GLAMR: Global Occlusion-Aware Human Mesh Recovery with Dynamic Cameras

Ye Yuan\textsuperscript{2}  
Umar Iqbal\textsuperscript{1}  
Pavlo Molchanov\textsuperscript{1}  
Kris Kitani\textsuperscript{2}  
Jan Kautz\textsuperscript{1}

\textsuperscript{1}NVIDIA  
\textsuperscript{2}Carnegie Mellon University

https://nvlabs.github.io/GLAMR

Abstract

We present an approach for 3D global human mesh recovery from monocular videos recorded with dynamic cameras. Our approach is robust to severe and long-term occlusions and tracks human bodies even when they go outside the camera’s field of view. To achieve this, we first propose a deep generative motion infiller, which autoregressively infills the body motions of occluded humans based on visible motions. Additionally, in contrast to prior work, our approach reconstructs human meshes in consistent global coordinates even with dynamic cameras. Since the joint reconstruction of human motions and camera poses is under-constrained, we propose a global trajectory predictor that generates global human trajectories based on local body movements. Using the predicted trajectories as anchors, we present a global optimization framework that refines the predicted trajectories and optimizes the camera poses to match the video evidence such as 2D keypoints. Experiments on challenging indoor and in-the-wild datasets with dynamic cameras demonstrate that the proposed approach outperforms prior methods significantly in terms of motion infilling and global mesh recovery.

1. Introduction

Recovering fine-grained 3D human meshes from monocular videos is essential for understanding human behaviors and interactions, which can be the cornerstone for numerous applications including virtual or augmented reality, assistive living, autonomous driving, etc. Many of these applications use dynamic cameras to capture human behaviors yet also require estimating human motions in global coordinates consistent with their surroundings. For instance, assistive robots and autonomous vehicles need a holistic understanding of human behaviors and interactions in the world to safely plan their actions even when they are moving. Therefore, our goal in this paper is to tackle the important task of recovering global human meshes from monocular videos captured by dynamic cameras.

However, this task is highly challenging for two main reasons. First, dynamic cameras make it difficult to estimate human motions in consistent global coordinates. Existing human mesh recovery methods estimate human meshes in
the camera coordinates \cite{71,123} or even in the root-relative coordinates \cite{49,72}. Hence, they can only recover global human meshes from dynamic cameras by using SLAM to estimate camera poses \cite{62}. However, SLAM can often fail for in-the-wild videos due to moving and dynamic objects. It also has the problem of scale ambiguity, which often leads to camera poses that are inconsistent with the human motions. Second, videos captured by dynamic cameras often contain severe and long-term occlusions of humans, which can be caused by missed detection, complete obstruction by objects and other people, or the person going outside the camera’s field of view (FoV). These occlusions pose serious challenges to standard human mesh recovery methods, which rely on detections or visible parts to estimate human meshes. Only a few works have attempted to tackle the occlusion problem in human mesh recovery \cite{19,40}. However, these methods can only address partial occlusions of a person and fail to handle severe occlusions when the person is completely invisible for an extended period of time.

To tackle the above challenges, we propose Global Occlusion-Aware Human Mesh Recovery (GLAMR), which can handle severe occlusions and estimate human meshes in consistent global coordinates— even for videos recorded with dynamic cameras. We start by using off-the-shelf methods (e.g., KAMA \cite{37} or SPEC \cite{51}) to estimate the shape and pose sequences (motions) of visible people in the camera coordinates. These methods also rely on multi-object tracking and re-identification, which provide occlusion information, and the motion of occluded frames is not estimated. To tackle potentially severe occlusions, we propose a deep generative motion infiller that autoregressively infills the local body motions of occluded people based on visible motions. The motion infiller leverages human dynamics learned from a large motion database, AMASS \cite{66}. Next, to obtain global motions, we propose a global trajectory predictor that can generate global human trajectories based on local body motions. It is motivated by the observation that the global root trajectory of a person is highly correlated with the local body movements. Finally, using the predicted trajectories as anchors to constrain the solution space, we further propose a global optimization framework that jointly optimizes the global motions and camera poses to match the video evidence such as 2D keypoints.

The contributions of this paper are as follows: (1) We propose the first approach to address long-term occlusions and estimate global 3D human pose and shape from videos captured by dynamic cameras; (2) We propose a novel generative Transformer-based motion infiller that autoregressively fills long-term missing motions, which considerably outperforms state-of-the-art motion infilling methods; (3) We propose a method to generate global human trajectories from local body motions and use the generated trajectories as anchors to constrain global motion and camera optimization; (4) Extensive experiments on challenging indoor and in-the-wild datasets demonstrate that our approach outperforms prior state-of-the-art methods significantly in tackling occlusions and estimating global human meshes.

2. Related Work

Camera-Relative Pose Estimation. 3D human mesh recovery from RGB images or videos is an ill-posed problem due to the depth ambiguity. Most existing methods simplify the problem by estimating human poses relative to the pelvis (root) of the human body \cite{1,6,9–11,23,41,43,44,49,52–56,61,64,72,73,76–78,84,87,92,95,96,104,106,113,117,120,126}. These methods assume an orthographic camera projection model and neglect the absolute 3D translation of the person w.r.t. the camera. To address the lack of translation, recent methods start to estimate human meshes in the camera coordinates \cite{37,40,57,62,80,83,90,105,114,116,118}. Several approaches recover the absolute translation of the person using an optimization framework \cite{68–70,86,115}. A few methods exploit various scene constraints during the optimization process to improve depth prediction \cite{102,114}. Alternatively, recent approaches use physics-based constraints to ensure the physical plausibility of the estimated poses \cite{13,38,90,105,112}. Iqbal et al. \cite{36} exploit a limb-length constraint to recover the absolute translation of the person using a 2.5D representation. Some approaches approximate the depth of the person using the bounding box size \cite{40,71,118}. HybrIK \cite{57} and KAMA \cite{37} employ inverse kinematics to estimate human meshes with absolute translations in the camera coordinates. Several methods directly predict the absolute depth of each person using a heatmap representation \cite{18,123}. Recently, SPEC \cite{51} learns to predict the camera parameters (pitch, yaw, FoV) from the image, which are used for absolute pose regression in the camera coordinates. THUNDR \cite{116} also adopts a similar strategy but uses known camera parameters. While these methods show impressive results, they cannot estimate global human motions from videos captured by dynamic cameras. In contrast, our approach can recover human meshes in consistent global coordinates for dynamic cameras and handle severe and long-term occlusions.

