HuMoR: 3D Human Motion Model for Robust Pose Estimation Supplementary Material

Davis Rempe[1] Tolga Birdal[1] Aaron Hertzmann[2] Jimei Yang[2]
Srinath Sridhar[3] Leonidas J. Guibas[1]
[1]Stanford University [2]Adobe Research [3]Brown University

1. Introduction

In this document, we provide details and extended evaluations omitted from the main paper[1] for brevity. Sec. 2 provides extended discussions, Sec. 3 and Sec. 4 give method details regarding the HuMoR model and test-time optimization (TestOpt), Sec. 5 derives our optimization energy from a probabilistic perspective, Sec. 6 provides experimental details from the main paper, and Sec. 7 contains extended experimental evaluations.

We encourage the reader to view the **supplementary videos** on the project webpage[2] and supplementary webpage[3] for extensive qualitative results. We further discuss these results in Sec. 7.

2. Discussions

State Representation. Our state representation is somewhat redundant to include both explicit joint positions $\mathbf{J}$ and SMPL parameters (which also give joint positions $\mathbf{J}^{\text{SMPL}}$). This is motivated by recent works [17, 40] which show that using an extrinsic representation of body keypoints (*e.g.* joint positions or mesh vertices) helps in learning motion characteristics like static contact, thereby improving the visual quality of generated motions. The over-parameterization, unique to our approach, additionally allows for consistency losses leveraged during CVAE training and in TestOpt.

Another noteworthy property of our state is that it does not explicitly represent full-body shape – only bone proportions are implicitly encoded through joint locations. During training, we use shape parameters β provided in AMASS [22] to compute $\mathcal{L}_{\text{SMPL}}$, but otherwise the CVAE is shape-unaware. Extending our formulation to include full-body shape is an important direction for improved generalization and should be considered in future work.

Conditioning on More Time Steps. Alternatively, we could condition the dynamics learned by the CVAE with additional previous steps, *i.e.* $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-p})$, however since $\mathbf{x}_{t-1}$ includes velocities this is unnecessary and only increases the chances of *overfitting* to training motions. It would additionally increases the necessary computation for both generation and TestOpt.

Why CVAE? Our use of a CVAE to model motion is primarily motivated by recent promising results in the graphics community [17, 11]. Not only is it a simple solution, but also affords the physical interpretation presented in the main paper. Other deep generative models could be considered for $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$, however each have potential issues compared to our CVAE. The conditional generative adversarial network [25] would use standard normal noise for $\mathbf{z}_t$, which we show is insufficient in multiple experiments. Furthermore, it does not allow for inferring a latent transition $\mathbf{z}_t$. Past works have had success with recurrent and variational-recurrent architectures [40]. As discussed previously, the reliance of these networks on multiple timesteps increases overfitting which is especially dangerous for our estimation application which requires being able to represent arbitrary observed motions. Finally, normalizing flows [16] and neural ODEs [8] show exciting potential for modeling human motion, however conditional generation with these models is not yet well-developed.

In comparison to Motion VAE (MVAE) [17]. Our proposed CVAE is inspired by MVAE, but introduces a number of key improvements that enable generalization and expressivity: (**i**) HuMoR uses a neural network to learn a conditional prior $p_\theta(\mathbf{z}_t|\mathbf{x}_{t-1})$ rather than assuming $p_\theta(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t; \mathbf{0}, \mathbf{I})$, (**ii**) the decoder predicts the *change in state* Δ_θ rather than the next state $\mathbf{x}_t$ directly, (**iii**) the decoder outputs person-ground contacts $\mathbf{c}_t$ which MVAE does not model, and (**iv**) HuMoR trains using $\mathcal{L}_{\text{SMPL}}$ regularization to encourage joint position/angle consistency whereas MVAE uses the typical ELBO. Tab. 3 in the main paper shows that these differences are crucial to achieve good results as MVAE does not work well. The conditional prior and $\mathcal{L}_{\text{SMPL}}$ are particularly important: learning $p_\theta(\mathbf{z}_t|\mathbf{x}_{t-1})$ is

[1] In the rest of this document we refer to the main paper briefly as *paper*.
[2] https://geometry.stanford.edu/projects/humor/
[3] https://geometry.stanford.edu/projects/humor/supp.html

theoretically justified when deriving the CVAE and is intuitive for human motion, while $\mathcal{L}_{\text{SMPL}}$ provides a strong supervision to improve stability of model rollout.

Furthermore, the pose state employed by HuMoR and MVAE differ slightly. In MVAE, the root is defined by projecting the pelvis onto the ground, giving 2D linear and 1D angular velocities. In HuMoR, the root is at the pelvis giving full 3D velocities.

A Note on β-VAE [13]. The KL weight w_{KL} in Eq. 7 of the main paper is not directly comparable to a typical β-VAE [13] due to various implementation details. First, $\mathcal{L}_{\text{rec}}$ is the mean-squared error (MSE) of the unnormalized state rather than the true log-likelihood. The use of additional regularizers $\mathcal{L}_{\text{reg}}$ that are not formulated probabilistically to be part of the reconstruction loss further compounds the difference. Furthermore, in practice losses are averaged over both the feature and batch dimensions as not to depend on chosen dimensionalities. All these differences result in setting $w_{\text{KL}} = 4e^{-4}$.

The Need for $\mathcal{E}_{\text{reg}}$ in Optimization. The motion prior term $\mathcal{E}_{\text{mot}} = \mathcal{E}_{\text{CVAE}} + \mathcal{E}_{\text{init}}$, which leverages our learned conditional prior and GMM, nicely falls out of the MAP derivation (see Sec. 5 below) and is by itself reasonable to ensure motion is plausible. However, in practice it can be prone to local minima and slow to converge without any regularization. This is primarily because HuMoR is trained on clean motion capture data from AMASS [22], but in the early stages of optimization the initial state $\mathbf{x}_0$ will be far from this domain. This means rolled out motions using the CVAE decoder will be implausible and the likelihood output from learned conditional prior is not necessarily meaningful (since inputs will be well outside the training distribution). The additional regularizers presented in the main paper, mainly $\mathcal{E}_{\text{skel}}$ and $\mathcal{E}_{\text{env}}$, allow us to resolve this issue by reflecting expected behavior of the motion model when it is producing truly plausible motions (*i.e.* $\mathbf{x}_0$ is similar to the training data).

On Evaluation Metrics. As discussed in prior work [30], traditional positional metrics used to evaluate root-relative pose estimates do not capture the accuracy of the absolute ("global") motion nor its physical/perceptual plausibility. This is why we use a range of metrics to capture both the global joint accuracy, local joint accuracy (after aligning root joints), and plausibility of a motion. However, these metrics still have flaws and there is a need to develop more informative motion estimation evaluation metrics for both absolute accuracy and plausibility. This is especially true in scenarios of severe occlusions where there is not a single correct answer: even if the "ground truth" 3D joints are available, there may be multiple motions that explain the partial observations equally well.

On Convergence. Our multi-objective optimization uses a mixture of convex and non-convex loss functions. As we utilize L-BFGS, the minimum energy solution we report is only locally optimal. While simulated annealing or MCMC / HMC (Markov Chain Monte Carlo / Hamiltonian Monte Carlo) type of exploration approaches can be deployed to search for the global optimum, such methods would incur heavy computational load and hence are prohibitive in our setting. Thanks to the accurate initialization, we found that most of the time TestOpt converges to a good minimum. This observation is also lightly supported by recent work arguing that statistically provable convergence can be attained for the human pose problem under convex and non-convex regularization using a multi-stage optimization scheme [38].

2.1. Assumptions and Limitations

On the Assumption of a Ground Plane. We use *the ground* during TestOpt to obtain a transformation to the canonical reference frame where our prior is trained. While this is a resonable assumption in a majority of scenarios, we acknowledge that certain applications might require *in-the-wild* operation where a single ground plane does not exist *e.g.* climbing up stairs or moving over complex terrain. In such scenarios, we require a consistent reference frame, which can be computed from: (i) an accelerometer if a mobile device is used, (ii) pose of static, rigid objects if an object detector is deployed, (iii) fiducial tags or any other means of obtaining a gravity direction.

