H4D: Human 4D Modeling by Learning Neural Compositional Representation

Boyan Jiang[1*] Yinda Zhang[2*] Xingkui Wei[1] Xiangyang Xue[1] Yanwei Fu[1]
[1] Fudan University [2] Google

Abstract

Despite the impressive results achieved by deep learning based 3D reconstruction, the techniques of directly learning to model the 4D human captures with detailed geometry have been less studied. This work presents a novel framework that can effectively learn a compact and compositional representation for dynamic human by exploiting the human body prior from the widely-used SMPL parametric model. Particularly, our representation, named H4D, represents dynamic 3D human over a temporal span into the latent spaces encoding shape, initial pose, motion and auxiliary information. A simple yet effective linear motion model is proposed to provide a rough and regularized motion estimation, followed by per-frame compensation for pose and geometry details with the residual encoded in the auxiliary code. Technically, we introduce novel GRU-based architectures to facilitate learning and improve the representation capability. Extensive experiments demonstrate our method is not only efficacy in recovering dynamic human with accurate motion and detailed geometry, but also amenable to various 4D human related tasks, including motion retargeting, motion completion and future prediction.

1. Introduction

The vanilla SMPL based parametric representations have been extensively studied and widely utilized for modeling 3D human shape, and thus shown critical impacts to many human-centric tasks, such as pose estimation [16, 23, 29, 31, 33, 45] and body shape fitting [9, 18, 32, 52, 62]. However, these representations are arguably insufficient for applications involving dynamic signals, *e.g.* 3D moving humans (Fig.1 top), since the temporal information is not captured.

As solutions, 4D representations are proposed and can be in general categorized into free-form and prior-based methods depending on the 3D representation of the output shape (Fig. 1). The free-form methods leveraging Neural ODE [13] and deep implicit function [25, 48] often rely on computational expensive architectures to learn the compact latent spaces and reconstruct the 4D sequences. Unfortunately, since the human body prior is not explicitly modeled, the reconstruction results of these methods may contain obvious geometry artifacts such as missing hands, and their modeling errors accumulate rapidly over time. On the other hand, prior-based methods [29, 31, 68] are mostly derived from the well-known SMPL parametric model [39], which typically employs one single shape parameter and a series of pose parameters per temporal frame to model dynamics. Though producing plausible results, the motion representations in these methods are not compact or only support a small time span, *e.g.* $\pm$ 5 frames [29].

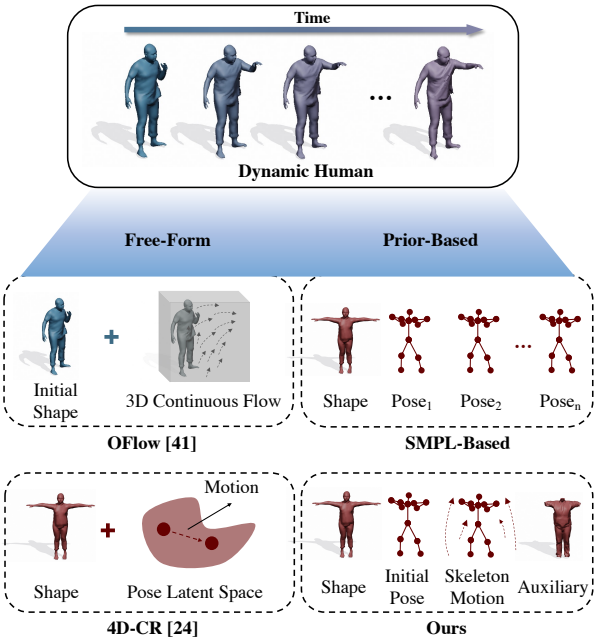


Figure 1. **Comparison with existing 4D human representations.** Our representation supports faster inference and more complete reconstructions compared with free-form methods (Fig. 3). And it provides long-range temporal context and additional fine-grained geometry controlled by low-dimensional latent codes, which is more compact compared with previous SMPL-based methods.

* indicates equal contributions.

Boyan Jiang, Xingkui Wei and Xiangyang Xue are with the School of Computer Science, Fudan University.

Yanwei Fu is with the School of Data Science, Fudan University, and Fudan ISTBI—ZJNU Algorithm Centre for Brain-inspired Intelligence, Zhejiang Normal University, Jinhua, China. This work was supported by NSFC Project 62076067.

In this paper, we propose H4D, which is a novel neural representation for human 4D modeling that combines the merits of both the prior-based and free-form solutions. To reflect the compositional natures, we encode each temporal sequence of 3D human shapes with compact latent codes representing body shape, initial pose, and temporal motion respectively, which can then be used to reconstruct the input sequence through a decoder. At the core of this decoder, a simple yet effective parametric model extended from SMPL [39] is designed to provide the coarse but long-term estimations of the 3D human geometry and motion. This can ensure more complete and plausible outputs compared to the prior arts of free-form reconstructions [25, 48], but potentially be inclined to suffer from the limited representation capability. To this end, we add an additional auxiliary latent code to our representation to compensate the inaccurate motion and enrich the geometry details. Such a representation takes full advantage of parametric models in exploiting strong prior based regularization for plausible initialization and complement them with powerful deep learning components to facilitate the human 4D modeling with impressive motion and geometry accuracy.

Our representation is learned via an auto-encoding framework. The encoder predicts latent codes for each aspect from densely sampled point clouds. These codes are then fed into the decoder to reconstruct the identical input dynamic human sequence. Once trained, the encoder and decoder are both fixed to support various applications, such as motion retargeting, completion, and prediction, through either forward propagation (feed-forward) or backward optimization (auto-decoding) depending on the inputs. We design novel Gated Recurrent Unit (GRU) [15] based architectures for both encoder and decoder to benefit the model performance while working in either mode. In feed-forward mode, we do not require the input point clouds to be temporarily tracked, *i.e*. the point trajectories like in previous work [25, 48]. This simplifies the training requirements and enhances the applicability of high-level applications. In auto-decoding mode, our model leverages the temporal information for optimization, which is critical for robustness to recover details of motion and geometry.

Contributions We propose H4D, a compact and compositional representation for 4D human captures, which combines a linear prior model with residual encoded in a learned auxiliary code. The framework is learned via 4D reconstruction, and the latent representation can be extracted from either non-registered point clouds in a feed-forward fashion or auto-decoding through optimization. Extensive experiments show that our representation and GRU-based architecture are effective in recovering accurate dynamic human sequence and providing robust performance for a variety of 4D human related applications, including motion retargeting/completion and future prediction.

2. Related Work

4D Representation There has been a lot of work aiming to reconstruct 3D object based on various representations, such as 3D voxels [17, 21, 66], point clouds [1, 20, 54, 55], meshes [12, 22, 28, 35, 65] and implicit functions [11, 14, 19, 26, 46, 50]. However, the deep representations for 4D data, *i.e.* a time-varying 3D object, has received less attention mostly due to the challenge of encoding the temporal dimension. Pioneer work mostly relies on Neural ODE [13] and combines with occupancy network [46, 48], point clouds [58], and compositional property [25]. Despite state-of-the-art performance in various motion-related tasks, Neural ODE tends to accumulate errors over time that causes incomplete geometry, and slows down training convergence and inference run-time. In contrast, our model relies on the prior model for comprehensive geometry and motion and recurrent network for efficient inference.