Global Pose Estimation. Most existing methods that estimate 3D poses in world coordinates rely on calibrated, synchronized, and static multi-view capture setups \cite{5,15,17,32,42,82,83,121,122,124}. Huang et al. \cite{7} use uncalibrated cameras but still assume time synchronization and static camera setups. Hasler et al. \cite{26} handle unsynchronized moving cameras but assume multi-view input and rely on audio stream for synchronization. More recently, Dong et al. \cite{16} propose to recover 3D poses from unaligned internet videos of different actors performing the same activity.
from unknown cameras. However, they assume that multiple viewpoints of the same pose are available in the videos. Different from these methods, our approach estimates human meshes in global coordinates from monocular videos recorded with dynamic cameras. Several methods rely on additional IMU sensors or pre-scanned environments to recover global human motions [24, 101], which is unpractical for large-scale adoption. Recently, another line of work starts to focus on estimating accurate human-scene interactions [28, 33, 65, 108]. Liu et al. [62] first obtain the camera poses and dense reconstruction of the scene from dynamic cameras using a SLAM algorithm, COLMAP [88]. The camera poses are used for camera-to-world transformation, while the reconstructed scene is used to encourage human-scene contacts. However, SLAM can often fail for the in-the-wild videos and is prone to error propagation. In contrast, our approach does not require SLAM but instead uses global trajectory prediction to constrain the joint reconstruction of human motions and camera poses. Additionally, our approach can also handle severe and long-term occlusions common in dynamic camera setups.

Occlusion-Aware Pose Estimation. Most existing human pose estimation methods assume the person is fully visible in the images and are not robust to strong occlusions. Only a few methods address the occlusion problem in pose estimation [19, 50, 84, 85, 120]. While these methods show impressive results under partial occlusions, they do not address severe and long-term occlusions when people are completely obstructed or outside the camera’s FoV for a long time. In contrast, our approach leverages deep generative human motion models to tackle severe and long-term occlusions.

Human Motion Modeling. Extensive research has studied 3D human dynamics for various tasks including motion prediction and synthesis [2, 4, 8, 20, 21, 27, 39, 60, 67, 79, 81, 100, 107, 109–111]. Recent human pose estimation methods start to leverage learned human dynamics models to improve the accuracy of estimated motions [49, 84, 119]. Several motion infilling approaches are also proposed to generate complete motions from partially observed motions [25, 31, 45, 46]. Additionally, recent work on motion capture shows that global human translations can be predicted from 3D local joint positions [89]. In contrast to prior work, our trajectory predictor does not require GT root orientations but can predict both global root translations and orientations. Furthermore, we also propose a novel generative autoregressive motion infiller that can use noisy poses as input instead of high-quality GT poses, and we demonstrate its effectiveness in tackling long-term occlusions in human pose estimation.

3. Method

The input to our framework is a video \( I = (I_1, \ldots, I_T) \) with \( T \) frames, which is captured by a dynamic camera, i.e., the camera poses can change every frame. Our goal is to estimate the global motion (pose sequence) \( \{Q^i\}_{i=1}^N \) of the \( N \) people in the video in a consistent global coordinate system. The global motion \( Q^i = (T^i, R^i, \Theta^i, B^i) \) for person \( i \) consists of the root translations \( T^i = (\tau^i_{s_1}, \ldots, \tau^i_{e_i}) \), root rotations \( R^i = (\gamma^i_{s_1}, \ldots, \gamma^i_{e_i}) \), as well as the body motion \( \Theta^i = (\theta^i_{s_1}, \ldots, \theta^i_{e_i}) \) and shapes \( B^i = (\beta^i_{s_1}, \ldots, \beta^i_{e_i}) \), where the motion spans from the first frame \( s_i \) to the last frame \( e_i \), when the person \( i \) is relevant in the video. In particular, each body pose \( \theta^i_{e_i} \in \mathbb{R}^{23 \times 3} \) and shape \( \beta^i_{e_i} \in \mathbb{R}^{10} \) corresponds to the pose parameters (excluding root rotation) and shape parameters of the SMPL model [63]. Using the root translation \( \tau \in \mathbb{R}^3 \) and (axis-angle) rotation \( \gamma \in \mathbb{R}^3 \), SMPL represents a human body mesh with a linear function
$S(\tau, \gamma, \theta, \beta)$ that maps a global pose $q = (\tau, \gamma, \theta, \beta)$ to an articulated triangle mesh $\Phi \in \mathbb{R}^{K \times 3}$ with $K = 6980$ vertices. We can therefore recover the global mesh sequence for each person from their global motion $Q^i$ via SMPL.

As outlined in Fig. 2, our framework consists of four stages. In Stage I, we first use multi-object tracking (MOT) and re-identification algorithms to obtain the bounding box sequence of each person, which is input to a human mesh recovery method (e.g., KAMA [37] or SPEC [51]) to extract the motion $\tilde{Q}^i$ of each person (including translation) in the camera coordinates. The motion $\tilde{Q}^i$ may be incomplete due to various occlusions (e.g., obstruction, missed detection, going outside FoV), where bounding boxes from MOT are missing for some frames. In Stage II (Sec. 3.1), we propose a generative motion infiller to tackle the occlusions in the estimated body motion $\tilde{\Theta}^i$ and produce occlusion-free body motion $\hat{\Theta}^i$. In Stage III (Sec. 3.2), we propose a global trajectory predictor that uses the infilled body motion $\hat{\Theta}^i$ to generate the global trajectory (root translations and rotations) of each person and obtain their global motion $\hat{Q}^i$. In Stage IV (Sec. 3.3), we jointly optimize the global trajectories of all people and the camera parameters to produce global motions $\hat{Q}^i$ consistent with the video evidence.

### 3.1. Generative Motion Infiller

The task of the generative motion infiller $\mathcal{M}$ is to infill the occluded body motion $\tilde{\Theta}^i$ of each person to produce occlusion-free body motion $\hat{\Theta}^i$. Here, we do not use the motion infiller $\mathcal{M}$ to infill other components in the estimated motion $\tilde{Q}^i$, i.e., root trajectory $(\tilde{T}^i, \tilde{R}^i)$ and shapes $\tilde{B}^i$. This is because it is difficult to infill the root trajectory $(\tilde{T}^i, \tilde{R}^i)$ using learned human dynamics, since it resides in the camera coordinates rather than a consistent coordinate system due to the dynamic camera. In Sec. 3.2, we will use the proposed global trajectory predictor to generate occlusion-free global trajectory $(\hat{T}^i, \hat{R}^i)$ from the infilled body motion $\hat{\Theta}^i$. The trajectory $(\hat{T}^i, \hat{R}^i)$ from the pose estimator is not discarded and will be used in the global optimization (Sec. 3.3). We use linear interpolation to produce occlusion-free shapes $\hat{B}^i$, which can be time-varying to be compatible with per-frame pose estimators such as KAMA.