Note that the ground plane is not an essential piece in the test-time optimization. It is a requirement only because of the way our CVAE is trained: on motions with a ground plane at $z = 0$, gravity in the $-z$ direction, and without complex terrain interactions. Although we empirically noticed that convergence of training necessitates this assumption, other architectures or the availability of larger *in-the-wild* motion datasets might make training HuMoR possible under arbitrary poses. This perspective should clarify why our method *can work* when the ground is invisible: TestOpt might converge from a bad initialization as long as our prior (HuMoR) is able to account for the observation.

On the Assumption of a Static Camera. While a static camera is assumed in all of our evaluations, recent advances in 3D computer vision make it possible to overcome this limitation. Our method, backed by either a structure from motion / SLAM pipeline or a camera relocalization engine, can indeed work in scenarios where the camera moves as well as the human targets. A more sophisticated solution could leverage our learned motion model to disambiguate between camera and human motion. Expectedly, this requires further investigation, making room for future studies as discussed at the end of the main paper.

Other Limitations and Failure Cases. As discussed in

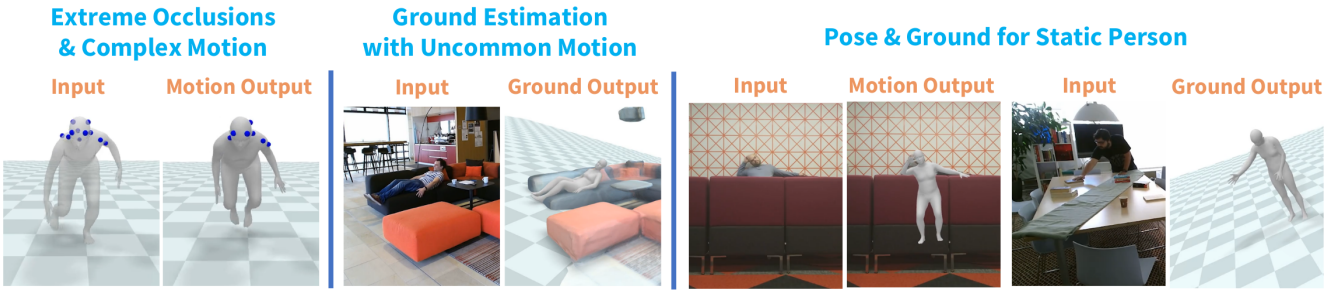

Figure 1: Failure cases of TestOpt using HuMoR. Please see Sec. 2.1 or the supplementary videos for details of each.

Sec. 6 of the main paper, HuMoR has limitations that motivate multiple future directions. First, optimization is generally slow compared to learning-based (direct prediction) methods. This also reflects on our test-time optimization. Approaches for *learning to optimize* can come handy in increasing the efficiency of our method. Additionally, our current formulation of TestOpt allows only for a single output, the local optimum. Therefore, future work may explore learned approaches yielding multi-hypothesis output, which can be used to characterize uncertainty.

Specific failure cases (as shown in the supplementary videos and Fig. 1) further highlight areas of future improvement. First, extreme occlusions (e.g. only a few visible points as in Fig. 1 left), especially at the first frame which determines $\mathbf{x}_0$, makes for a difficult optimization that often lands in local minima with implausible motions. Second, uncommon motions that are rare during CVAE training, such as laying down in Fig. 1 (middle), can cause spurious ground plane outputs as TestOpt attempts to make the motion more likely. Leveraging more holistic scene understanding methods and models of human-environment interaction will help in these cases. Finally, our method is dependent on motion in order to resolve ambiguity, which is usually very helpful but has corner cases as shown in Fig. 1 (right). For example, if the observed person is nearly static, the optimization may produce implausible poses due to ambiguous occlusions (*e.g.* standing when really the person is sitting) and/or incorrect ground plane estimations.

3. HuMoR Model Details

In this section, we provide additional implementation details for the HuMoR motion model described in Sec. 3 of the main paper.

3.1. CVAE Architecture and Implementation

Body Model. We use the SMPL+H body model [31] since it is used by the AMASS [22] dataset. However, our focus is on modeling body motion, so HuMoR and TestOpt do not consider the hand joints (leaving the 22 body joints including the root). Hand joints could be straightforwardly optimized with body motion, but was not in our current scope.

Canonical Coordinate Frame. To ease learning and improve generalization, our network operates on inputs in a canonical coordinate frame. Specifically, based on $\mathbf{x}_{t-1}$ we apply a rotation around the up ($+z$) axis and translation in x, y such that the x and y components of $\mathbf{r}_{t-1}$ are 0 and the person's body right axis (w.r.t. Φ_{t-1}) is facing the $+x$ direction.

Architecture. The encoder and prior networks are identical multi-layer perceptrons (MLP) with 5 layers and hidden size 1024. The decoder is a 4-layer MLP with hidden sizes (1024, 1024, 512). The latent transition $\mathbf{z}_t \in \mathbb{R}^{48}$ is skip-connected to every layer of the decoder in order to emphasize its importance and help avoid posterior collapse [17]. ReLU non-linearities and group normalization [39] with 16 groups are used between all layers except outputs in each network. Input rotations are represented as matrices, while the network outputs the axis-angle representation in $\mathbb{R}^3$. In total, the CVAE network contains $\sim$9.7 million parameters.

3.2. CVAE Training

Losses. The loss function used for training is primarily described in the main paper (see Eq. 7). For a training pair $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, the KL divergence loss term is computed between the output distributions of the encoder and conditional prior as

$$\begin{aligned}\mathcal{L}_{\text{KL}} &= D_{\text{KL}}(q_\phi(\mathbf{z}_t|\mathbf{x}_t, \mathbf{x}_{t-1}) \parallel p_\theta(\mathbf{z}_t|\mathbf{x}_{t-1})) \\ &= D_{\text{KL}}(\mathcal{N}(\mathbf{z}_t; \mu_\phi(\mathbf{x}_t, \mathbf{x}_{t-1}), \sigma_\phi(\mathbf{x}_t, \mathbf{x}_{t-1})) \\ &\qquad ||\mathcal{N}(\mathbf{z}_t; \mu_\theta(\mathbf{x}_{t-1}), \sigma_\theta(\mathbf{x}_{t-1}))). \end{aligned} \tag{1}$$

The SMPL loss $\mathcal{L}_{\text{SMPL}}$ is computed using the ground truth shape parameters β provided in AMASS on the ground truth gendered body model.

Dataset. For training, we use AMASS [22]: a large, publicly-available motion capture (mocap) database containing over 11k motion sequences from 344 different people fit to SMPL. The database aggregates and standardizes many mocap datasets into one. We pre-process AMASS

by cropping the middle 80% of each motion sequence, subsampling to 30 Hz, estimating velocities with finite differences, and using automated heuristics based on foot contacts to remove sequences with substantial terrain interaction (*e.g.* stairs, ramps, or platforms). We automatically annotate ground contacts for 8 body joints (*left and right toes*, *heels*, *knees*, and *hands*) based on velocity and height. In particular, if a joint has moved less than $0.5cm$ in the last timestep and its z component is within $8cm$ of the floor, it is considered to be in contact. For toe joints, we use a tighter height threshold of $4cm$.

For training the CVAE, we use the recommended training split (save for TCD Hands [14] which contains mostly hand motions): CMU [6], MPI Limits [2], TotalCapture [35], Eyes Japan [20], KIT [24], BMLrub [34], BMLmovi [10], EKUT [24], and ACCAD [1]. For validation during training we use MPI HDM05 [27], SFU [37], and MPI MoSh [19]. Finally for evaluations (Sec. 5.3 of the main paper), we use HumanEva [32] and Transitions [22].

Training Procedure. We train using 10-step sequences sampled on-the-fly from the training set (in order to use scheduled sampling as detailed below). To acquire a training sequence, a full mocap sequence is randomly (uniformly) chosen from AMASS and then a random 10-step window within that sequence is (uniformly) sampled. Training is performed using batches of 2000 sequences for 200 epochs with Adamax [15] and settings $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-8}$. We found this to be more stable than using Adam. The learning rate starts at $1e^{-4}$ and decays to $5e^{-5}$, $2.5e^{-5}$, and $1.25e^{-5}$ at epochs 50, 80, and 140, respectively. We use early stopping by choosing the network parameters that result in the best validation split performance throughout training.