Human Body Estimation For human shape and pose estimation [9, 16, 23, 31–33, 45, 52] or motion prediction [2, 3, 7, 10, 42, 43], most of works are based on SMPL or its extension [39, 51, 59]. Specifically, HMMR [29] learns to encode temporal information by reconstructing a small number of past and future frames. Zhang *et al*. [68] propose the first autoregressive model for predicting 3D human motion from image sequences. VIBE [31] leverages GRU to regress SMPL parameters, and designs an adversarial learning framework to predict temporal transitions. Though producing plausible motion, the motion representations in these methods are either implicit [31], coupled with geometry [29, 31, 68], or limited to short temporal range [29]. In contrast, we formulate the motion with a low-dimensional prior model for long-range context followed by per-frame adjustments controlled by a learned latent code, which is compact, compositional, and tolerant to error accumulation.

Fine-grained Human Reconstruction Many human reconstruction methods [9, 28, 29, 31, 32, 45] are limited to the minimally-clothed bodies as SMPL-based models may suffer from limited expressive power in the shape space. To capture fine-grained geometry, such as clothing or hair, the neural implicit function has been used to reconstruct a free-form surface [14, 18, 60, 61, 69], but it is still challenging to recover detailed structure like fingers, face, or wrinkles on clothes. Another family of approaches extends the parametric model by predicting per-vertex displacements upon canonical body mesh [4–6, 34, 40, 67], which achieves a good balance between the expressiveness and prior regularization. Most related to us, CAPE [40] trains a generator to synthesize fine-grained geometry from a latent space, and can run in the auto-decoding mode for fitting. However, it is empirically not robust to work with temporal frames and sensitive to the errors in imperfect poses, which is not ideal to plug and play in our scenario.

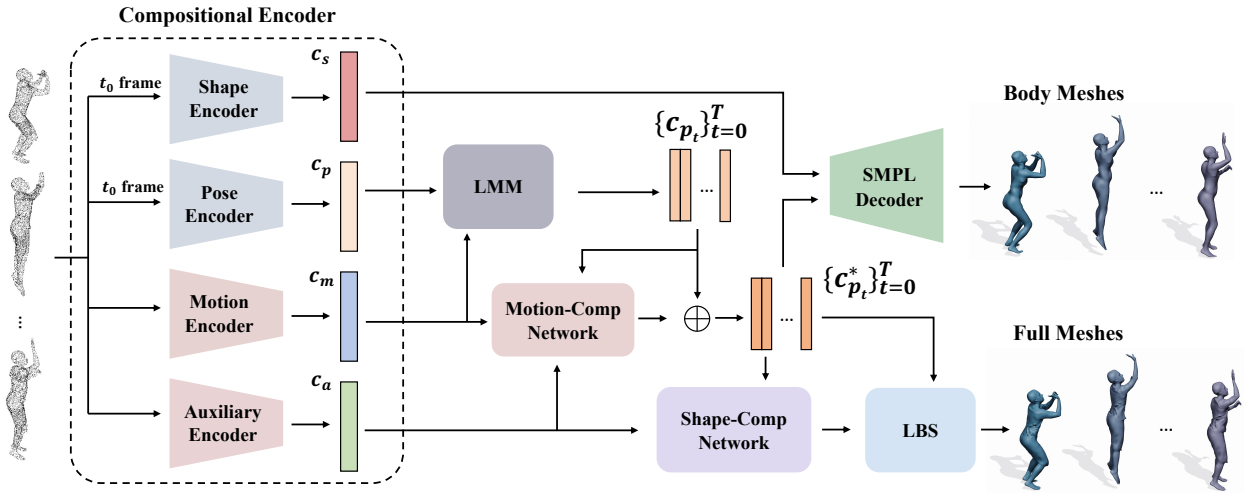


Figure 2. **Overview of our framework.** We learn the compositional representation for dynamic humans through 4D reconstruction. Specifically, given an input point cloud sequence, we first extract latent codes of shape, initial pose, motion and auxiliary information with the compositional encoder, and obtain rough motion estimation with the Linear Motion Model (LMM). Then we predict the residuals on temporal motion and the shape in canonical pose with the GRU-based Motion-Comp Network and Shape-Comp Network, respectively. Our method is able to output accurate body mesh sequence with the SMPL decoder [39] as well as full mesh sequence with surface geometry details, *e.g.* clothing and hair, by using Linear Blend Skinning (LBS). Detailed architectures can be found in Supp. Material (Sec. 1.1).

3. Method

This section introduces our compositional neural representation for dynamic human (H4D), which is learned through reconstruction task, and the whole framework is overviewed in Fig. 2. Given a 3D human model performing motion in a time span ($L = 30$ frames of meshes in our setup), we sample a point cloud of 8192 points from each as the input sequence to the network. Note that from a realistic perspective, we do not assume the temporal correspondences among frames (*e.g.* point trajectories) are available. In contrast, this is critically required by previous 4D representation methods [25, 48].

The input sequence is fed into a compositional encoder to extract latent codes representing shape, initial body pose, temporal motion, and auxiliary of additional compensation on motion and geometry (Sec. 3.1). To reconstruct the input temporal sequence, the shape, initial pose, and motion codes are combined through a pre-learned Linear Motion Model (LMM) to generate a rough estimation of per-frame 3D shapes represented as SMPL [39] (Sec. 3.2). Due to the limited capacity of LMM, the output, though plausible, demands additional refinements for accurate reconstruction. To this end, we feed the motion code, auxiliary code, and initial estimation to the GRU based Motion-Comp network (Sec. 3.3) and Shape-Comp network (Sec. 3.4) to predict the residual on temporal motion and the shape in canonical pose respectively. The final sequence is obtained by deforming the refined canonical shape using SMPL linear blending weight according to the refined motion sequence, *i.e.* per-frame poses.

3.1. Compositional Encoder

To keep the representation compositional, we train four separate encoders to extract latent code representing the shape c_s, initial pose c_p, motion c_m, and auxiliary information c_a respectively. The shape and pose encoders are implemented as PointNet-based [55] network with ResNet blocks, which take only the starting frame as input since it is sufficient to tell the canonical body shape and initial pose. On the other hand, the motion and auxiliary encoders take all the frames as input since temporal information is needed. To achieve that, we firstly encode the point cloud of each frame into a feature vector using a shallow PointNet, and then further aggregate per-frame feature with a GRU layer. The feature extractor is shared between motion and auxiliary encoders, and only GRUs are trained respectively. Note that our temporal encoders do not require as input the temporal correspondences, *i.e.* point trajectories, and thus can process sequences with unordered point clouds.

