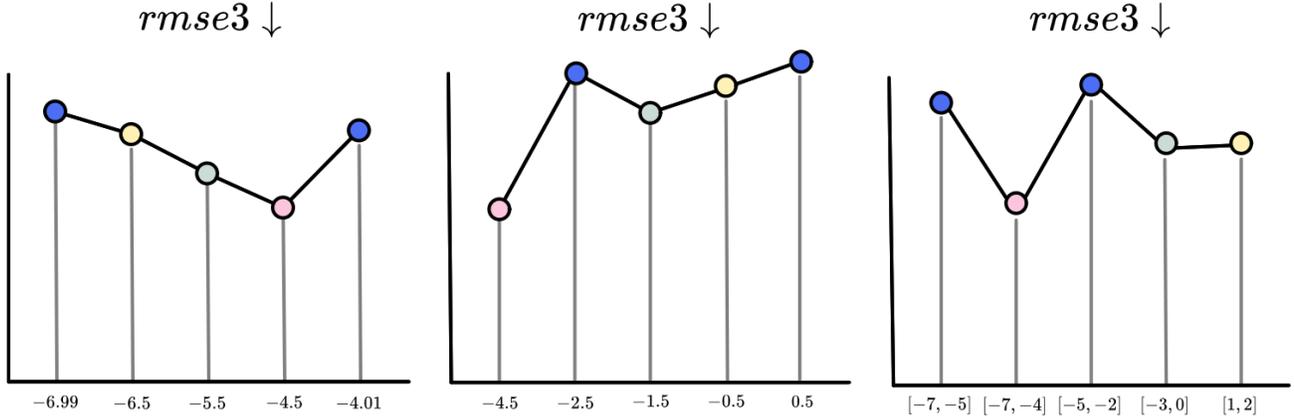
I. Landmarks and fitting ablation

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. 12 show that our approach maintains its performance across different noise sources, while the method proposed in [7] 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 the burden of hyperparameter tuning. In Fig. 15, we qualitatively compare the performance of our method with that of [20], colorised each mesh based on its distance error from the ground truth. Finally, for a fair comparison with [7] we conducted several experiments to find the best range of α values, as well as their initial values. Fig. 13 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.

J. Additional Qualitative Results

We present additional qualitative results comparing our direct regression approach to labeling [20] in the THuman

(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 13: Ablation on α values.

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. 14. Finally, Fig. 16 presents qualitative results using real-world data acquired from the developed system presented in Appendix K, including both model predictions and post-fitting body results, showcasing the benefits of the noise-aware fitting process.

K. 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 [76] 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 [43] 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.

Figure 14: Fits to our regressed versus SOMA labeled markers. The fitting process is more sensitive to labeling errors.

Figure 15: The figure shows the qualitative results of our noise-aware fitting method on the left and the method proposed in [44] on the right. Each mesh is colored using a Jet color map based on the Euclidean distance error metric from the ground truth mesh.
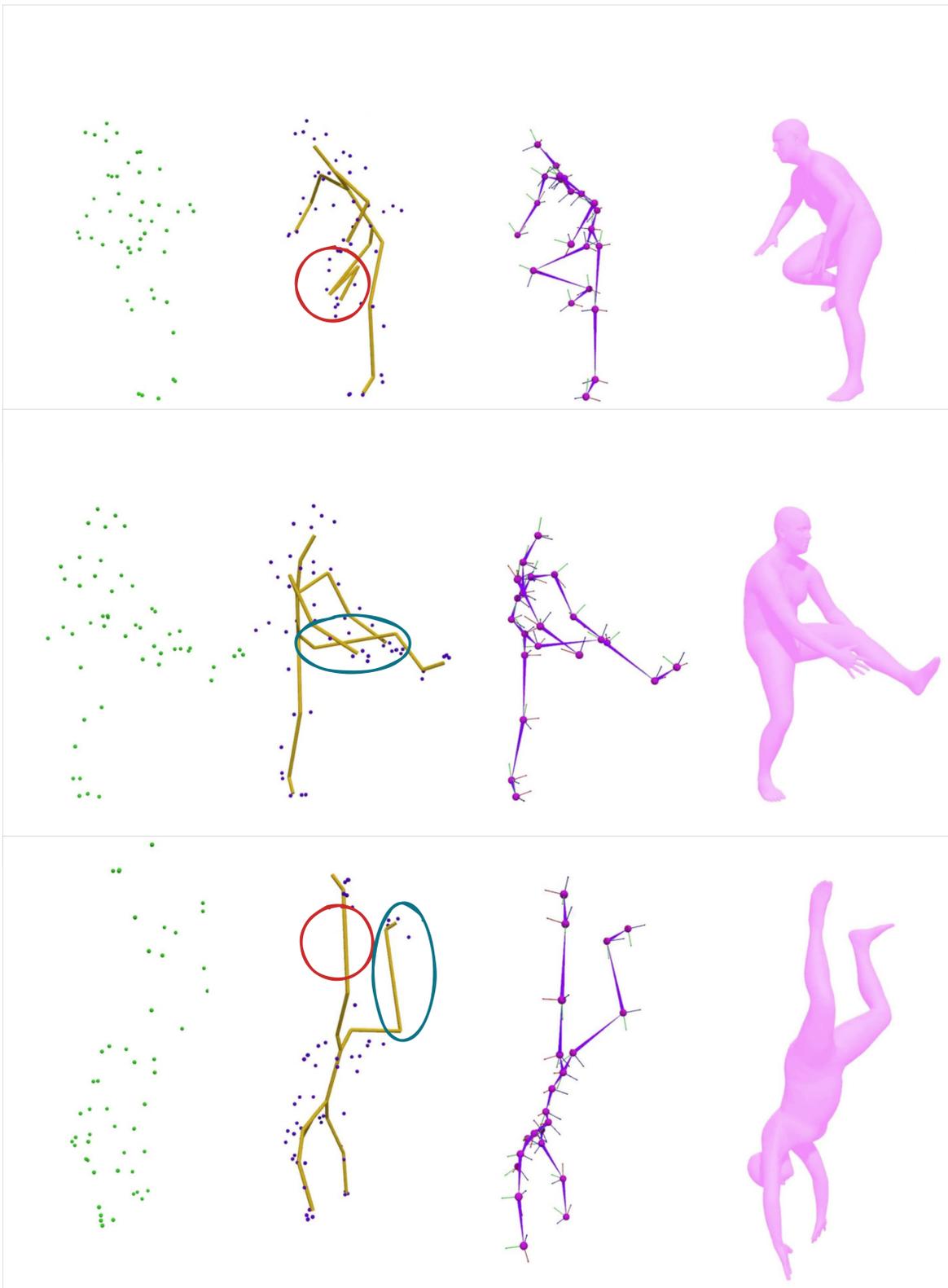
Figure 16: Additional qualitative results of our system in the wild using a setup comprising a very sparse set of low-cost sensors. Starting from the left, we present the raw input collected from our multi-sensor acquisition system (Appendix K), with the raw (unfiltered) estimated ℓ_{est} from our model 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.