A common difficulty in training VAEs is posterior collapse [21] – when the learned latent encoding $\mathbf{z}_t$ is effectively ignored by the decoder. This problem is exacerbated in CVAEs since the decoder receives additional conditioning [17, 33]. To combat collapse, we linearly anneal w_{KL} from 0.0 to its full value of $4e^{-4}$ over the first 50 epochs. We also found that our full model, which uses a learned conditional prior, was less susceptible to posterior collapse than the baselines that assume $p_\theta(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t; \mathbf{0}, \mathbf{I})$.

Training Computational Requirements. We train our CVAE on a single Tesla V100 16GB GPU, which takes approximately 4 days.

Scheduled Sampling. As explained in the main paper, our scheduled sampling follows [17]. In particular, at each training epoch i we define a probability $s_i \in [0.0, 1.0]$ of using the ground truth state input $\mathbf{x}_{t-1}$ at each timestep t in a training sequence, as opposed to the model's own previous output $\hat{\mathbf{x}}_{t-1}$. Training is done using a curriculum that includes $s_i = 1.0$ (regular supervised training), $s_i \in (0.0, 1.0)$ (mix of true and self inputs at each step), and finally $s_i = 0.0$ (always use full generated rollouts). Importantly for training stability, if using the model's own prediction $\hat{\mathbf{x}}_{t-1}$ as input to t, we do not backpropagate gradients from the loss on $\hat{\mathbf{x}}_t$ back through $\hat{\mathbf{x}}_{t-1}$.

For CVAE training, we use 10 epochs of regular supervised training, 10 of mixed true and self inputs, and the rest using full self-rollouts.

3.3. Initial State GMM

State Representation. Since the GMM models a single state, we use a modified representation that is minimal (*i.e.* avoids redundancies) in order to be useful during test-time optimization. In particular the GMM state is

$$\mathbf{x}^{\text{GMM}} = [\quad \dot{\mathbf{r}} \quad \dot{\Phi} \quad \mathbf{J} \quad \dot{\mathbf{J}} \quad] \tag{2}$$

with $\dot{\mathbf{r}}, \dot{\Phi} \in \mathbb{R}^3$ the root linear and angular velocities, and joint positions and velocities $\mathbf{J}, \dot{\mathbf{J}} \in \mathbb{R}^{3\times 22}$. During TestOpt, joints are determined from the current SMPL parameters $\mathbf{J} = \mathbf{J}^{\text{SMPL}} = M(\mathbf{r}, \Phi, \Theta, \beta)$ so that gradients of the GMM log-likelihood (Eq. 11 in the main paper) will be used to update the initial state SMPL parameters.

Implementation Details. The GMM uses full covariance matrices for each of the 12 components and operates in the same canonical coordinate frame as the CVAE. It trains using expectation maximization[4] on every state in the same AMASS training set used for the CVAE.

4. Test-Time Optimization Details

In this section, we give additional details of the motion and shape optimization detailed in Sec. 4 of the main paper.

State Representation. In practice, for optimizing $\mathbf{x}_0$ we slightly modify the state from Eq. 1 in the main paper. First, we remove the joint positions $\mathbf{J}$ to avoid the previously discussed redundancy, which is good for training the CVAE but bad for test-time optimization. Instead, we use $\mathbf{J}^{\text{SMPL}}$ whenever needed at t_0. Second, we represent body pose Θ in the latent space of the VPoser [29] pose prior with $\mathbf{z}_0^{\text{pose}} \in \mathbb{R}^{32}$. Whenever needed, we can map between full joint angles and latent pose using the VPoser encoder and decoder. Finally, in our implementation state variables $\mathbf{x}_t$ are by default represented in the coordinate frame of the given observations, *e.g.* relative to the camera, to allow easily fitting to data - they are transformed back into the canonical CVAE frame when necessary as discussed below.

Floor Parameterization. As detailed in the main paper, to obtain the transformation between the canonical coordinate frame in which our CVAE is trained and the observation frame used for optimization, we additionally optimize the

[4] using scikit-learn

floor plane of the scene $\mathbf{g} \in \mathbb{R}^3$. This parameterization is $\mathbf{g} = d\hat{\mathbf{n}}$ where $\hat{\mathbf{n}}$ is the ground unit normal vector and d the plane offset. To disambiguate the normal vector direction $\hat{\mathbf{n}}$ given $\mathbf{g}$, we assume that the y-component of the normal vector must be negative, *i.e.* it points upward in the camera coordinate frame. This assumes the camera is not severely tilted such that the observed scene is "upside down".

Observation-to-Canonical Transformation. We assume that gravity is orthogonal to the ground plane. Therefore, given the current floor $\mathbf{g}$ and root state $\mathbf{r}, \Phi$ (in the observation frame) we compute a rotation and translation to the canonical CVAE frame: after the transformation, $\hat{\mathbf{n}}$ is aligned with $+z$ and $d = 0$, Φ faces body right towards $+x$, and the x, y components of $\mathbf{r}$ are 0. With this ability, we can always compute the (observed) state at time $\mathbf{x}_t$ from $\mathbf{z}_{1:t}$, $\mathbf{x}_0$, and $\mathbf{g}$ by (i) transforming $\mathbf{x}_0$ to the canonical frame, (ii) using the CVAE to rollout $\mathbf{x}_t = f(\mathbf{x}_0, \mathbf{z}_{1:t})$, and (iii) transforming $\mathbf{x}_t$ back to the observation frame.

Optimization Objective Details. The optimization objective is detailed in Sec. 4.2 of the main paper. To compensate for the heavy tailed behavior of real data, we use robust losses for multiple data terms. $\mathcal{E}_{\text{data}}^{\text{2D}}$ uses the Geman-McClure function [9] which for our purposes is defined as $\rho(r, \sigma) = (\sigma^2 r^2)/(\sigma^2 + r^2)$ for a residual r and scaling factor σ. We use $\sigma = 100$ for all experiments. $\mathcal{E}_{\text{data}}^{\text{PC3D}}$ uses robust bisquare weights [4]. These weights are computed based on the one-way chamfer distance term (see Eq. 14 in the main paper): residuals over the whole sequence are first normalized using a robust estimate of the standard deviation based on the median absolute deviation (MAD), then each weight is computed as

$$w_{\text{bs}} = \begin{cases} (1 - (\hat{r}/\kappa)^2)^2 & |\hat{r}/\kappa| < 1 \\ 0.0 & \text{else} \end{cases} \tag{3}$$

In this equation, $\hat{r}$ is a normalized residual and κ is a tuning constant which we set to 4.6851.

In the $\mathcal{E}_{\text{env}}$ energy term, we use $\delta = 8\ cm$ to ensure the z-component of contacting joints are within $8\ cm$ of the floor when in contact (since joints are inside the body) in the canonical frame.

Initialization. As detailed in Sec. 4.2 of the main paper, our optimization is initialized by directly optimizing SMPL pose and shape parameters using $\mathcal{E}_{\text{data}}$ and $\mathcal{E}_{\text{shape}}$ along with a pose prior $\mathcal{E}_{\text{pose}}$ and joint smoothing $\mathcal{E}_{\text{smooth}}$. The latter are weighted by λ_{pose} and λ_{smooth}. This two-stage initialization first optimizes global translation and orientation for 30 optimization steps, followed by full pose and shape for 80 steps. At termination, we estimate velocities using finite differences, which allows direct initialization of the state $\mathbf{x}_0^{\text{init}}$. To get $\mathbf{z}_{1:T}^{\text{init}}$, the CVAE encoder is used to infer the latent transition between every pair of frames. The initial shape parameters β_{init} are a direct output of the initialization optimization. Finally, for fitting to RGB(-D) the ground plane is initialized from video with PlaneRCNN [18], though we found simply setting the floor to $y = 0$ (*i.e.* the normal is aligned with the camera up axis) works just as well in most cases.

Optimization (TestOpt) Details. Our optimization is implemented in PyTorch [28] using L-BFGS with a step size of 1.0 and *autograd*. For all experiments, we optimize using the neutral SMPL+H [31] body model in 3 stages. First, only the initial state $\mathbf{x}_0$ and first 15 frames of the latent sequence $\mathbf{z}_{1:15}$ are optimized for 30 iterations in order to quickly reach a reasonable initial state. Next, $\mathbf{x}_0$ is fixed while the full latent dynamics sequence $\mathbf{z}_{1:T}$ is optimized for 25 iterations, and then finally the full sequence and initial state are tuned together for another 15 iterations. The ground $\mathbf{g}$ and shape β are optimized in every stage.