3.2. Linear Motion Model

We take the predicted c_p and c_m to reconstruct a coarse estimation of motion. To ensure robustness, we employ the parameter space of SMPL model and pre-learn a linear model for the motion. Each input temporal sequence can be represented as $\Phi = [\theta_1, \dots \theta_L], L = 30$, where $\theta_i \in \mathbb{R}^{72}$ is the SMPL pose parameter for frame i. We then represent motion as the per-frame difference of the pose parameter from the first frame, i.e. $\Psi = [\theta_2 - \theta_1, \dots, \theta_L - \theta_1] \in \mathbb{R}^{72(L-1)}$, and run a Principal Component Analysis (PCA) to reduce the dimension. The input motion now can be re-

constructed through the linear model:

$$\hat{\Phi} = [\theta_1, \alpha^T \cdot \mathcal{M} + \mu_\Psi + \theta_1] \quad (1)$$

where $\alpha \in \mathbb{R}^K$ is the coefficient of principal components, $\mathcal{M} = [\mathbf{M}_1, \ldots, \mathbf{M}_K] \in \mathbb{R}^{72(L-1)\times K}$ and μ_Ψ are the top K principal components and mean of the Ψ learned from the training data.

In practice, we found it more robust to run PCA separately for the global orientation (*i.e*. pelvis) and body joint rotation. We pick 4 basis for global orientation and 86 basis for body joint rotation, which explains 90% of the variance[1]. Finally, we plug the linear motion model into our pipeline, amenable to the output of compositional encoder,

$$\{c_{p_t}\}_{t=0}^{L-1} = [c_p, c_m^T \cdot \mathcal{M} + \mu_\Psi + c_p], \quad (2)$$

where c_{p_t} is the pose parameter for frame t.

3.3. Motion Compensation Network

The LMM is effective in representing motions with a relatively large number of temporal frames; unfortunately, it lacks the capacity to represent motion details. As a result, the predicted pose sequences are not accurate enough.

To improve the motion accuracy, we build a motion compensation network (Motion-Comp) to adjust the pose parameter of each frame. Specifically, we adopt a GRU-based network [15] as it was demonstrated to be effective for processing temporal information. We concatenate the motion code c_m and the auxiliary code c_a to the pose parameters of each frame from LMM prediction $\{c_{p_t}\}_{t=0}^{L-1}$, and then feed them sequentially into GRU to produce the residual of each frame. Once the per-frame pose parameters are updated with the Motion-Comp network outputs, we combine them with the shape parameter c_s from the encoder into the standard SMPL decoder to reconstruct the per-frame mesh. Overall, our motion model benefits from both the strong prior in the linear motion model and the impressive capacity from the motion compensation network.

3.4. Shape Compensation Network

So far, we are able to reconstruct the correct motion sequences, which can be further converted to body mesh sequences via SMPL decoder. However, the predicted shapes are still inferior, as many details such as hairs or clothes are missing. This is mostly due to the constrained capacity of SMPL shape space. To enhance the geometry, the shape representation presented in CAPE [40] is introduced: a per-vertex offset is estimated for the body mesh in the canonical space via a graph-based neural network conditioned on target pose. The added details would be then transferred to the target body pose via the pre-defined linear blending weight in SMPL. When combined into our framework, one straightforward way is to have the auxiliary code c_a encode shape details and feed it through the CAPE decoder for per-vertex offsets. We found this works reasonably well in feed-forward mode but not the back-propagation. We suspect this might because of the inconsistent gradients from different temporal frames, especially when the pose estimation is not perfectly accurate. As a result, the compensated geometry is vaguely correct (*e.g*. bump on the head for some hairstyle) but not precise. To improve the stability, we propose a shape compensation network (Shape-Comp), in which a GRU takes the auxiliary code c_a as input and predicts a new latent vector for each temporal frame conditioned on the predicted pose. The latent vector is then fed into the graph network to predict per-vertex offset, which is similar to the CAPE decoder. We remove the VAE and adversarial loss as they empirically hurt the performance. The GRU enables information exchanging across temporal frames, which is critical for robust back-propagation when running applications like motion completion and prediction.

3.5. Training Strategy

Neural networks with multiple stages are highly nonlinear and could easily fall in the local minimum. We advocate a stage-wise training strategy to enhance the training stability. Specifically, we first train the shape encoder, pose encoder, point feature extractor and motion encoder jointly with the pre-learned linear motion model. Once the model converges, we enable the Motion-Comp and Shape-Comp networks for the end-to-end joint training. Similar training strategy has been commonly used by other works, *e.g*. BCNet [24], XNect [44] and Predicting Human Dynamics [68].

Loss Functions Since our reconstructions are registered with SMPL topology, we use per-vertex L1-loss with the ground truth mesh as the loss. To further alleviate the ambiguity between body shape and clothing, we add an L2-loss on the shape code c_s with regard to the ground truth. During the first training stage, we use the mesh reconstructed with the motion from linear motion model for suervision. In the second stage, we use added loss on both meshes before and after the Shape-Comp network. The detailed loss functions can be found in Supp. Material.

4. Experiments

In this section, we perform extensive experiments to verify the efficacy of our method. First, we evaluate the capacity of our representation for encoding accurate shape and motion with the tasks of 4D reconstruction and human shape and motion recovery. We then demonstrate that a large variety of 4D related applications, including motion retargeting, completion, and prediction can be achieved with high quality with our representation. In the end, we provide an ablation study to test the impact of each component in our framework on the reconstruction quality.