Given a general occluded human body motion $\Theta = (\theta_1, \ldots, \theta_h)$ of $h$ frames and its visibility mask $V = (V_1, \ldots, V_h)$ as input, the motion infiller $\mathcal{M}$ outputs a complete occlusion-free motion $\hat{\Theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_h)$. The visibility mask $V$ encodes the visibility of the occluded motion $\tilde{\Theta}$, where $V_t = 1$ if the body pose $\tilde{\theta}_t$ is visible in frame $t$ and $V_t = 0$ otherwise. Since the human pose for occluded frames can be highly uncertain and stochastic, we formulate the motion infiller $\mathcal{M}$ using the conditional variational autoencoder (CVAE) [48]:

$$\hat{\Theta} = \mathcal{M}(\tilde{\Theta}, V, z),$$  \hspace{1cm} (1)

where the motion infiller $\mathcal{M}$ corresponds to the CVAE decoder and $z$ is a Gaussian latent code. We can obtain different occlusion-free motions $\hat{\Theta}$ by varying $z$.

### Autoregressive Motion Infilling

To ensure that the motion infiller $\mathcal{M}$ can handle much longer test motions than the training motions, we propose an autoregressive motion infilling process at test time as illustrated in Fig. 3 (Left). The key idea is to use a sliding window of $h$ frames, where we assume the first $h_c$ frames of motion are already occlusion-free or infilled and serve as context, and we also use the last $h_l$ frames as look-ahead. The look-ahead is essential to the motion infiller since it may contain visible poses that can guide the ending motion and avoid generating discontinuous motions. Excluding the context and look-ahead frames, only the middle $h_o = h - h_c - h_l$ frames of motion are infilled. We iteratively infill the motion using the sliding
window and advance the window by \(h_c\) frames every step.

**Motion Infiller Network.** The overall network design of the CVAE-based motion infiller is outlined in Fig. 3 (Right). In particular, we employ a Transformer-based seq2seq architecture, which consists of three parts: (1) a context network that uses a Transformer encoder to encode the visible poses from the occluded motion \(\Theta\) into a context sequence, which serves as the condition for other networks; (2) a decoder network that uses the latent code \(z\) and context sequence to generate occlusion-free motion \(\hat{\Theta}\) via a Transformer decoder and a multilayer perceptron (MLP); (3) prior and posterior networks that generate the prior and posterior distributions for the latent code \(z\). In the networks, we adopt a time-based encoding that replaces the position in the original positional encoding [99] with the time index. Unlike prior CNN-based methods [31, 45], our Transformer-based motion infiller does not require padding missing frames, but instead restricts its attention to visible frames to achieve effective temporal modeling.

**Training.** We train the motion infiller \(M\) using a large motion capture dataset, AMASS [66]. To synthesize occluded motions \(\hat{\Theta}\), for any GT training motion \(\tilde{\Theta}\) of \(h\) frames, we randomly occlude \(h_{occ}\) consecutive frames of motion where \(h_{occ}\) is uniformly sampled from \([H_{lb}, H_{ub}]\). Note that we do not occlude the first \(h_c\) frames which are reserved as context. We use the standard CVAE objective to train the motion infiller \(M\):

\[
L_M = \sum_{t=1}^{h} \|\hat{\theta}_t - \tilde{\theta}_t\|^2_2 + \sum_{i=1}^{L_{KL}} L_{KL},
\]

where \(L_{KL}\) is the KL divergence between the prior and posterior distributions of the CVAE latent code \(z\).

### 3.2. Global Trajectory Predictor

After we obtain occlusion-free body motion \(\hat{\Theta}\) for each person using the motion infiller, a key problem still remains: the estimated trajectory \((\hat{T}^i, \hat{R}^i)\) of the person is still occluded and not in a consistent global coordinate system. To tackle this problem, we propose to learn a global trajectory predictor \(T\) that generates a person’s occlusion-free global trajectory \((\hat{T}^i, \hat{R}^i)\) from the local body motion \(\hat{\Theta}\).

Given a general occlusion-free body motion \(\Theta = (\theta_1, \ldots , \theta_m)\) as input, the trajectory predictor \(T\) outputs its corresponding global trajectory \((T, R)\) including the root translations \(T = (t_1, \ldots , t_m)\) and rotations \(R = (\gamma_1, \ldots , \gamma_m)\). To address any potential ambiguity in the global trajectory, we also formulate the global trajectory predictor using the CVAE:

\[
\Psi = T(\Theta, v), \quad (T, R) = \text{EgoToGlobal}(\Psi),
\]

where the global trajectory predictor \(T\) corresponds to the CVAE decoder and \(v\) is the latent code for the CVAE. In Eq. (3), the immediate output of the global trajectory predictor \(T\) is an egocentric trajectory \(\Psi = (\psi_1, \ldots , \psi_m)\), which by design can be converted to a global trajectory \((T, R)\) using a conversion function \(\text{EgoToGlobal}\).

**Egocentric Trajectory Representation.** The egocentric trajectory \(\Psi\) is just an alternative representation of the global trajectory \((T, R)\). It converts the global trajectory into relative local differences and represents rotations and translations in the heading coordinates (y-axis aligned with the heading, i.e., the person’s facing direction). In this way, the egocentric trajectory representation is invariant of the absolute \(xy\) translation and heading. It is more suitable for the prediction of long trajectories, since the network only needs to output the local trajectory change of every frame instead of the potentially large global trajectory offset.

The conversion from the global trajectory to the egocentric trajectory is given by another function: \(\Psi = \text{GlobalToEgo}(T, R)\), which is the inverse of the function \(\text{EgoToGlobal}\). In particular, the egocentric trajectory \(\psi_t = (\delta x_t, \delta y_t, \gamma_t, \delta \phi_t, \eta_t)\) at time \(t\) is computed as:

\[
(\delta x_t, \delta y_t) = \text{ToHeading}(\tau_t^{xy}, \tau_{t-1}^{xy}), \quad z_t = \tau_t^z, \quad \delta \phi_t = \gamma_t^\phi - \gamma_{t-1}^\phi, \quad \eta_t = \text{ToHeading}(\gamma_t),
\]

where \(\tau_t^{xy}\) is the \(xy\) component of the translation \(\tau_t\), \(\tau_t^z\) is the \(z\) component (height) of \(\tau_t\), \(\gamma_t^\phi\) is the heading angle of the rotation \(\gamma_t\), \(\text{ToHeading}\) is a function that converts translations or rotations to the heading coordinates defined by the heading \(\gamma_t^\phi\), and \(\eta_t\) is the local rotation. As an exception, \((\delta x_0, \delta y_0)\) and \(\delta \phi_0\) are used to store the initial \(xy\) translation \(\tau_0^{xy}\) and heading \(\tau_0^z\). These initial values are set to the GT during training and arbitrary values during inference (as the trajectory can start from any position and heading). The inverse process of Eq. (5)-(7) defines the inverse conversion \(\text{EgoToGlobal}\) used in Eq. (4), which accumulates the egocentric trajectory to obtain the global trajectory. To correct potential drifts in the trajectory, in Sec. 3.3, we will optimize the global trajectory of each person to match the video evidence, which also solves the trajectory’s starting point \((\delta x_0, \delta y_0, \delta \phi_0)\). More details about the egocentric trajectory are given in Appendix D.