The energy weights used for each experiment in the main paper are detailed in Tab. 1. The left part of the table indicates weights for the initialization phase (*i.e.* the *VPoser-t* baseline), while the right part is our full proposed optimization. A dash indicates the energy is not relevant for that data modality and therefore not used. Weights were manually tuned using the presented evaluation metrics and qualitative assessment. Note that for similar modalities (*e.g.* 3D joints and keypoints, or RGB and RGB-D) weights are quite similar and so only slight tuning should be necessary to transfer to new data. The main tradeoff comes between reconstruction accuracy and motion plausibility: *e.g.* the motion prior is weighted higher for i3DB, which contains many severe occlusions, than for PROX RGB where the person is often nearly fully visible.

5. MAP Objective Derivation

In this section, we formulate the core of the pose and shape optimization objective (Eq. 10 in the main paper) from a probabilistic perspective. Recall, we want to optimize the initial state $\mathbf{x}_0$, a sequence of latent variables $\mathbf{z}_{1:T}$, ground $\mathbf{g}$, and shape β based on a sequence of observations $\mathbf{y}_{0:T}$. We are interested in the maximum a-posteriori (MAP) estimate:

$$\max_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} p(\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta | \mathbf{y}_{0:T}) \tag{4}$$

$$= \max_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} p(\mathbf{y}_{0:T} | \mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta) p(\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta) \tag{5}$$

Assuming $\mathbf{y}_0$ is independent of $\mathbf{g}$, the left term is written

$$p(\mathbf{y}_0 | \mathbf{x}_0, \beta) \prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{z}_{\leq t}, \mathbf{x}_0, \mathbf{g}, \beta) = \prod_{t=0}^{T} p(\mathbf{y}_t | \mathbf{x}_t, \beta) \tag{6}$$

where $\mathbf{y}_t$ is assumed to only be dependent on the initial state and *past* transitions. Additionally, $\{\mathbf{z}_{\leq t}, \mathbf{x}_0, \mathbf{g}\}$ is replaced

Dataset	Initialization						Full Optimization										
	$\lambda_{\text{data}}^{\text{3D}}$	$\lambda_{\text{data}}^{\text{2D}}$	$\lambda_{\text{data}}^{\text{PC3D}}$	λ_{shape}	λ_{pose}	λ_{smooth}	$\lambda_{\text{data}}^{\text{3D}}$	$\lambda_{\text{data}}^{\text{2D}}$	$\lambda_{\text{data}}^{\text{PC3D}}$	λ_{shape}	λ_{CVAE}	λ_{init}	λ_{c}	λ_{b}	λ_{cv}	λ_{ch}	λ_{gnd}
AMASS (occ keypoints)	1.0	-	-	0.015	$2e^{-4}$	0.1	1.0	-	-	0.015	$5e^{-4}$	$5e^{-4}$	1.0	10.0	1.0	1.0	-
AMASS (noisy joints)	1.0	-	-	0.015	$2e^{-4}$	10.0	1.0	-	-	0.015	$1e^{-3}$	$1e^{-3}$	1.0	10.0	1.0	1.0	-
i3DB (RGB)	-	$1e^{-3}$	-	4.5	0.04	100.0	-	$1e^{-3}$	-	4.5	0.075	0.075	100.0	$2e^{3}$	0.0	10.0	15.0
PROX (RGB)	-	$1e^{-3}$	-	4.5	0.04	100.0	-	$1e^{-3}$	-	4.5	0.05	0.05	100.0	$2e^{3}$	100.0	10.0	15.0
PROX (RGB-D)	-	$1e^{-3}$	1.0	3.0	0.1	100.0	-	$1e^{-3}$	1.0	3.0	0.075	0.075	100.0	$2e^{3}$	100.0	10.0	90.0

Table 1: Energy weightings used in test-time optimization for each experiment in Sec. 5 of the main paper.

with $\mathbf{x}_t = f(\mathbf{x}_0, \mathbf{z}_{1:t})$ using CVAE rollout as detailed previously. The right term in Eq. (5) is written as

$$p(\mathbf{x}_0, \mathbf{g}, \beta) \prod_{t=1}^{T} p(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_0, \mathbf{g}, \beta) \tag{7}$$

$$= p(\mathbf{x}_0|\mathbf{g}) p(\mathbf{g}) p(\beta) \prod_{t=1}^{T} p(\mathbf{z}_t | \mathbf{x}_{t-1}) \tag{8}$$

where $\mathbf{x}_0$, $\mathbf{z}_t$, and $\mathbf{g}$ are assumed to be independent of β. We then use these results within Eq. (5) to optimize the log-likelihood:

$$\begin{aligned}
&\max_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} \log p(\mathbf{y}_{0:T} | \mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta) + \log p(\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta) \\
&= \min_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} - \sum_{t=0}^{T} \log p(\mathbf{y}_t | \mathbf{x}_t, \beta) - \sum_{t=1}^{T} \log p(\mathbf{z}_t | \mathbf{x}_{t-1}) \\
&\qquad - \log p(\mathbf{x}_0|\mathbf{g}) - \log p(\mathbf{g}) - \log p(\beta) \\
&= \min_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} \mathcal{E}_{\text{data}} + \mathcal{E}_{\text{CVAE}} + \mathcal{E}_{\text{init}} + \mathcal{E}_{\text{gnd}} + \mathcal{E}_{\text{shape}} \qquad (9) \\
&= \min_{\mathbf{x}_0, \mathbf{z}_{1:T}, \mathbf{g}, \beta} \mathcal{E}_{\text{mot}} + \mathcal{E}_{\text{data}} + \mathcal{E}_{\text{gnd}} + \mathcal{E}_{\text{shape}}. \qquad (10)
\end{aligned}$$

Assuming each energy presented in the main paper can be written as the log-likelihood of a distribution, this formulation recovers our optimization objective besides the additional regularizers $\mathcal{E}_{\text{skel}}$ and $\mathcal{E}_{\text{env}}$ (these terms could, in principle, be written as part of a more complex motion prior term $\mathcal{E}_{\text{mot}}$, however for simplicity we do not do this). Next, we connect each energy term as presented in Sec. 4.2 of the paper to the probabilistic perspective.

Motion Prior $\mathcal{E}_{\text{mot}}$. This term is already the log-likelihood of our HuMoR motion model (Eq. 11 of the paper), which exactly aligns with the MAP derivation.

Data Term $\mathcal{E}_{\text{data}}$. The form of $p(\mathbf{y}_t|\mathbf{x}_t, \beta)$ is modality-dependent. In the simplest case the observations $\mathbf{y}_t$ are 3D joints (or keypoints with known correspondences) and $p(\mathbf{y}_t|\mathbf{x}_t, \beta)$ is defined by $\mathbf{y}_t = \mathbf{J}_t^{\text{SMPL}} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_{\text{data}})$. Then the energy is as written in Eq. 12 of the paper. For other modalities (Eq. 13 and 14 in the paper), the data term can be seen as resulting from a more sophisticated noise model.

Ground Prior $\mathcal{E}_{\text{gnd}}$. We assume the ground should stay close to initialization so $p(\mathbf{g}) = \mathcal{N}(\mathbf{g}; \mathbf{g}^{\text{init}}, \sigma_{\text{gnd}})$ corresponding to the objective in the paper $\mathcal{E}_{\text{gnd}} = \lambda_{\text{gnd}} ||\mathbf{g} - \mathbf{g}^{\text{init}}||^2$.

Shape Prior $\mathcal{E}_{\text{shape}}$. The shape β should stay near neutral zero and so $p(\beta) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ which gives the energy $\mathcal{E}_{\text{shape}} = \lambda_{\text{shape}} ||\beta||^2$.

6. Experimental Evaluation Details

In this section, we provide details of the experimental evaluations in Sec. 5 of the main paper.