[1] Please refer to Supp. Mat. for visualization of principal components

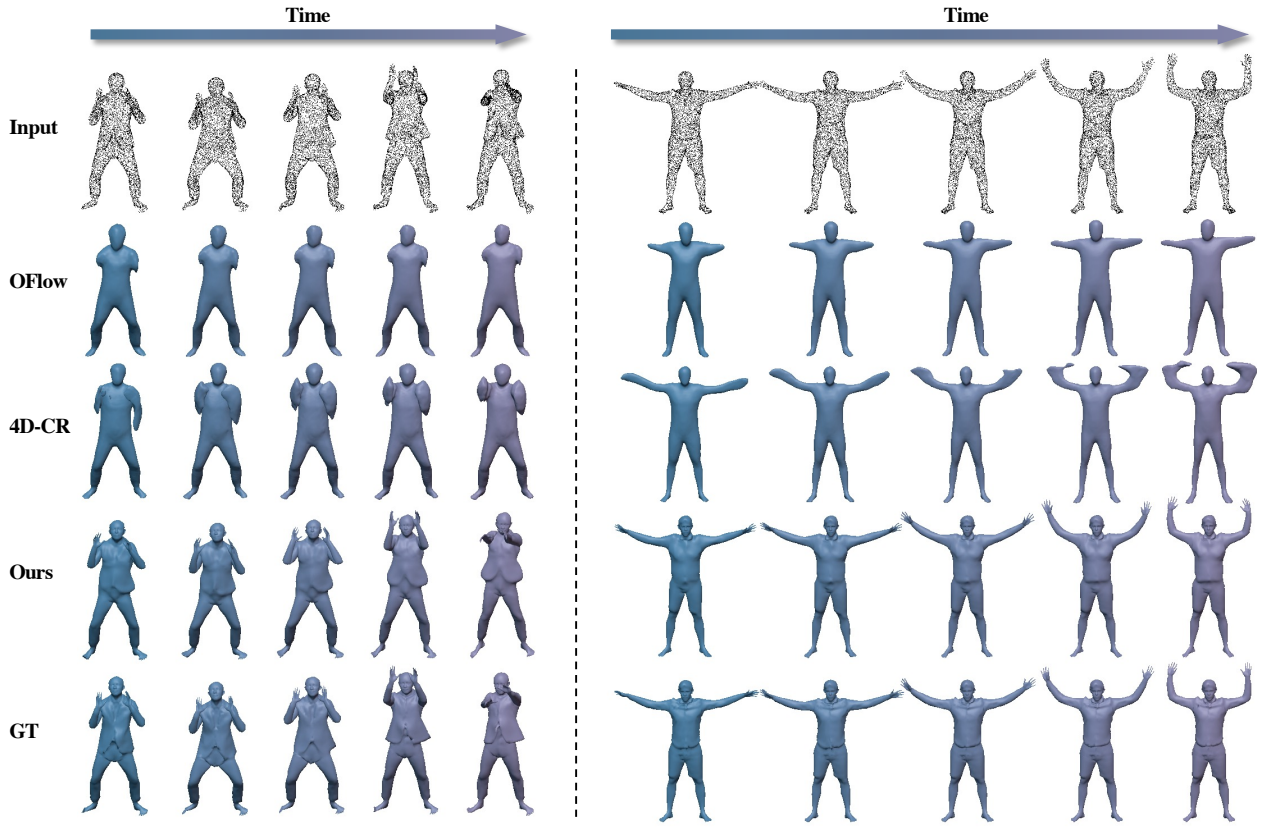


Figure 3. **4D Reconstruction.** Given the dense point cloud sequence (Row 1) uniformly sampled from the SMPL registered meshes, our method (Row 4) can reconstruct fine-grained meshes with accurate motion, while baseline methods (Row 2, 3) tend to overly smooth and often have incomplete geometry, *e.g.* missing hands. We uniformly sample 5 frames (out of 30 frames) for visualization.

Dataset We use the CAPE dataset [40, 53] for training and evaluating, which is a dataset of 3D dynamic clothed humans containing 10 male and 5 female subjects wearing different types of outfits. More than 600 motion sequences of large pose variations are provided. In each sequence, the clothed body shapes are captured at 60 FPS along with corresponding mesh in the canonical pose and pose parameter of each frame. Overall, the dataset provides good diversity on both 3D geometry and motion. Following OFlow [48], we divide all sequences in CAPE into sub-sequences with $L = 30$ frames. We use sub-sequences from 488 motion sequences for training, and randomly sampled 2000 sub-sequences from the other 123 motion sequences for testing.

Implementation We use PyTorch to implement the model, and train with the Adam optimizer [30]. In the first stage, the learning rate is 10^{-4} with batch size 16. In the second stage, the initial learning rate is set to 10^{-4} and dropped to 10^{-5} after 200K iterations with batch size 4 due to the limitation of GPU memory footprint. We use 4 GeForce RTX 2080Ti GPU cards. The Codes and Models will be released.

Evaluation To measure the difference between the prediction and ground truth 3D shape, we use Chamfer Distance (CD) and Volumetric IoU (IoU) [46] for free-form geometry and Per Vertex Error (PVE) for SMPL registered shape. To measure the accuracy of motion, we additionally use Procrustes-aligned mean per joint position error (PA-MPJPE), mean per joint position error (MPJPE), and acceleration error (mm/s^2) computed on 45 keypoints which include 24 SMPL joints and 21 keypoints on the face, feet and hands. Please refer to [29, 31, 46] for more details about these standard metrics. For temporal sequences, we take the mean of all the frames as the final score.

4.1. Representation Capability

We first show that our representation is capable of encoding and reconstructing human sequences with correct motion and geometry.

4D Reconstruction We compare to state-of-the-art 4D representation, Occupancy Flow (OFlow) [48] and 4D-CR [25], on mesh reconstruction from sampled point cloud inputs. As shown in Tab. 1 (I), our method significantly outperforms other methods on the 4D reconstruction accuracy. Qualitative results are shown in Fig. 3 for two temporal sequences. OFlow tends to produce incomplete geometry with missing hand, and results from 4D-CR are

I. Comparison with Previous 4D Representation Methods								
Methods	4D Reconstruction		Motion Retargeting		Motion Completion		Future Prediction	
	IoU ↑	CD ↓	IoU ↑	CD ↓	IoU ↑	CD ↓	IoU ↑	CD ↓
OFlow [48]	61.5%	0.199	30.7%	0.470	65.8%	0.181	58.8%	0.218
4D-CR [25]	62.9%	0.165	47.3%	0.296	76.6%	0.128	64.0%	0.200
Ours	**73.3%**	**0.093**	**70.7%**	**0.100**	**90.3%**	**0.031**	**71.7%**	**0.121**

II. Comparison with Human Body Estimation Methods (Forward)								
Methods	Shape and Motion Recovery				Motion Retargeting			
	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓
HMMR [29]	87.8	102.1	89.2	20.9	85.7	98.0	86.9	19.4
VIBE [31]	45.3	54.3	47.6	13.4	46.3	54.1	47.0	12.8
4D-CR-SMPL [25]	59.2	68.5	59.5	9.9	62.4	73.2	63.7	10.1
4D-CR-SMPL* [25]	49.8	57.7	49.8	8.9	52.2	59.6	51.6	8.7
Ours	**38.4**	**44.9**	**39.2**	**8.8**	**39.5**	**45.2**	**39.0**	**8.6**

III. Comparison with Human Body Estimation Methods (Backward)								
Methods	Motion Completion				Future Prediction			
	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓
HMMR [29]	146.5	141.6	148.3	48.7	148.4	142.9	146.9	48.3
Zhang *et al*. [68]	–	–	–	–	134.7	146.5	143.4	23.0
4D-CR-SMPL [25]	87.3	67.3	66.9	14.1	91.9	77.9	77.1	11.3
Ours	**53.8**	**42.7**	**41.7**	**9.4**	**73.1**	**62.8**	**59.7**	**11.2**

Table 1. **Comparison to SoTA methods on various tasks.** For evaluation, we adopt Volumetric IoU (IoU) and Chamfer Distance (CD) [46] for comparisons with free-form methods (block I), and several standard metrics following [29, 31] for SMPL-based methods (block II&III, the numbers are measured in *mm*). * denotes the input point cloud sequence has temporal correspondence.

overly smoothed, *e.g*. around face and hand. On the contrary, thanks to the human prior, our results are significantly better than others with complete geometry, correct motions, and rich details such as fingers, clothes, and hair. It is also worth noting that both OFlow and 4D-CR require point clouds with temporal correspondence as input, whereas our method can take unregistered point cloud sequences which is more convenient for many applications.