**Network and Training.** The trajectory predictor adopts a similar network design as the motion infiller with one main difference: we use LSTMs for temporal modeling instead of Transformers since the output of each frame is the local trajectory change in our egocentric trajectory representation, which mainly depends on the body motion of nearby frames and does not require long-range temporal modeling. We will show in Sec. 4.2 that the egocentric trajectory and use
of LSTMs instead of Transformers are crucial for accurate trajectory prediction. Please refer to Appendix D for the detailed network architectures. We use the standard CVAE objective to train the trajectory predictor $T$:

$$L_T = \sum_{t=1}^{m} (\|\tau_t - \tau'_t\|^2_2 + \|\gamma_t \odot \gamma'_t\|^2_2) + L_{\text{KL}},$$

where $\tau'_t$ and $\gamma'_t$ denote the GT translation and rotation, $\odot$ computes the relative rotation, $\| \cdot \|$ computes the rotation angle, and $L_{\text{KL}}$ is the KL divergence between the predicted and posterior distributions of the CVAE latent code $v$. We again use AMASS [66] to train the trajectory predictor $T$.

### 3.3. Global Optimization

After using the generative motion infiller and global trajectory predictor, we have obtained an occlusion-free global motion $\hat{Q}^{i} = (\hat{T}^{i}, \hat{R}^{i}, \hat{\Theta}^{i}, \hat{\Phi}^{i})$ for each person in the video. However, the global trajectory predictor generates trajectories for each person independently, which may not be consistent with the video evidence. To tackle this problem, we propose a global optimization process that jointly optimizes the global trajectories of all people and the extrinsic camera parameters to match the video evidence such as 2D keypoints. The final output of the global optimization and our framework is $\hat{Q}^{i} = (\check{T}^{i}, \check{R}^{i}, \check{\Theta}^{i}, \check{\Phi}^{i})$ where $(\check{\Theta}^{i}, \check{\Phi}^{i}) = (\hat{\Theta}^{i}, \hat{\Phi}^{i})$, i.e., we directly use the occlusion-free body motion and shapes from the previous stages.

#### Optimization Variables

The first set of variables we optimize is the egocentric representation $\{\hat{\Psi}^{i}_t\}_{t=1}^{N}$ of the global trajectories $\{\hat{T}^{i}, \hat{R}^{i}\}_{t=1}^{N}$. We adopt the egocentric representation since it allows corrections of the translation and heading at one frame to propagate to all future frames. Therefore, it enables optimizing the trajectories of occluded frames since they will impact future visible frames under the egocentric trajectory representation. We will empirically demonstrate its effectiveness in Sec. 4.2.

The second set of optimization variables is the extrinsic camera parameters $C = (C_1, \ldots, C_T)$ where $C_t \in \mathbb{R}^{4 \times 4}$ is the camera extrinsic matrix at frame $t$ of the video.

#### Energy Function

The energy function we aim to minimize is defined as

$$E(\{\hat{\Psi}^{i}_t\}_{t=1}^{N}, C) = \lambda_{2D} E_{2D} + \lambda_{\text{tra}} E_{\text{tra}} + \lambda_{\text{reg}} E_{\text{reg}} + \lambda_{\text{can}} E_{\text{can}} + \lambda_{\text{pen}} E_{\text{pen}},$$

where we use five energy terms with their corresponding coefficients $\lambda_{2D}, \lambda_{\text{tra}}, \lambda_{\text{reg}}, \lambda_{\text{can}}, \lambda_{\text{pen}}$.

The first term $E_{2D}$ measures the error between the 2D projection $\hat{x}^{i}_t$ of the optimized 3D keypoints $\hat{X}^{i}_t \in \mathbb{R}^{4 \times 3}$ and the estimated 2D keypoints $\tilde{x}^{i}_t$ from a keypoint detector:

$$E_{2D} = \frac{1}{NTJ} \sum_{i=1}^{N} \sum_{t=1}^{T} V^i_t \|\hat{x}^{i}_t - \tilde{x}^{i}_t\|^2_2,$$

where $V^i_t$ is person $i$’s visibility at frame $t$, $\Pi$ is the camera projection with extrinsics $C_t$ and approximated intrinsics $K$, and $\hat{x}^{i}_t$ is computed using the SMPL joint function $J$ from the optimized global pose $\hat{q}^{i}_t = (\hat{v}^{i}_t, \hat{\gamma}^{i}_t, \hat{\beta}^{i}_t, \hat{\beta}^{i}_t)$ in $\hat{Q}^{i}$.

The second term $E_{\text{tra}}$ measures the difference between the optimized global trajectory $\hat{T}^{i}, \hat{R}^{i}$ viewed in the camera coordinates and the trajectory $T^{i}, R^{i}$ output by the pose estimator (e.g., KAMA [37]) in Stage 1:

$$E_{\text{tra}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} V^i_t \left(\|\hat{\gamma}^{i}_t, C_t \odot \hat{\gamma}^{i}_t\|^2_a + \|w_t \|\hat{\gamma}^{i}_t, C_t - \hat{\gamma}^{i}_t\|^2_2\right),$$

where the function $\Gamma(\cdot, C_t)$ transforms the global rotation $\hat{\gamma}^{i}_t$ or translation $\hat{\tau}^{i}_t$ to the camera coordinates defined by $C_t$, and $w_t$ is a weighting factor for the translation term.

The third term $E_{\text{reg}}$ regularizes the egocentric trajectory $\hat{\Psi}^{i}_t$ to stay close to the output $\hat{\Psi}^{i}_t$ of the trajectory predictor:

$$E_{\text{reg}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \|w_{\psi} \circ \hat{\psi}^{i}_t - \hat{\psi}^{i}_t\|^2_2,$$

where $\circ$ denotes the element-wise product and $w_{\psi}$ is a weighting vector for each element inside the egocentric trajectory. As an exception, we do not regularize each person’s initial $xy$ position and heading $(\delta \hat{x}^{i}_0, \delta \hat{y}^{i}_0, \delta \hat{\phi}^{i}_0) \subset \hat{\psi}^{i}_0$ as they need to be inferred from the video.