6.1. Datasets

AMASS [22] We use the same processed AMASS dataset as described in Sec. 3.2 for experiments. Experiments in Sec. 5.3 and 5.4 of the main paper use the held out Transitions and HumanEva [32] subsets which together contain 4 subjects and about 19 minutes of motion.

i3DB [26] is a dataset of RGB videos captured at 30 Hz containing numerous person-environment interactions involving medium to heavy occlusions. It contains annotated 3D joint positions at 10 Hz along with a primitive cuboid 3D scene reconstruction. We run off-the-shelf 2D pose estimation (OpenPose) [5], person segmentation [7], and plane detection [18] models to obtain inputs and initialization for our test-time optimization. We evaluate our method in Sec. 5.5 of the main paper on 6 scenes (scenes 5, 7, 10, 11, 13, and 14) containing 2 people which totals about 1800 evaluation frames. From the annotated 3D objects, we fit a ground plane which is used to compute plausibility metrics.

PROX [12] is a large-scale dataset of RGB-D videos captured at 30 Hz containing person-scene interactions in a variety of environments with light to medium occlusions. We use a subset of the qualitative part of the dataset to evaluate the plausibility of our method's estimations. The data does *not* have pose annotations, but does contain the scanned scene mesh to which we fit a ground plane for plausibility evaluation. We obtain 2D pose, person masks, and ground plane initialization in the same way as for i3DB. We evaluate in Sec. 5.5 of the main paper on all videos from 4 chosen scenes (N3Office, N3Library, N0Sofa, and MPH1Library) that tend to have more dynamic motions and occlusions. In total, these scenes contain 12 unique people and about 19 minutes of video.

6.2. Baselines and Evaluation Metrics

Motion Prior Baselines. To be usable in our whole framework (*e.g.* test-time optimization with SMPL), the *MVAE*

baseline is our proposed CVAE with all ablations applied simultaneously (no delta step prediction, no contact prediction, no SMPL losses, and no learned conditional prior). Note that this differs slightly from the model as presented in [17]: the decoder is an MLP rather than a mixture-of-experts and the layer sizes are larger to provide the necessary representational capacity for training on AMASS. All ablations and *MVAE* are trained in the exact same way as the full model. Additionally, when used in test-time optimization we use the same energy weightings as described in Tab. 1 but with irrelevant energies removed (*e.g.* the *No Contacts* ablation does not allow the use of $\mathcal{E}_{\text{env}}$). Note that $\mathcal{E}_{\text{init}}$ *is still used* with *MVAE* and all ablations, the only thing that changes is the prior in $\mathcal{E}_{\text{CVAE}}$.

Motion Estimation Baselines. The *VPoser-t* baseline is exactly the initialization phase of our proposed test-time optimization, *i.e.* we use weightings in Tab. 1.

The *PROX-RGB* baseline fits the neutral SMPL-X [29] body model to the same 2D OpenPose detections used by our method. It does not use the face or hand keypoints for fitting similar to our approach. The *PROX-D* baseline uses the fittings provided with the PROX dataset, which are on the known gendered SMPL-X body model and use face/hand 2D keypoints for fitting.

The *VIBE* baseline uses the same 2D OpenPose detections as our method in order to define bounding boxes for inference. We found this makes for a more fair comparison since the real-time trackers used in their implementation[5] often fail for medium to heavy occlusions common in our evaluation datasets.

Evaluation Metrics. In order to report occluded (**Occ**) and visible (**Vis**) positional errors separately, we must determine which joints/keypoints are occluded during evaluation. This is easily done for 3D tasks where "occlusions" are synthetically generated. For RGB data in i3DB, we use the person segmentation mask obtained with DeepLabv3 [7] to determine if a ground truth 3D joint is visible after projecting it to the camera.

For a joint $\mathbf{p}_t \in \mathbf{J}_t^{\text{SMPL}}$ at time t the acceleration magnitude (**Accel**) is computed as

$$a = ||(\mathbf{p}_{t-1} - 2\mathbf{p}_t + \mathbf{p}_{t+1})/h^2|| \tag{11}$$

where $h = 1/30$ for all datasets. Ground penetration frequency (**Freq**) for a given penetration threshold g_{thresh} is computed over all D frames in a dataset as

$$\frac{\sum_D \mathbf{1}(d_{\text{pen}}^{\text{ltoe}} > g_{\text{thresh}}) + \mathbf{1}(d_{\text{pen}}^{\text{rtoe}} > g_{\text{thresh}})}{2D} \tag{12}$$

where $\mathbf{1}(\cdot)$ is the indicator function and $d_{\text{pen}}^{\text{ltoe}}, d_{\text{pen}}^{\text{rtoe}}$ are the penetration distances (shortest distance to the ground plane) for the left and right toe joints at the current frame.

[5] see Github

6.3. Estimation from 3D Observations

For fitting to 3D data, as presented in Sec. 5.4 of the main paper, the observation and canonical coordinate frames are identical since AMASS data is used, therefore TestOpt does not optimize the ground plane $\mathbf{g}$.

6.4. Estimation from RGB(-D) Observations

i3DB. Positional joint errors are computed using a 12-joint subset of the ground truth 3D joint annotations which correspond to the SMPL joints used by our method and all baselines. These include ankles, knees, wrists, elbows, shoulders, the neck, and the root. The leg joints reported in Tab. 3 of the main paper include the ankles and knees.

In fitting to i3DB videos, we mask 2D joint observations using the person segmentation mask which we found beneficial under the numerous severe occlusions where OpenPose may predict spurious 2D pose with incorrectly high confidence.

PROX. For fitting to PROX RGB(-D) videos we found it best *not* to mask out 2D joints based on the person segmentation mask, as occlusions are typically minor and OpenPose is relatively accurate. However, the segmentation is always used on the point cloud back-projected from the depth map to ignore points far from the person.

7. Extended Evaluations

In this section, we present experimental evaluations to supplement those in Sec. 5 of the main paper, which were ommitted due to space constraints.

7.1. Qualitative Evaluation

Please see the **supplementary videos** for extensive qualitative results corresponding to each experiment in Sec. 5 of the main paper. In this document, we show various representative examples from these videos and summarize important results.

Fig. 4 shows results fitting to occluded 3D keypoints (Sec. 5.4 of the main paper). Performance of TestOpt with HuMoR is compared to the *VPoser-t* and *MVAE* baselines on two sequences. *VPoser-t* fails to produce any plausible lower-body motion since it uses only a pose prior, while using *MVAE* as the motion prior often gives unnatural and implausible motions that don't align well with the observed keypoints.

Fig. 5 shows results using TestOpt with HuMoR for fitting to noisy 3D joints (Sec. 5.4 of the main paper). Both the estimated motion and contacts are shown for a crawling sequence. Note that HuMoR recovers complex contact patterns involving not only the feet, but also hands and knees.

Fig. 6 demonstrates fitting performance on RGB videos from PROX compared to the *PROX-RGB* baseline (Sec. 5.5

Method	Global Joint Error				Root-Aligned Joint Error				Ground Pen		
	Vis	**Occ**	**All**	**Legs**	**Vis**	**Occ**	**All**	**Legs**	**Accel**	**Freq**	**Dist**
No $\mathcal{E}_{\text{init}}$	28.04	39.37	30.96	34.91	12.11	22.27	14.73	21.40	2.57	2.80%	0.85
No $\mathcal{E}_{\text{c}}$	29.17	40.62	32.12	35.83	12.99	24.34	15.92	22.66	2.75	**1.98%**	0.67
No $\mathcal{E}_{\text{b}}$	28.29	39.77	31.25	34.37	13.00	24.36	15.93	22.31	2.93	4.43%	1.13
No $\mathcal{E}_{\text{skel}}$	32.87	44.16	35.78	37.55	14.67	26.82	17.80	24.18	3.81	4.35%	1.29
No $\mathcal{E}_{\text{env}}$	26.83	37.62	29.61	32.91	12.09	22.04	14.66	20.96	2.51	2.10%	**0.66**
Full Energy	**26.00**	**34.36**	**28.15**	**31.26**	**12.02**	**21.70**	**14.51**	**20.74**	**2.43**	2.12%	0.68

Table 2: Motion and shape from RGB video (*i.e.* 2D joints) on i3DB [26]. Joint errors are in cm and acceleration is m/s^2. Results use TestOpt with HuMoR. The top part shows various energy terms ablated.

of the main paper). *PROX-RGB* produces temporally incoherent results since it operates on single frames. However, it also uses the scene mesh as input which allows for plausible poses when the person is fully visible. This does not greatly improve results under occlusions, though, often reverting to a mean leg pose similar to *VPoser-t* and *VIBE*.