Shape and Motion Recovery We then study the performance of our motion model, which consists of a linear motion model and per-frame compensations recovered from the auxiliary code. As baselines, we compare to SoTA video-based human shape and pose estimation methods HMMR [29] and VIBE [31]. Originally designed for color images inputs, we replace their image encoder with our point cloud encoder for our setup. Moreover, as an additional baseline, we extend 4D-CR [25] to an SMPL-based version by replacing their implicit occupancy decoder with the SMPL decoder so that it also benefits from the human prior. Since all of these methods produce only SMPL defined shapes, *i.e*. minimally-clothed without hair or cloth details, we disable our Shape-Comp network and use the output from the SMPL decoder for fair comparisons. All the baseline methods are retrained on our dataset. For extending 4D-CR, we train two models with registered point clouds as in their work (4D-CR-SMPL*) and unregistered point clouds like us (4D-CR-SMPL) respectively.

The quantitative comparisons are shown in Tab. 1 (II). Our method achieves more accurate motion estimations (as measured at body keypoints by PA-MPJPE, MPJPE, acceleration error) and SMPL shape (as measured by PVE) than HMMR and VIBE. 4D-CR-SMPL performs relatively poorly when the input point cloud is unordered and gets much better once given tracked point clouds (4D-CR-SMPL*) but still performs worse than our method.

4.2. Applications

Our representation can support various applications. Note that for all applications, the encoder and decoder are both fixed after training.

Motion Retargeting The goal of motion retargeting is to transfer the motion sequence from one subject to another. Traditional methods typically require manual works, *e.g*. provide correspondences between source and target identities [63], to fulfill such a task.

We achieve motion retargeting without any human intervention. Taking two point cloud sequences, one as the identity (I) and the other as the motion (M), we feed both into our compositional encoder to get latent codes for each $(c_s^I, c_p^I, c_m^I, c_a^I)$ and $(c_s^M, c_p^M, c_m^M, c_a^M)$. We then conduct the motion retargeting by using (c_s^I, c_p^M, c_m^M) for linear motion model, c_a^M for Motion-Comp network, and

c_a^I for Shape-Comp network. Note that two c_a are used for Motion-Comp and Shape-Comp networks separately as they encode motion and shape information respectively.

For evaluation purpose, we randomly sampled 100 pairs of identity and motion sequences with $L = 30$ frames. We use the provided ground truth shape in canonical pose and pose parameters provided by CAPE [40] dataset to generate motion retargeted ground truth sequences. We compare to free-form geometry based approach (OFlow and 4D-CR) on full geometry (in Tab. 1 (I)) and SMPL based approach (HMMR, VIBE, 4D-CR-SMPL) on the minimally-clothed mesh from SMPL (in Tab. 1 (II)). Our method significantly outperforms OFlow and 4D-CR. As shown in a qualitative example in Fig. 4, our method produces much more complete motion retargeting results. Note how clothing details are successfully transferred, *e.g.* long trousers in the identity sequence compared to shorts in the motion sequence. Our method also outperforms all the human body estimation methods, showing that our compositional encoder is more effective in extracting correct information from inputs.

Motion Completion Our representation can also fulfill fitting tasks in the auto-decoding fashion, in which the latent codes are optimized to produce output similar to the observation. With this, our representation can perform motion completion, where the goal is to predict the missing data in a dynamic human sequence. For evaluation, we randomly choose 100 sequences with 30 frames from our test set. For each sequence, we randomly pick 15 frames as the observation, optimize the latent code, reconstruct the full sequence, and then measure the geometry accuracy on the other 15 frames. Note that we use Chamfer loss on uniformly sampled points instead of PVE to simulate the case in real application, where the observed meshes may not be registered.

Comparisons to free-form based methods and SMPL based methods are shown in Tab. 1 (I) and (III) respectively, and qualitative results are in Supp. Material. Zhang *et al*. [68] uses a similar motion model with HMMR [29], so we only evaluate one of them. Overall, our method consistently outperforms all the other methods.

Moreover, we compare the robustness of auto-decoding based fitting using our Shape-Comp network against the naive CAPE decoder, and show the error of completion w.r.t the amount of random noises added to the observed frames in Fig. 6 (b). The error from our model is consistently lower than the naive CAPE decoder and deteriorates less with increasing noise. This is presumably because CAPE performs per-frame optimization, which may confuse the latent space if gradients are not consistent from temporal frames, whereas we use GRU to model the temporal sequence for more robustness.

Last but not least, our model can also complete the temporal sequences from partial spatial observation. To show this, we generate one depth image per-frame from a camera rotating concurrently with the motion of the 3D shape, and run auto-decoding based fitting to complete the sequence. This can be also considered as a typical non-rigid fusion with known camera poses. We show the qualitative results and quantitative comparison in Supp. Material.

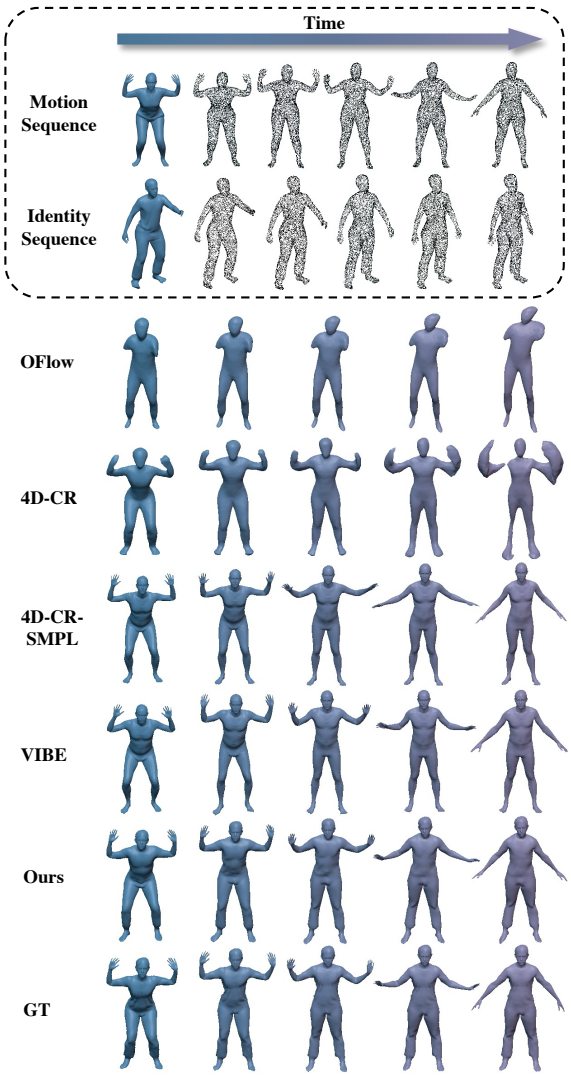


Figure 4. **Motion Retargeting.** Our goal is to transfer the human movements of the motion sequence (Row 1) to the people in the identity sequence (Row 2). We can accurately transfer the motion to the new identity and keep the original geometry details, *e.g.* garments and hairstyle, at the same time (Row 7). The free-form baselines (Row 3, 4) either fail due to shape and motion entanglement or produce more artifacts over time due to error accumulation. The SMPL-based baselines (Row 5, 6) also fulfill retargeting but are not as accurate as us, and they can only represent minimally-clothed body.