The fourth term $E_{\text{can}}$ measures the smoothness of the camera parameters $C$ and the uprightness of the camera:

$$E_{\text{can}} = \frac{1}{T} \sum_{t=1}^{T} \langle C^{i}_t, Y \rangle + \frac{1}{T-1} \sum_{t=2}^{T} \|C^{i}_t - C^{i}_{t-1}\|^2_2 + \|C^{i}_t - C^{i}_0\|^2_2,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $C^{i}_0$ is the $+y$ vector of the camera $C_t$, and $Y$ is the global up direction. $C^{i}_t$ and $C^{i}_t$ denote the rotation and translation of the camera $C_t$.

The final term $E_{\text{pen}}$ is an signed distance field (SDF)-based inter-person penetration loss adopted from [40].
4. Experiments

Datasets. We employ the following datasets in our experiments: (1) AMASS [66], which is a large human motion database with 11000+ human motions. We use AMASS to train and evaluate the motion infiller and trajectory predictor. (2) 3DPW [101], which is an in-the-wild human motion dataset that uses videos and wearable IMU sensors to obtain GT poses, even when the person is occluded. We evaluate our approach using the test split of 3DPW. (3) Dynamic Human3.6M is a new benchmark for human pose estimation with dynamic cameras that we create from the Human3.6M dataset [35]. We simulate dynamic cameras and occlusions by cropping each frame with a small view window that oscillates around the person (see Fig. 5). More details are provided in Appendix A.

Evaluation Metrics. We use the following metrics for evaluation: (1) G-MPJPE and G-PVE, which extend the mean per joint position error (MPJPE) and per-vertex error (PVE) by computing the errors in the global coordinates. As errors in estimated global trajectories accumulate over time in our dynamic camera setting, we follow standard evaluations for open-loop reconstruction (e.g., SLAM [93] and inertial odometry [30]) to compute errors using a sliding window (10 seconds) and align the root translation and rotation with the GT at the start of the window. (2) PA-MPJPE, which is the Procrustes-aligned MPJPE for evaluating estimated body poses. For invisible poses, since there can be many plausible poses beside the GT, we follow prior work [3,110] to compute the best PA-MPJPE out of multiple samples for our probabilistic approach. (3) Accel, which computes the mean acceleration error of each joint and is commonly used to measure the jitter in estimated motions [39,112]. (4) FID, which is an extension of the original Frechet Inception Distance that calculates the distribution distance between estimated motions and the GT. FID is a standard metric in motion generation literature to evaluate the quality of generated motions [34,58,59,98]. Following prior work [59], we compute FID using the well-designed kinetic motion feature extractor in the fairmotion library [22].

Implementation Details. Thorough details about the entire framework are provided in Appendix A to E.

4.1. Evaluation of GLAMR

Baselines. Since no prior methods can estimate global motions from dynamic cameras and address long-term occlusions, we design various baselines by combining state-of-the-art human mesh recovery methods (KAMA [37] or SPEC [51]), motion infilling methods, and SLAM-based camera estimation (OpenSfM [74]). In particular, we use the estimated camera parameters to convert estimated motions from the camera coordinates to the global coordinates. For motion infilling, we use (1) linear interpolation, (2) last pose, i.e., replicating the last visible pose, and (3) a state-of-the-art CNN-based motion infilling method, ConvAE [45]. The results on Dynamic Human3.6M and 3DPW are summarized in Table 1 and 2 respectively. We only report G-MPJPE and G-PVE on Dynamic Human3.6M since they require accurate GT trajectories, which 3DPW does not provide. It is evident that our approach, GLAMR, outperforms the baselines in almost all metrics. In particular, GLAMR achieves significantly lower G-MPJPE and G-PVE, which demonstrates its strong ability to reconstruct global human motions. Furthermore, GLAMR retains considerably lower FID and PA-MPJPE (with ten samples) for occluded (invisible) poses. The lower FID means GLAMR can infill more humanlike motions, and the lower PA-MPJPE also shows GLAMR’s probabilistic motion samples can cover the GT better. Finally, while GLAMR achieves almost the same PA-MPJPE for visible poses as the best method, it yields much smoother motions (smaller acceleration error). This is because our motion infiller leverages human dynamics learned from a large motion dataset to produce motions.

Qualitative Results. Fig. 4 and 5 show qualitative comparisons of GLAMR against the strong baseline, KAMA + Linear Interpolation. Additionally, we provide abundant qualitative results on the project page.

4.2. Evaluation of Key Components

Benchmarking Motion Infiller. We evaluate the proposed generative motion infiller on the test split of the AMASS dataset [66]. We compare against three motion infilling baselines: linear interpolation, replicating the last pose, and ConvAE [45]. As shown in Table 3, our generative motion infiller achieves significantly better PA-MPJPE for both
Figure 4. Qualitative comparison of GLAMR with a strong baseline on 3DPW. The infilled motion (transparent) by GLAMR is more natural especially for the legs, while the baseline has very slow leg motions due to interpolation in a large window (frame 10 to 75). On the right, we plot how the x-axis joint angles of left and right hips of the person (green) change over time for GLAMR and the baseline.

Figure 5. Qualitative comparison of GLAMR on Dynamic Human3.6M. GLAMR can generate natural hand motions for invisible frames instead of just doing linear interpolation.

Table 3. Benchmarking motion infiller on AMASS.

| Method                  | (Sampled) PA-MPJPE | (Reconstructed) PA-MPJPE | FID   |
|-------------------------|--------------------|--------------------------|-------|
| Linear Interpolation    | 83.5               | 83.5                     | 35.3  |
| Last Pose               | 104.4              | 104.4                    | 41.6  |
| ConvAE [45]             | 72.8               | 72.8                     | 31.4  |
| Ours                    | 61.4               | 36.1                     | 16.7  |

Table 4. Benchmarking trajectory predictor on AMASS.

| Method                           | G-MPJPE | G-PVE | Accel |
|----------------------------------|---------|-------|-------|
| Transformer                      | 660.1   | 678.6 | 121.9 |
| Ours w/o Ego Trajectory          | 763.0   | 780.6 | 8.7   |
| Ours (GLAMR)                     | 806.2   | 824.1 | 6.0   |

Table 5. Global optimization ablations on Dynamic Human3.6M.

| Method                           | G-MPJPE | G-PVE | Accel |
|----------------------------------|---------|-------|-------|
| Ours w/o Trajectory Predictor    | 1750.8  | 1761.4| 12.6  |
| Ours w/o Opt Ego Trajectory      | 877.3   | 895.0 | 15.5  |
| Ours (GLAMR)                     | 806.2   | 824.1 | 6.0   |

5. Discussion and Limitations

In this paper, we proposed an approach for 3D human mesh recovery in consistent global coordinates from videos captured by dynamic cameras. We first proposed a novel Transformer-based generative motion infiller to address severe occlusions that often come with dynamic cameras. To resolve ambiguity in the joint reconstruction of global human motions and camera poses, we proposed a new solu-
tion by predicting global human trajectories from local body motions. Finally, we proposed a global optimization framework to refine the predicted trajectories, which serve as anchors for camera optimization. Our method achieves SOTA results on challenging datasets and marks a significant step towards global human mesh recovery in the wild.