Fig. 7 demonstrates fitting to RGB-D videos from PROX compared to the *PROX-D* baseline (Sec. 5.5 of the main paper). Using motion as a prior allows for natural interaction within the scene, as detailed in the figure caption.

Fig. 8 shows ground plane estimations when fitting to RGB-D data for each of the scenes in our PROX dataset. The estimated floor is rendered within the true scene mesh for reference.

Finally, we evaluate TestOpt with HuMoR on highly dynamic dancing data to demonstrate the generalization ability of the CVAE motion model. Fig. 9 shows a sample of frames from motions fit to the DanceDB [3] subset of AMASS [23]. In this case, the observations are full-body 3D keypoints. Though HuMoR is trained on data with few dancing motions, it is able to capture these difficult motions at test time since it only operates on pairs of frames. Additionally, Fig. 2 shows fitting results on RGB videos from the AIST dance dataset [36]. Since HuMoR allows for large accelerations, it accurately generalizes to fast motions (top - note motion blur). Moreover, it is able to recover from poor 2D joint detections from OpenPose due to the cartwheel motion (bottom).

7.2. Optimization Objective Ablation

In this experiment, we analyze the effect of the energy terms and regularizers in our test-time optimization (TestOpt) formulation. Tab. 2 reports results on the i3DB [26] dataset using TestOpt with HuMoR for different energy ablations.

No $\mathcal{E}_{init}$ does not use the initial state Gaussian mixture model (GMM) as part of the motion prior (*i.e.* assumes a uniform prior over the initial state). This means the input $\mathbf{x}_0$ to CVAE rollout may not be plausible, especially early in optimization, leading to degraded performance. *No* $\mathcal{E}_c$ and *No* $\mathcal{E}_b$ remove individual terms of the skeleton regularization: the joint and bone length consistency. *No* $\mathcal{E}_{skel}$ removes the entire skeleton regularization (both $\mathcal{E}_{\text{c}}$ and $\mathcal{E}_{\text{b}}$), severely affecting final performance. This term is important to ensuring the CVAE is actually rolling out realistic motions. Finally, *No* $\mathcal{E}_{env}$ removes the contact velocity and height terms, increasing errors particularly for occluded joints while resulting in similar plausibility metrics.

Dataset	Batch Size	Mean Seq Time
AMASS (occ keypoints)	12	2.95
AMASS (noisy joints)	12	2.45
i3DB (RGB)	6	6.42
PROX (RGB)	6	4.48
PROX (RGB-D)	6	4.85

Table 3: Mean per-sequence optimization times (in minutes) for evaluations on each dataset. Optimizations are done on batches of $3s$ (90 frame) sequences.

7.3. Sensitivity to Occlusions and Noise

Next, we look at performance on estimation from 3D data under increasing levels of occlusions and noise. Similar to Sec. 5.4 in the main paper, we consider fitting to occluded 3D keypoints (points under a given height threshold are unobserved) and 3D joint locations with added Gaussian noise. For this experiment, we use the held out Transitions subset of AMASS [22].

Mean keypoint errors for **all** and **occluded** points are shown for increasing occlusions in Fig. 3(a)(b). From left to right the occluded height threshold is 0.0, 0.3, 0.6, 0.9, and 1.2 m which roughly corresponds to the lower body being occluded from the floor up through the body parts on the x-axis. With no occlusions (*None*) *VPoser-t* most closely fits the clean points, while HuMoR outperforms *MVAE* due to improved expressiveness. As occlusions increase, HuMoR and *MVAE* perform similarly while occluded errors increase greatly for *VPoser-t* as also observed in other experiments. Note that after the knees become occluded, errors for occluded keypoints tend to saturate as performance is depen-

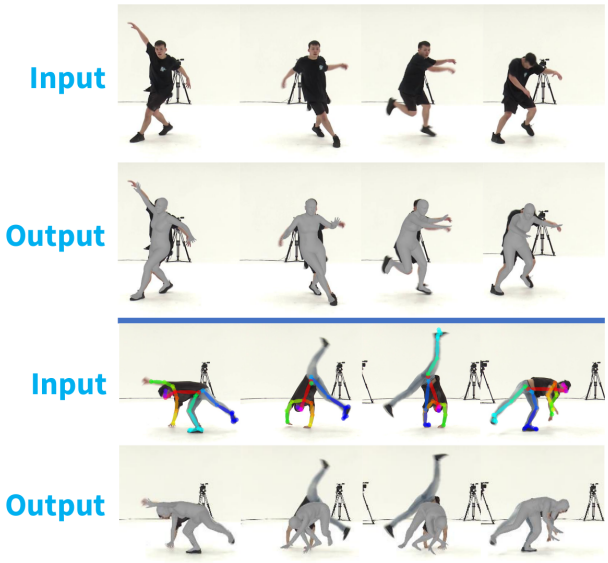

Figure 2: Example sequences using TestOpt with HuMoR to fit to 2D joints in AIST dance videos [36]. HuMoR generalizes to these highly dynamic motions and robustly recovers from inaccurate 2D joint detections (bottom).

dent nearly entirely on the motion prior.

Mean joint acceleration magnitude is shown for increasingly noisy 3D joint observations in Fig. 3(c). From left to right noise increases to 8 cm standard deviation. Importantly, the performance of HuMoR stays relatively stable. HuMoR increases only 25% while *VPoser-t* and *MVAE* increase 124% and 57%, respectively.

7.4. Computational Requirements of TestOpt

For all experiments presented in the main paper, we perform optimization on batches of $3s$ sequences (90 frames). Tab. 3 shows the mean per-sequence optimization times for each of these experiments where *AMASS* corresponds to experiments in Sec. 5.4 of the paper and *i3DB* and *PROX* correspond to Sec. 5.5. The per-sequence time is computed by taking the total time to optimize over the whole dataset divided by the number of $3s$ sequences. Note that batching speeds up this large scale optimization significantly, *i.e.* optimizing a single sequence will only be slightly faster than a batch of sequences since the primary bottleneck is CVAE rollout. All batched optimizations were performed on a 24 GB Titan RTX GPU, though an 8 GB GPU is sufficient to optimize a single sequence.

References

[1] Advanced Computing Center for the Arts and Design. ACCAD MoCap Dataset. 4

[2] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, June 2015. 4

[3] Andreas Aristidou, Ariel Shamir, and Yiorgos Chrysanthou. Digital dance ethnography: Organizing large dance collections. *J. Comput. Cult. Herit.*, 12(4), Nov. 2019. 8, 13

[4] Albert E Beaton and John W Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974. 5

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6

[6] Carnegie Mellon University. CMU MoCap Dataset. 4

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6, 7

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583, 2018. 1

[9] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987. 5

[10] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. MoVi: A large multipurpose motion and video dataset, 2020. 4

[11] Saeed Ghorbani, Calden Wloka, Ali Etemad, Marcus A Brubaker, and Nikolaus F Troje. Probabilistic character motion synthesis using a hierarchical deep latent variable model. In *Computer Graphics Forum*, volume 39. Wiley Online Library, 2020. 1

[12] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, pages 2282–2292, 2019. 6

[13] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. 2

[14] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O'Sullivan. Sleight of hand: Perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, page 79–86, 2012. 4

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 4

[16] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[17] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, volume 39. ACM, 2020. 1, 3, 4, 7

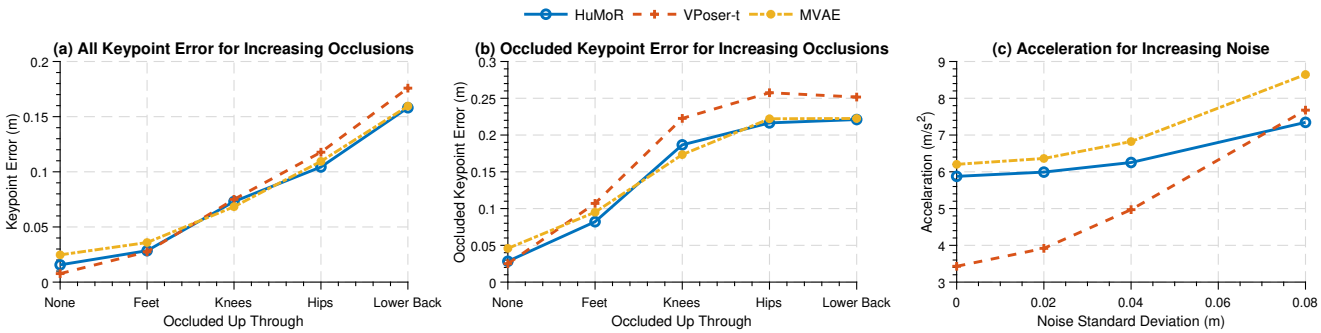

Figure 3: Keypoint errors for **all** (**a**) and **occluded** (**b**) points for increasing levels of occlusion when fitting to 3D keypoints with TestOpt. (**c**) Joint acceleration magnitude for increasing levels of noise when fitting to 3D joints with TestOpt.