Future Prediction Our representation also supports future prediction. Specifically, we run fitting algorithm with the first 20 frames, generate the latent codes, and then reconstruct full sequence to predict the 10 frames in the future.

Tab. 1 (I, III) and Fig. 5 show the comparison to the

previous methods. Again, we obtain significantly better performance than other 4D representation methods (OFlow and 4D-CR). When comparing only the motion accuracy using SMPL mesh with previous work on motion prediction (HMMR [29], Zhang *et al*. [68] and 4D-CR-SMPL [25]), our method still achieves better performance. Moreover, we empirically found that, though given pose prior term during backward optimization, these baseline methods are more easily to produce unnatural poses than us, and predict unreasonable motions as shown in Fig. 5, possibly because our PCA-based motion model provides regularization and global context for output motions.

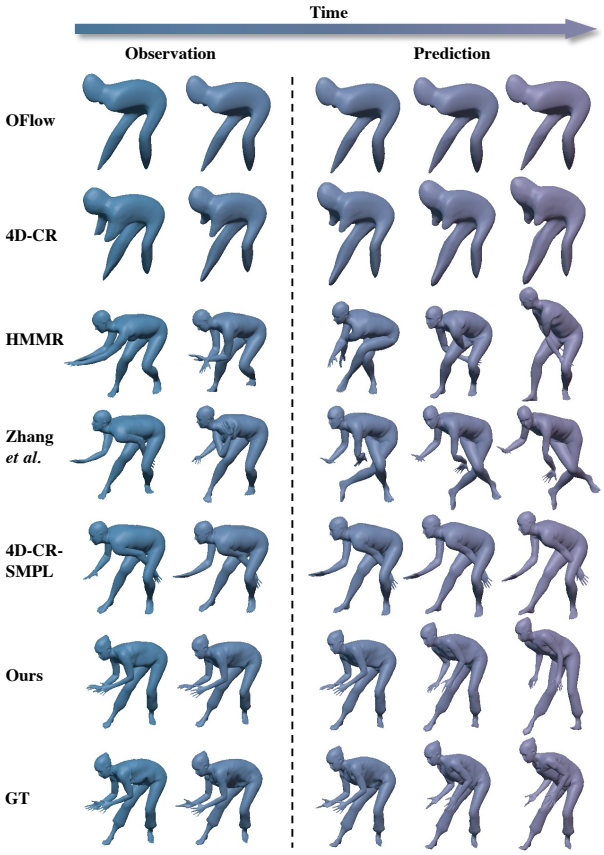


Figure 5. **Future Prediction.** We are aiming to extrapolate 10 future temporal frames based on 20 past observed frames. The baseline methods (Row 1-5) either produce unsatisfactory geometry or stuck into the unnatural pose while our method (Row 6) successfully keeps the movement trend and fulfills the reasonable prediction of future motion. The meshes on the left are reconstruction results of the observations, and the meshes on the right are the predictions for future time steps.

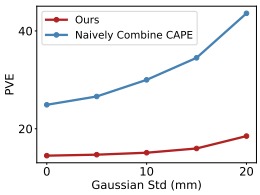
4.3. Ablation Study

In this section, we perform an ablation study to demonstrate the effect of the major designs in our method.

Motion Model We first study the effect of the linear motion model and auxiliary code for motion recovery. We compare the ablation cases on the output of the SMPL decoder with the registered SMPL model on the ground truth mesh, which removes the free-form deformation and allows us to focus on the motion quality. In Fig. 6 (a), we show the performance of our model removing linear motion model ("-LMM"), which directly updates the initial pose code, or removing Motion-Comp network ("-Motion c_a"), which only relies on linear motion model (LMM) for motion recovery. In either case, the motion accuracy drops consistently as measured by all the metrics, indicating the necessity of combining prior model with learned compensations.

	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓
- GRU Enc.	49.6	57.0	49.6	10.6
- LMM	40.2	46.8	41.2	**8.8**
- Motion c_a	43.7	50.4	43.4	9.0
Full Model	**38.4**	**44.9**	**39.2**	**8.8**
- Shape c_a †	–	–	43.8	–
Full Model†	–	–	**42.0**	–

(a) Ablation Study



(b) Noise Tolerance

Figure 6. (a) **Ablation study.** We verify the effectiveness of our temporal encoder by replacing the GRU with the modified PointNet used in [25, 48] (Row 1). Additionally, we remove the major modules in our framework in turn to demonstrate the effect of different components (Row 2, 3, 5). † denotes we compute metric with the ground truth clothed mesh. (b) **Robustness against noise.** The x-axis is the standard deviation of added Gaussian noise, and the y-axis is Per Vetex Error (PVE, lower is better).

Shape Model We then verify if the auxiliary code helps to recover detailed geometry. In Fig. 6 (a), we show the performance of the final mesh without and with auxiliary code-driven shape compensation (the last two rows), which is measured by PVE between the output mesh with the ground truth clothed mesh. The advantage of the shape compensation can also be found in Fig. 4 and 5, which show that the auxiliary code helps to improve the geometry details when comparing our results with the SMPL outputs from VIBE, HMMR or 4D-CR-SMPL.

Encoder Last but not least, we verify the effectiveness of our GRU-based temporal encoder. We replace our temporal encoder with the PointNet adopted in OFlow [48] and 4D-CR [25], and see a significant performance drop ("-GRU Enc."), which shows our GRU-based encoder helps to extract temporal information from the point cloud sequences without temporal correspondence.

5. Conclusion

This paper introduces H4D, a compact and compositional neural representation for 4D human captures, which combines the merits of both the prior-based and free-form solutions. A novel GRU-based framework is designed for learning our representation, which encodes the input point cloud sequences into the latent codes of shape, initial pose, motion and auxiliary information. Extensive experiments on 4D reconstruction, shape and motion recovery, motion retargeting/completion and future prediction validate the efficacy of the proposed approach.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Representation learning and adversarial generation of 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2(3):4, 2017. 2

[2] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. *arXiv e-prints*, pages arXiv–2004, 2020. 2

[3] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7144–7153, 2019. 2

[4] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1186, 2019. 2

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018. 2

[6] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019. 2, 16

[7] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018. 2

[8] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, aug 2020. 13, 14

[9] 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 *European conference on computer vision*, pages 561–578. Springer, 2016. 1, 2

[10] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *European Conference on Computer Vision*, pages 226–242. Springer, 2020. 2

[11] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[12] ChaoWen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *ICCV*, 2019. 2

[13] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018. 1, 2, 14, 16

[14] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2, 4, 12

[16] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[17] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 2

[18] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, 2021. 1, 2

[19] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2surf: Learning implicit surfaces from point clouds. In *ECCV*, 2020. 2

[20] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[21] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[22] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018. 2

[23] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10884–10894, 2019. 1, 2

[24] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *European Conference on Computer Vision*, pages 18–35. Springer, 2020. 4

[25] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5340–5350, 2021. 1, 2, 3, 5, 6, 8, 13, 16

[26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[27] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and

pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 12

[28] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[29] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019. 1, 2, 5, 6, 7, 8, 13