As the first paper on this new problem, our method has a few limitations: propagation of errors in multiple stages, limited body shape estimation, not being real-time, not including scene information, etc. A detailed discussion is provided in Appendix H. We believe these limitations are exciting avenues for future work to explore.

References

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In CVPR, 2015. 2
[2] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3D human motion modelling. In ICCV, 2019. 3
[3] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In CVPR, 2020. 7
[4] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3D human motion prediction via gan. In CVPR Workshops, 2018. 3
[5] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3D pictorial structures for multiple human pose estimation. In CVPR, 2014. 2
[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In ECCV, 2016. 2
[7] Tianshu Zhang Buzhen Huang, Yuanyu Fu and Yangang Wang. Dynamic multi-person mesh recovery from uncalibrated multi-view cameras. In 3DV, 2021. 2
[8] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In ECCV, pages 387–404. Springer, 2020. 3
[9] Hongshu Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In ECCV, 2020. 2
[10] Hongshu Choi, Gyeongsik Moon, and Kyoung Mu Lee. Beyond static features for temporally consistent 3D human pose and shape from a video. In CVPR, 2021. 2
[11] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In ECCV, 2020. 2
[12] MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. https://github.com/open-mmlab/mmtracking, 2020. 13
[13] Rishabh Dabral, Soshi Shimada, Arjun Jain, Christian Theobalt, and Vladislav Golyanik. Gravity-aware monocular 3D human-object reconstruction. In ICCV, 2021. 2
[14] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In ECCV, pages 612–628. Springer, 2020. 16
[15] Junting Dong, Qi Fang, Wen Jiang, Yurou Yang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3D pose estimation and tracking from multiple views. TPAMI, 2021. 2
[16] Junting Dong, Qing Shuai, Yuanqing Zhang, Xian Liu, Xiaowei Zhou, and Hujun Bao. Motion capture from internet videos. In ECCV, 2020. 2
[17] Zijian Dong, Jie Song, Xu Chen, Chen Guo, and Otmar Hilliges. Shape-aware multi-person pose estimation from multi-view images. In ICCV, 2021. 2
[18] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Compressed volumetric heatmaps for multi-person 3D pose estimation. In CVPR, June 2020. 2
[19] Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. Three-dimensional reconstruction of human interactions. In CVPR, 2020. 2, 3
[20] Katerina Fragiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In ICCV, 2015. 3
[21] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In CVPR, 2019. 3
[22] Deepak Gopinath and Jungdam Won. fairmotion - tools to load, process and visualize motion capture data. https://github.com/facebookresearch/fairmotion, 2020. 7
[23] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3D human reconstruction in-the-wild. In CVPR, 2019. 2
[24] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human pose estimation and self-localization in large scenes from body-mounted sensors. In CVPR, 2021. 3
[25] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. ACM Transactions on Graphics (TOG), 39(4):60–1, 2020. 3
[26] Nils Hasler, Bodo Rosenhahn, Thorsten Thormählen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. In CVPR, 2009. 2
[27] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In ICCV, pages 11374–11384, 2021. 3
[28] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3D human pose ambiguities with 3D scene constraints. In ICCV, pages 2282–2292, 2019. 3
[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016. 13

[30] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In ICRA, 2020. 7

[31] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In CVPR, 2019. 3, 5

[32] Buzhen Huang, Yuan Shu, TianShu Zhang, and Yangang Wang. Dynamic multi-person mesh recovery from uncalibrated multi-view cameras. In 3DV, 2021. 2

[33] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J Black. Capturing and inferring dense full-body human-scene contact. In CVPR, 2022. 3

[34] Ruozhi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with the music via curriculum learning. In ICLR, 2021. 7

[35] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. TPAMI, 36(7):1325–1339, 2014. 7, 13

[36] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In CVPR, 2020. 2

[37] Umar Iqbal, Kevin Xie, Yunrong Guo, Jan Kautz, and Pavlo Molchanov. Kama: 3d keypoint aware body mesh articulation. In 3DV, 2021. 2, 4, 6, 7, 13, 15

[38] Mariko Isogawa, Ye Yuan, Matthew O’Toole, and Kris M Kitani. Optical non-line-of-sight physics-based 3d human pose estimation. In CVPR, 2020. 2

[39] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rmn: Deep learning on spatio-temporal graphs. In CVPR, 2016. 3

[40] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In CVPR, 2020. 2, 6

[41] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. In 3DV, 2021. 2

[42] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In CVPR, 2018. 2

[43] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In CVPR, 2018. 2, 13

[44] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In CVPR, 2019. 2

[45] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In 3DV, 2020. 3, 5, 7, 8

[46] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. In ICCV, pages 3174–3184, 2021. 3

[47] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In arXiv preprint arXiv:1412.6980, 2014. 14, 15

[48] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013. 4

[49] Mohammad Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In CVPR, 2020. 2, 3, 7

[50] Mohammad Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regression for 3D human body estimation. In ICCV, 2021. 3

[51] Mohammad Kocabas, Chun-Hao P. Huang, Joachim Tesch, Lea Müller, Otmar Hilliges, and Michael J. Black. SPEC: Seeing people in the wild with an estimated camera. In ICCV, 2021. 2, 4, 7, 13

[52] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In ICCV, 2019. 2

[53] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In CVPR, 2019. 2

[54] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In ICCV, 2021. 2

[55] Jegendra Nath Kundu, Mugalodi Rakesh, Varun Jampani, Rahul Mysore Venkatesh, and R. Venkatesh Babu1. Appearance consensus driven self-supervised human mesh recovery. In ECCV, 2020. 2

[56] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In CVPR, 2017. 2

[57] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In CVPR, 2021. 2

[58] Jianli Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. arXiv preprint arXiv:2008.08171, 2020. 7

[59] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In ICCV, 2021. 7

[60] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Learning to generate diverse dance motions with transformer. arXiv preprint arXiv:1412.6980, 2013. 4

[61] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In CVPR, 2021. 2

[62] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4d human body capture from egocentric video via 3d scene grounding. In 3DV, 2021. 2, 3

[63] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. SIGGRAPH Asia, 34(6):248:1–248:16, 2015. 3
A. Details for the Datasets

AMASS [66] is a large human motion database with 11000+ human motions. We use AMASS to train and evaluate the motion infiller and trajectory predictor. Specifically, we use the Transitions, SSM, and HumanEva [91] subsets for testing and all other subsets for training.