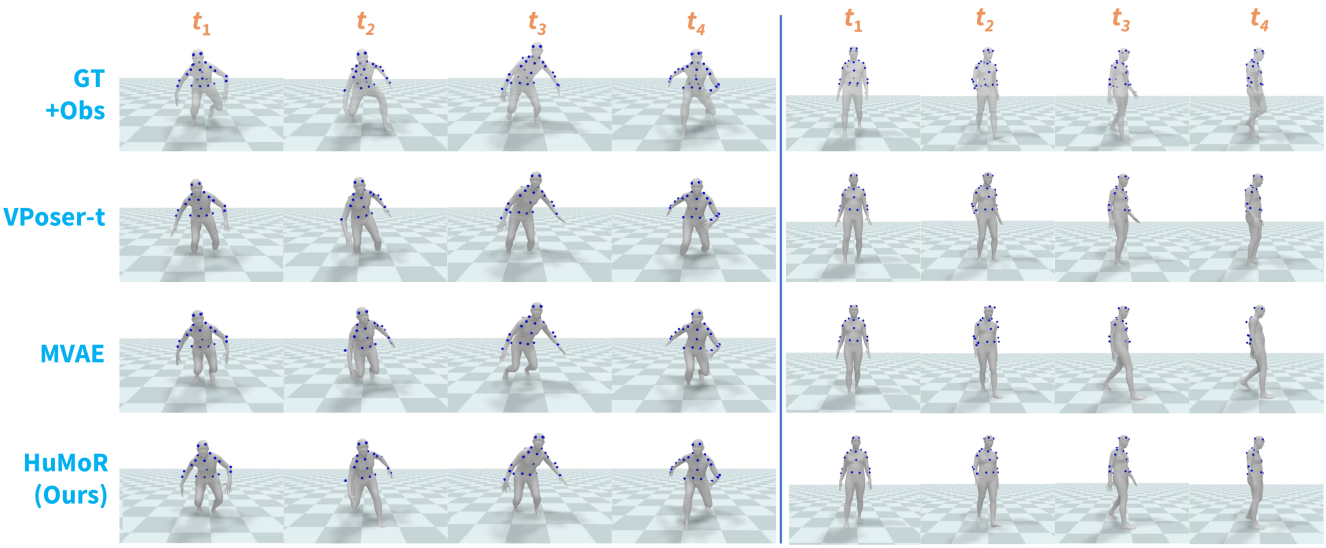

Figure 4: Comparison to baselines when fitting to 3D keypoints from held out sequences in the AMASS dataset. *GT+Obs* shows the ground truth body motion and observed keypoints in blue, while each method output shows the predicted motion with the observed keypoints for reference.

[18] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 5, 6

[19] Matthew Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Trans. Graph.*, 33(6), Nov. 2014. 4

[20] Eyes JAPAN Co. Ltd. Eyes Japan MoCap Dataset. 4

[21] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 4

[22] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, 2019. 1, 2, 3, 4, 6, 8

[23] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5441–5450, Oct. 2019. 8, 13

[24] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour. The KIT whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 329–336, July 2015. 4

[25] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[26] Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J. Mitra. iMapper: Interaction-guided scene mapping from monocular videos. *ACM SIGGRAPH*, 2019. 6, 8

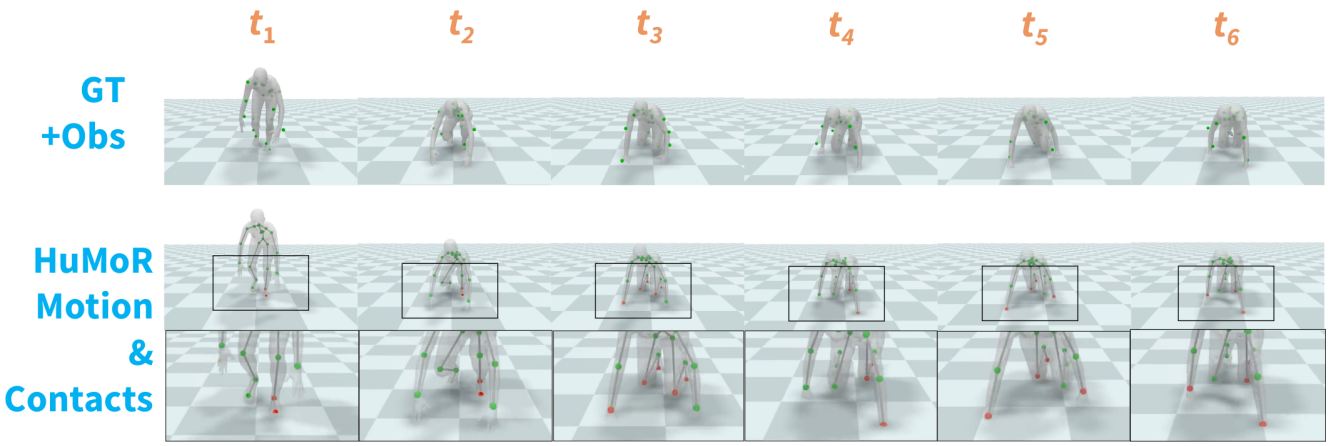

Figure 5: TestOpt with HuMoR recovers complex contact patterns involving feet, knees, and hands from noisy 3D joint observations (shown on top along with the ground truth motion).

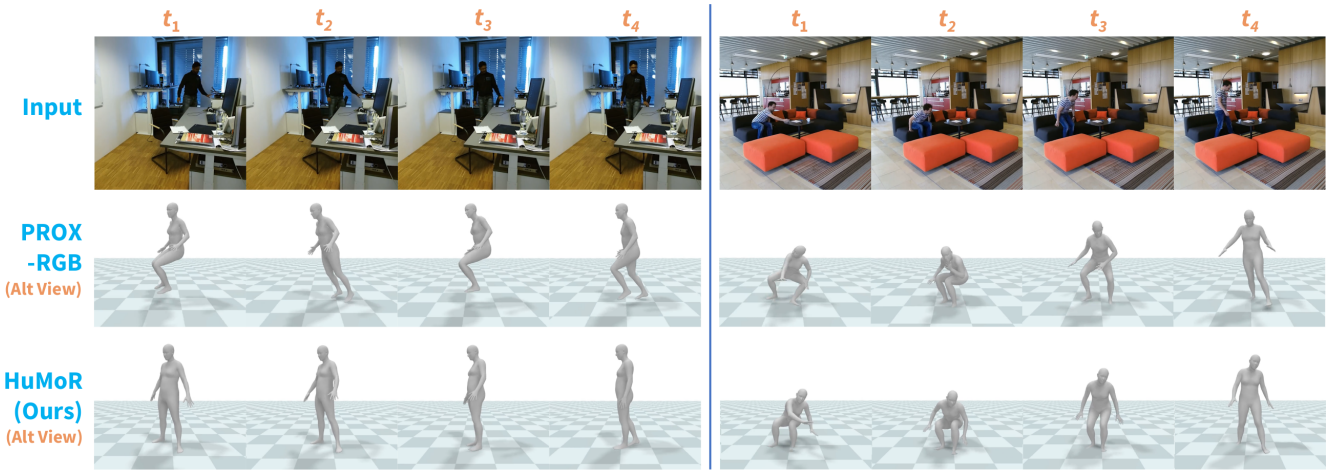

Figure 6: Comparison to *PROX-RGB* on videos from the PROX dataset. Predicted motion is shown from an alternate viewpoint for both methods. In both sequences, *PROX-RGB* is temporally inconsistent. In the left example, the lower body occlusion causes implausible neutral standing or sitting poses. The predicted ground plane from HuMoR is shown for reference.