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 13

[31] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 1, 2, 5, 6, 12

[32] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019. 1, 2, 12

[33] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6050–6059, 2017. 1, 2

[34] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *2019 International Conference on 3D Vision (3DV)*, pages 643–653. IEEE, 2019. 2, 16

[35] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 2

[36] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2019. 16

[37] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 16

[38] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020. 16

[39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1, 2, 3

[40] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 5, 7, 13, 14, 16

[41] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019. 14

[42] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9489–9497, 2019. 2

[43] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 2

[44] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1907.00837*, 2019. 4

[45] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018. 1, 2

[46] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5, 6, 13

[47] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 16

[48] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019. 1, 2, 3, 5, 6, 8, 12, 16

[49] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12695–12705, 2021. 14

[50] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[51] 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. 2

[52] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. 1, 2

[53] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):1–15, 2017. 5

[54] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, 2018. 2

[55] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2, 3, 12

[56] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 13

[57] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11488–11499, 2021. 14

[58] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[59] 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. 2

[60] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[61] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 2

[62] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16094–16104, 2021. 1

[63] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 6

[64] Raquel Urtasun, David J Fleet, and Pascal Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer vision and image understanding*, 104(2-3):157–177, 2006. 12

[65] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2

[66] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017. 2

[67] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5908–5917, 2019. 2

[68] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7114–7123, 2019. 1, 2, 4, 6, 7, 8

[69] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

Supplementary Material

In this supplementary material, we provide implementation details, results about the generalization ability of our motion model, additional quantitative comparisons, visualization of principal components, additional qualitative results, run-time comparison, and discussions about limitations, future work and broader impact of our approach.

1. Implementation Details

In this section, we first provide network architectures used for the compositional encoder, Motion-Comp and Shape-Comp networks in our framework. Next, we explain the strategy of choosing the number of principal components for our linear motion model. Finally, we discuss more details in our experiments.

1.1. Network Architecture

Compositional Encoder Both the shape encoder and the initial pose encoder take point cloud of the first time step as input, and we adopted the same architecture as the spatial encoder in Occupancy Flow (OFlow) [48]. The network is a variation of PointNet [55] which has five residual blocks as show in Fig. 7a. Each of the first four blocks has an additional max-pooling operation to obtain aggregated feature of size $(B, 1, C)$ where C denotes the dimension of hidden layers, and an expansion operation (repeat the pooled feature to the size (B, N, C)) to make it suitable for concatenation. The output of the fifth block is passed through a max-pooling layer and a fully connected layer to get the final latent vector of dimension 10 for shape code and 72 for initial pose code.

Our temporal encoder, for the purpose of learning motion and auxiliary codes, is composed of a point feature extractor (shallow PointNet) and a double layers GRU [15], as shown in Fig. 7b. The shallow PointNet extracts spatial features for each input point cloud, which has 3 hidden layers with hidden sizes equal to 128. We use the same max-pooling and concatenating operations as the spatial encoder. Then the per-frame features are processed sequentially by the GRU layer to provide the latent vector of dimension 90 for motion code and 128 for auxiliary code.

Motion-Comp Network We design a conditional GRU for our Motion-Comp network to learn the compensation of the input motion sequence. Specifically, we use the motion code c_m and auxiliary code c_a as conditions, copy and concatenate them with the pose parameter of each time step estimated by our linear motion model. The detailed architecture is shown in Fig. 7c. The output of the conditional GRU is the motion compensation, and we apply a residual connection to obtain the refined motion sequence, *i.e.* per-frame poses. We can recover body mesh sequences with the predicted shape and per-frame pose codes by using SMPL decoder, here we use the neutral shape model as in previous work [27, 31, 32].

Shape-Comp Network We propose a Shape-Comp network, in which a conditional GRU takes the auxiliary code c_a as input and predicts a new latent vector for each temporal frame conditioned on the predicted pose (we follow CAPE to represent each joint with the flattened rotational matrix and filter the joints that are not related to clothing). The latent vector of each frame is then fed into the graph network to predict per-vertex offsets, which is similar to the CAPE decoder. We remove the one-hot vector of clothing type and only use the predicted pose as condition since we do not focus on the generative task. The architecture is shown in Fig. 7d.

Implementation of GRUs We use the standard API of GRU provided in PyTorch. All the GRUs in our framework share the same architecture, which has 2 layers with the hidden size of 512, except we apply an additional linear layer for each GRU to transform the output dimensions for different modules.

1.2. Linear Motion Model

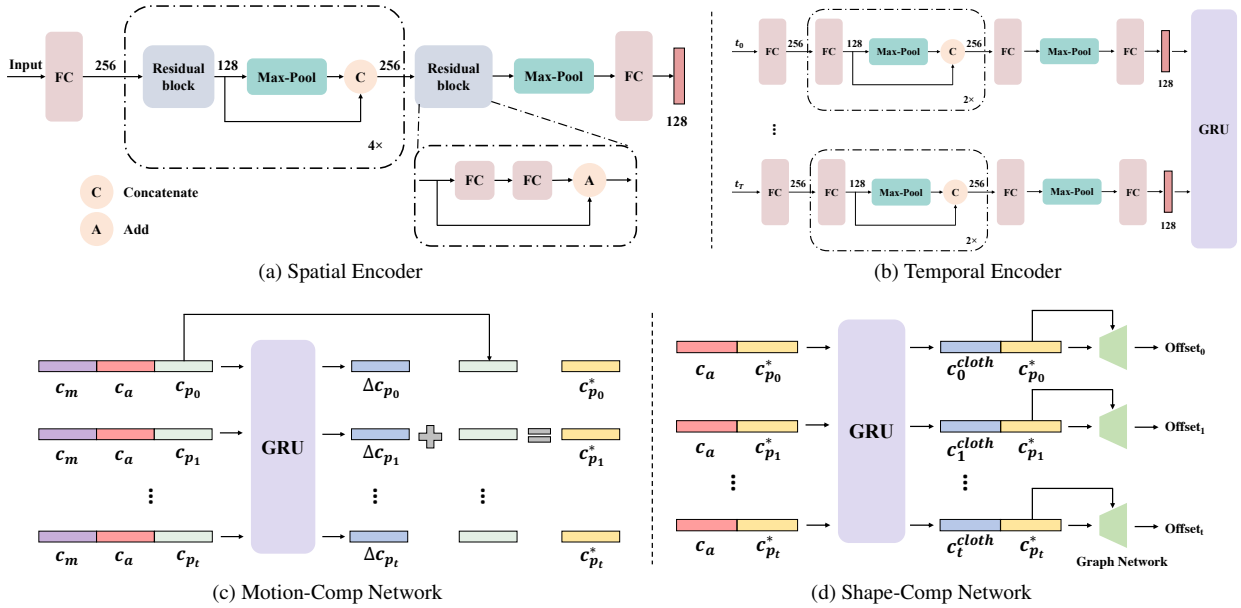
For the linear motion model (Section 3.2 in the main paper), we employ the Principal Component Analysis (PCA) to model the per-frame difference of the pose parameter with regard to the first frame in a sequence. As stated in Sec. 3.2 of the main paper, we run PCA separately for the global orientation (*i.e.* pelvis) and the remaining body joint rotations. Inspired by Urtasun *et al.* [64], we choose the number of PCA components depending on the fraction of the total variance of the training data that is captured by the subspace, denoted by $Q(m)$:

$$Q(m) = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \tag{3}$$

where m controls the number of principal components, λ_i are ordered eigenvalues of the data covariance matrix such that $\lambda_i \geq \lambda_{i+1}$, and M is the total number of eigenvalues. In our experiments, we choose $m = 4$ for the global rotation and $m = 86$ for the remaining body joints rotation, which satisfy $Q(m) > 0.9$. We visualize some principal components in Fig. 9 and 10 (Sec. 4).