3DPW [101] is an in-the-wild human motion dataset that consists of 60 videos recorded with dynamic cameras in diverse environments. The GT 3D poses are obtained using wearable IMU sensors. Since non-optical sensors are used to obtain GT data, the dataset also provides body pose information when the persons go outside the FoV of the camera. 3DPW also provides the global trajectories of people in the dataset. However, the global trajectories are quite inaccurate since they are estimated from IMU data. Therefore, we do not use 3DPW to evaluate global trajectory reconstruction in the paper. Since we do not use 3DPW for training, we use sequences from the entire 3DPW dataset for visualization. We use the official 3DPW test split to report quantitative results in the paper.

Dynamic Human3.6M is a new benchmark for global human pose estimation with dynamic cameras that we create from the Human3.6M dataset [35]. We simulate dynamic cameras and occlusions by cropping each frame with a view window of 300 × 600 that horizontally oscillates around the person’s bounding box center with a period of 4.8 seconds and a magnitude of 200 pixels. In this way, we synthesize large camera motions and severe occlusions where the person is occluded for almost half of the time, which makes it very challenging for existing 3D human pose and shape estimation methods. Additionally, since Human3.6M provides accurate global human trajectories and human poses, we use Dynamic Human3.6M to evaluate global trajectory reconstruction and pose estimation for occluded frames. We follow the standard protocol [43] and use the official test split (subjects 9 and 11) for evaluation. Please refer to the [supplementary video](https://youtu.be/wpObDXcYueo) for an example sequence of the Dynamic Human3.6M dataset. Code for generating Dynamic Human3.6M are available here for users who have downloaded the original Human3.6M dataset [35].

B. Implementation Details for Preprocessing

3D Multi-Object Tracking and Re-identification. We use DeepSORT [103] with ResNet-50 [29] in the MMTracking package [12] for 3D multi-object tracking (MOT) and re-identification. We use the GT tracks to evaluate our approach and the baselines, following the standard protocol for human pose estimation.

Initial Human Pose and Shape Estimation. As mentioned in the main paper, we use KAMA [37] or SPEC [51] to provide the initial human pose and shape estimation from the bounding boxes extracted by 3D MOT. We choose these two methods since both KAMA and SPEC estimate 3D human poses in the camera coordinates with absolute root translations, while many state-of-the-art human pose estimation methods do not provide the root translations. We also use HRNet [94] to extract 2D human keypoints from the video, which are used in the proposed global optimization framework.

C. Implementation Details for Generative Motion Infiller

![Figure 6. The detailed network architecture of the CVAE-based generative motion infiller. For all the Transformer modules, the dimensions for keys, queries, and values are set to 256, the number of transformer blocks is 2, the hidden dimensions of the feedforwards layers are 512, the dropout rate is 0.1, and 8 heads are used for the multi-head attention. Two hidden layers (512, 256) with ReLU activations are used for all the token-wise MLPs.](image-url)
Network Architecture. The detailed network architecture of the CVAE-based generative motion infiller is outlined in Fig. 6. For all the Transformer modules, the dimensions for keys, queries, and values are set to 256, the number of transformer blocks is 2, the hidden dimensions of the feedforwards layers are 512, the dropout rate is 0.1, and 8 heads are used for the multi-head attention. The time-based encoding takes the same sinusoidal form as the original positional encoding [99] but replaces the position with the time index. We use two hidden layers (512, 256) with ReLU activations for all the token-wise MLPs. In the prior network, two learnable tokens are used to form queries to produce the mean \( \mu^p_z \) and standard deviation \( \sigma^p_z \) of the prior distribution of the latent code \( z \). Similarly, in the posterior network, two learnable tokens are appended to the GT pose sequence \( \hat{\Theta} \) to output the mean \( \mu^q_z \) and standard deviation \( \sigma^q_z \) of the posterior distribution of the latent code \( z \).

Hyperparameters and Training. The dimension of the latent code \( z \) is 128. The sliding window size \( h \) of the autoregressive motion infilling is 50. Both the number of context frames \( h_c \) and the number of look-ahead \( h_l \) frames are 10. When synthesizing occluded motions, for any GT training motion of \( h = 50 \) frames, we randomly occlude \( H_{occ} \) consecutive frames of motion where \( H_{occ} \) is uniformly sampled from \([10, 40]\). Note that we do not occlude the first \( h_c = 10 \) frames which are reserved as context. The KL divergence term in Eq. (2) uses a weighting factor of 0.001. We train the networks for 2000 epochs with a batch size of 1024 where each epoch uses a total of 10 million frames of motion. For optimization, we use the Adam optimizer [47] with a learning rate of 0.001 and clip the gradient if its norm is larger than 5. We use PyTorch [75] to implement and train the networks.

D. Implementation Details for Global Trajectory Predictor

Heading Coordinate and Egocentric Trajectory Representation. The heading vector of a person points towards where the person is facing and is parallel to the ground. We obtain the heading vector by aligning the \( z \)-axis of the person’s root coordinate with the world \( z \)-axis and use the resulting \( y \)-axis of the aligned root coordinate as the heading vector. This way of obtaining the heading is more stable than using the yaw of the Euler angle representation, which suffers from singularities and can be quite unstable. The heading coordinate is defined by first placing the world coordinate at the root position of the person and then rotating the world coordinate around the \( z \)-axis (vertical) to align the \( y \)-axis with the heading vector. By definition, representing and predicting human trajectories in the heading coordinate allows the predicted trajectory to be invariant of the person’s absolute \( xy \) translation and heading. In the egocentric trajectory representation \( \psi_t = (\delta x_t, \delta y_t, z_t, \delta \phi_t, \eta_t) \), we use absolute height \( z_t \) since the height of a person relative to the ground does not vary a lot and is highly correlated with the body motion of the person. For the local rotation \( \eta_t \), we adopt the 6D rotation representation [125] to avoid discontinuity.

Network Architecture. The detailed network architecture of the CVAE-based global trajectory predictor is illustrated in Fig. 7. We use two bidirectional LSTM layers with hidden dimension 256 for all the LSTM blocks in the networks. We use two hidden layers (512, 256) with ReLU activations for all the token-wise MLPs. For the input poses, we first convert them to
3D joint positions using the SMPL joint function without global rotations and translations. This is because we find that using 3D joint positions leads to better performance than using joint rotations directly. In both the prior and posterior networks, token-wise mean pooling is used to produce a single feature from a sequence of tokens, which is then used to produce the parameters of the prior or posterior distribution of the latent code $v$.