[27] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database HDM05. Technical Report CG-2007-2, Universität Bonn, 2007. 4

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems*, 2017. 5

[29] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 4, 7

[30] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[31] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 3, 5

[32] 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. 4, 6

[33] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 3483–3491.

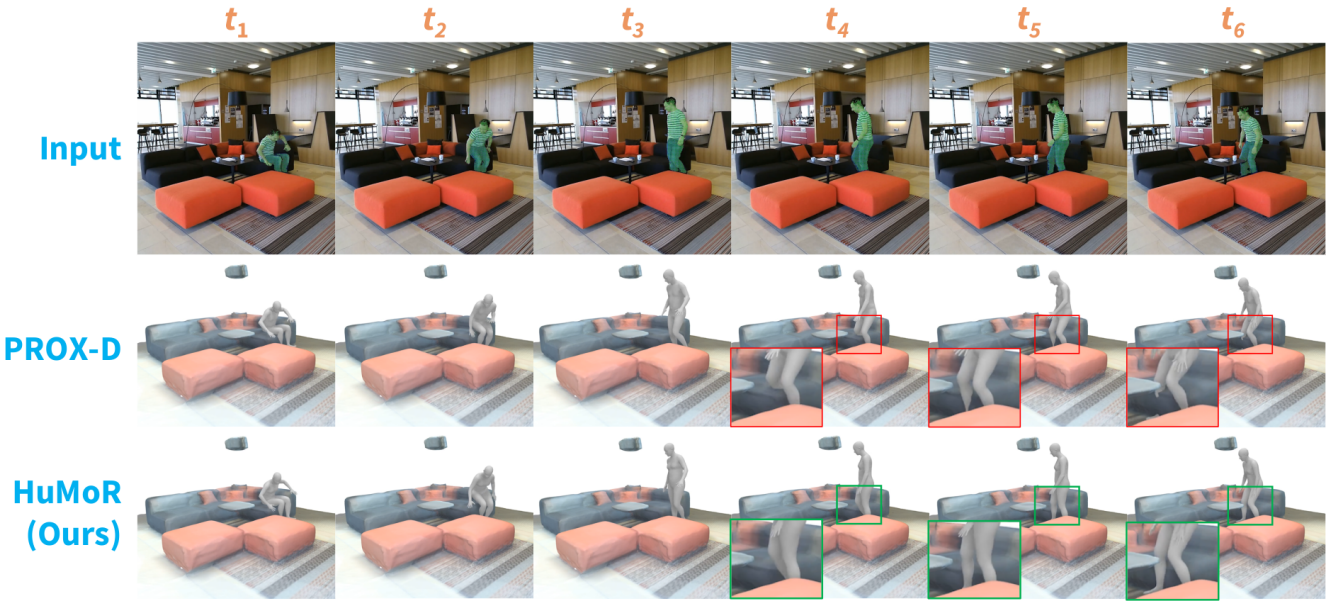

Figure 7: Comparison to *PROX-D* on PROX RGB-D data. Motion and shape results are rendered within the ground truth scene mesh for reference. Though our method does not use the scene mesh as input like *PROX-D*, it still produces motions that are plausible within the environment by using HuMoR as the motion prior, sometimes better than *PROX-D* as indicated by the red and green boxes here.

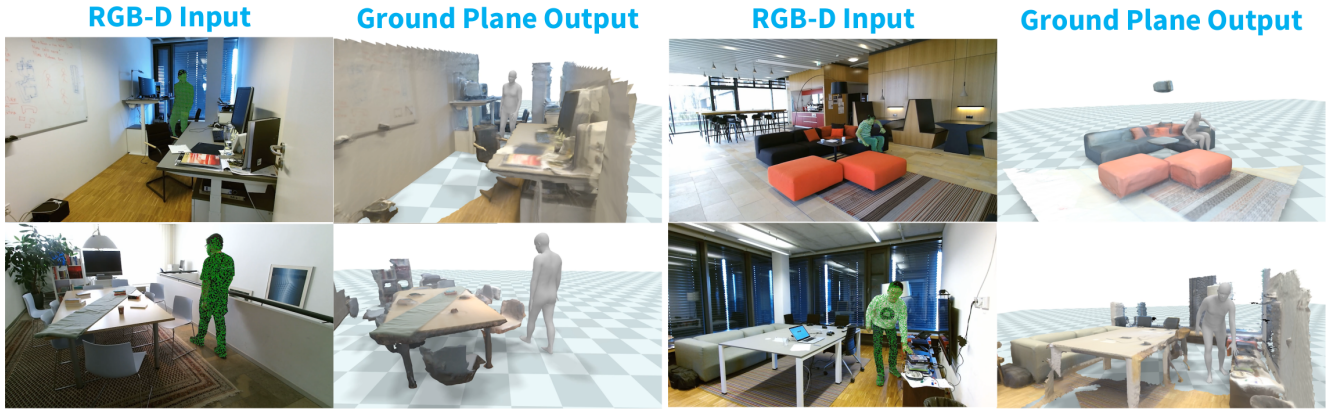

Figure 8: Ground estimation examples from each scene in PROX from RGB-D. The scene mesh is shown for reference only, it is not an input or output of TestOpt with HuMoR.

Curran Associates, Inc., 2015. 4

[34] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2–2, Sept. 2002. 4

[35] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of the British Machine Vision Conference (BMVC)*, Sept. 2017. 4

[36] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, pages 501–510, Delft, Netherlands, Nov. 2019. 8, 9

[37] Simon Fraser University and National University of Singapore. SFU Motion Capture Database. 4

[38] Jianqiao Wangni, Dahua Lin, Ji Liu, Kostas Daniilidis, and Jianbo Shi. Towards statistically provable geometric 3d human pose recovery. *SIAM Journal on Imaging Sciences*, 14(1):246–270, 2021. 2

[39] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[40] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceed-*

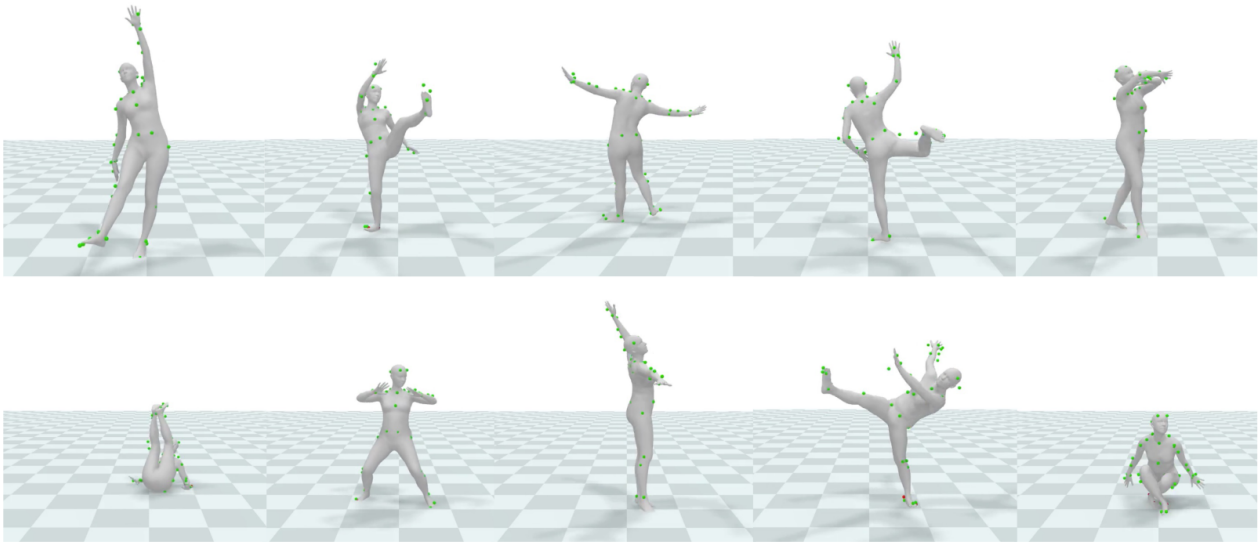

Figure 9: Example frames from using TestOpt with HuMoR to fit to full-body 3D keypoint motions from dynamic DanceDB [3] data in AMASS [23]. Estimated body shape and pose is shown along with ground truth keypoints in green. Despite not training on this dance data, HuMoR is able to effectively generalize to capture these complex motions.

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3372–3382, 2021. 1