**Hyperparameters and Training.** The dimension of the latent code $v$ is 128. The KL divergence term in Eq. (8) uses a weighting factor of 0.001. We train the networks for 2000 epochs with a batch size of 256 where each epoch uses a total of 2 million frames of motion. The training sequence length is 100 frames. For optimization, we use the Adam optimizer [47] with a learning rate of 0.0001 and clip the gradient if its norm is larger than 5. We use PyTorch [75] to implement and train the networks.

**E. Implementation Details for Global Optimization**

**Initialization.** We initialize the egocentric trajectories using the output from the global trajectory predictor. For the camera, we approximate the camera intrinsic parameters $K$ using the dimensions of the image $[w, h]$ where we assume the principal point is at the image center $[w/2, h/2]$. Note that the camera intrinsics are kept fixed during the optimization process. For the camera extrinsic parameters $C$, we initialize them from the persons’ global trajectories using the following equations:

$$C_t = \Omega \left( \frac{1}{\sum_{i=1}^{N} V_t^i} \sum_{i=1}^{N} V_t^i \cdot P_t^{i,\text{global}} \cdot P_t^{i,\text{cam}}^{-1} \right),$$

where $V_t^i$ is the visibility of person $i$ at frame $t$, $P_t^{i,\text{global}} \in \mathbb{R}^{4 \times 4}$ is the person’s transformation in the global coordinates based on the predicted global trajectory $(\hat{T}, \hat{R})$, $P_t^{i,\text{cam}} \in \mathbb{R}^{4 \times 4}$ is the person’s transformation in the camera coordinates based on the estimated trajectory $(\hat{T}, \hat{R})$ by the pose estimator (e.g., KAMA [37]), $\Omega$ is a projection operator that projects the matrix into a valid transformation. If no person is visible at frame $t$, the camera extrinsics $C_t$ is initialized to the camera extrinsics of the most recent frame with visible people. Eq. (15) is the least squares solutions of the following (transposed) linear systems:

$$P_t^{i,\text{global}} = C_t \cdot P_t^{i,\text{cam}} , \quad \forall i, V_t^i = 1.$$

**Hyperparameters and Optimization.** The optimization loss coefficients $(\lambda_{\text{3D}}, \lambda_{\text{tra}}, \lambda_{\text{reg}}, \lambda_{\text{cam}}, \lambda_{\text{pen}})$ in Eq. (9) are set to $(1, 100000, 100, 10000, 100000)$ for 3DPW and $(1, 100000, 100, 10000, 0)$ for Human3.6M. We do not use the inter-person penetration loss for Human3.6M since it only has one person in each video. The weighting factor $w_i$ for the translation term in Eq. (12) is set to 0 since the translation estimated by the pose estimator can be quite noisy. The trajectory regularization weighting factor $w_\psi$ in Eq. (13) is set to $(3,10,10000,5,10000)$ for each element in the egocentric trajectory $\psi_t = (\delta x_t, \delta y_t, z_t, \delta \phi_t, \eta_t)$, where we use large weights to penalize changes in height $z_t$ and local rotation $\eta_t$. The global optimization is also implemented in PyTorch [75], where we use the Adam optimizer [47] with a learning rate of 0.001 to optimize the global trajectories and camera extrinsics.

**Computation Time.** The overall processing time for a 1-min scene is around 5 mins with 500 optimization iterations, which is much faster than using OpenSfM (> 30 mins).

**F. Evaluation of Global Optimization on 3DPW**

| Method                  | Relative Translation Error | Relative Rotation Error |
|-------------------------|----------------------------|-------------------------|
| Ours w/o Global Optimization | 1.92                       | 1.07                    |
| Ours (GLAMR)            | 0.66                       | 0.30                    |

Table 6. Evaluation of our global optimization framework on 3DPW. We evaluate the relative translation error (in meters) and relative rotation error (in angles) between pairs of humans. Here, “relative” denotes the relative spatial relationship between two humans.

We also perform experiments on 3DPW with and without our global optimization framework to study the importance of global optimization when there are multiple people in the video. Although 3DPW does not provide accurate GT human trajectories in the global coordinates, the relative translations and rotations between people in 3DPW are quite accurate. Therefore, we compute the relative translations and rotations between pairs of humans and calculate their errors w.r.t. the
ground truth. These metrics, i.e., relative translation and rotation errors, serve as an alternative way to evaluate global reconstruction quality. As shown in Table 6, using global optimization can greatly reduce the relative translation and rotation errors between humans, which means our global optimization framework can greatly help to reconstruct the spatial relationships of humans in the video.

**G. Effect of Sliding Window Length.**

As shown in Fig. 8, when increasing the window length $h$ (with context $h_c$ and look-ahead $h_l$ being $0.2h$), the reconstruction error increases because it is harder for the latent code $z$ to encode a longer window which contains more motion variations than a shorter window. In the meantime, the sample error first drops and then increases since there is a trade-off: a longer window provides more context for better inference, but it also puts more burden on the latent code as indicated by the increasing reconstruction error.

![AMASS Motion Infilling Results](image)

**Figure 8.** Sample and reconstruction PA-MPJPE vs. sliding window length $h$. The context $h_c$ and look-ahead $h_l$ are always $0.2h$.

**Motion Infilling without Visible Pose.** In the extreme case, when there is no visible pose ($h_c = h_l = 0$), our motion infiller can still produce plausible motions sampled from the prior learned from the training motion datasets. In this case, the motion infiller essentially becomes an unconditional VAE model.

**H. Discussion of Limitations**

As the first paper on this new problem, our method has a few limitations that are important for future research to address. First, our approach has five stages that are sequentially dependent. Therefore, errors in early stages can propagate to late stages, which may lead to inaccurate global pose estimation. Future work could integrate these stages together to form an end-to-end learnable framework. Second, like many works in human mesh recovery, our approach can only recover the SMPL parameters which omit the fine details of human meshes such as clothing. Integrating neural articulated shapes such as [14] into our approach could potentially address this problem. Third, our approach is not real-time due to the batch processing and global optimization. Future work could explore a causal version of our approach where only a small window around the incoming frame is optimized, which could substantially improve computational efficiency. Finally, the generative motion infiller and global trajectory predictor in our approach operate for each person independently. Therefore, the generated motions and trajectories may not capture potentially complex and nuanced interactions between occluded people such as hugging or dancing. Future work could address this limitation by employing new generative models that produce interaction-aware motions of multiple people.

**I. Discussion of Potential Negative Impact**

With its strong ability to reconstruct global human motions and tackle severe occlusions, our method marks a significant step towards global human mesh recovery in the wild. However, misuse of this technology could lead to potential privacy concerns and the propagation of misinformation. For instance, combined with advanced neural rendering approaches [97], the reconstructed global human motion of our approach could be used to fabricate videos of human actions that are indistinguishable from real ones. To address this issue, future research should continue to study the detection of synthesized videos with realistic human motion.