1.3. Experiment Details

Loss Functions Given an input point cloud sequence, our model generates one shape code c_s and three mesh sequences $\mathbf{X}_{\text{linear}}$, $\mathbf{X}_{\text{motion}}$ and $\mathbf{X}_{\text{shape}}$, which correspond to the outputs of LMM, Motion-Comp network and Shape-Comp network respectively. Each sequence has $L = 30$ mesh frames and each mesh has $K = 6890$ vertices. We also have the ground truth shape parameter c_s^* and posed SMPL body mesh sequence $\mathbf{Y}_{\text{body}}$. Furthermore, we compute ground



(a) Spatial Encoder (b) Temporal Encoder

(c) Motion-Comp Network (d) Shape-Comp Network

Figure 7. **Detailed network architectures in our framework.**

truth offsets sequence by $\mathbf{Y}_{\text{offset}} = \mathcal{M}_{\text{clothed}} - \mathcal{M}_{\text{SMPL}}$, where $\mathcal{M}_{\text{clothed}}$ and $\mathcal{M}_{\text{SMPL}}$ stand for the vertices of the clothed human mesh and corresponding SMPL body mesh in the canonical pose respectively.

Then we define the reconstruction loss as the per-vertex L_1 error with the ground truth mesh

$$\mathcal{L}_{\text{r}}\left(\mathbf{X}, \mathbf{Y}\right) = \frac{1}{LK} \sum_{l=1}^{L} \sum_{k=1}^{K} \left\| \mathbf{X}_{l,k} - \mathbf{Y}_{l,k} \right\|_1 . \tag{4}$$

Furthermore, we apply L_2 penalization on the predicted shape code to further alleviate the ambiguity between body shape and clothing, given by

$$\mathcal{L}_{\text{s}}\left(\mathbf{c}, \mathbf{c}^*\right) = \left\| \mathbf{c} - \mathbf{c}^* \right\|_2^2 . \tag{5}$$

Finally, the total loss for training can be formulated as

$$\begin{aligned} \mathcal{L} = \lambda_s \mathcal{L}_{\text{s}}\left(\mathbf{c_s}, \mathbf{c_s^*}\right) &+ \lambda_{r_1} \mathcal{L}_{\text{r}}\left(\mathbf{X}_{\text{linear}}, \mathbf{Y}_{\text{body}}\right) \\ &+ \lambda_{r_2} \mathcal{L}_{\text{r}}\left(\mathbf{X}_{\text{motion}}, \mathbf{Y}_{\text{body}}\right) \\ &+ \lambda_{r_3} \mathcal{L}_{\text{r}}\left(\mathbf{X}_{\text{shape}}, \mathbf{Y}_{\text{offset}}\right) , \end{aligned} \tag{6}$$

we set $\lambda_s = \lambda_{r_1} = 1$, $\lambda_{r_2} = \lambda_{r_3} = 0$ for the first training stage and $\lambda_s = \lambda_{r_2} = 1$, $\lambda_{r_3} = 30$ and $\lambda_{r_1} = 0$ for the second stage.

Backward Experiments For our auto-decoding based experiments, *i.e.* completion and prediction, we use the trained model to perform a backward fitting algorithm. Specifically, we remove the encoder, freeze the parameters of the remaining modules and optimize the latent codes with back-propagation to produce the outputs as similar to the observations as possible.

We initialize the latent code with random vector sampled from $N(0, 0.01)$ and use the Adam optimizer [30] with learning rate $3e^{-2}$ to perform back-propagation for 500 iterations. In each iteration, we uniformly sample 8192 points on the surfaces of the predicted meshes, and compute Chamfer loss [46, 56] w.r.t the observed points for penalizing. Additionally, we follow IPNet [8] to add pose and shape prior terms, which penalize unnatural output bodies during optimization.

Completion We conduct two different types of motion completion experiments, *i.e.* temporal completion and spatial completion. Given a temporal sequence of $L = 30$ frames, for temporal completion, we randomly select 15 frames as observation and optimize the latent codes to complete the missing frames. We choose HMMR [29] and 4D-CR-SMPL (an extension we implement for 4D-CR [25]) as baselines. To implement 4D-CR-SMPL, we replace their implicit decoder with the SMPL decoder, and set the dimensions of their identity code and initial pose code to 10 and 72 respectively. Then we can obtain the pose code for each time step with the Neural ODE conditioned on the motion code, and input it to the SMPL decoder with the identity code to produce the reconstructed mesh frame. For 4D-CR-SMPL, we use the model trained on our dataset, and for HMMR, we use the official pre-trained model.

The goal of spatial completion is to complete the temporal sequence with partial spatial observation. To this end, we use the raw scanned mesh sequences of CAPE [40], and render the depth images of resolution 512×512 with the approach illustrated in the main paper (Sec. 4.2) to simulate the real world scenario. We assume the camera poses

are known and back project the depth images to obtain partial point clouds. We initialize our codes with the random vectors sampled from $N(0, 0.01)$ with no requirement for an additional initialization step like NPMs [49], and use the Adam optimizer with learning rate $3e^{-2}$ to perform back-propagation for 500 iterations. Note that in this experiment, we adopt one-directional point-to-surface loss instead of two-directional Chamfer loss due to the partial geometry. We show some qualitative examples in Fig. 16.

2. Generalization of Our Motion Model

In this section, our goal is to investigate the capacity of our motion model for representing novel motions from another dataset. To this end, we choose some motion sequences from AMASS [41], a large 3D MoCap dataset. And then we use our model trained on the CAPE dataset [40], perform the similar backward algorithm in completion and prediction experiments to fit the whole sequence of $L = 30$. Instead of dense SMPL, we randomly sample 8192 points from the SMPL mesh of each frame as observations, then use the Chamfer loss to the points sampled from the predicted mesh. Prior terms [8] are also used to penalize unnatural output. Since AMASS only provides SMPL parameters, we disable the Shape-Comp network and use the results from Motion-Comp network for visualization. Fig. 8 shows that the proposed method successfully reconstruct the full sequence from such sparse input, which demonstrates the generalization capability of our model to represent novel motions from another data source.

3. Additional Quantitative Comparisons

Comparison to HuMoR We compare with a SoTA human body estimation method HuMoR [57] on the task of fitting point cloud sequences. Specifically, we choose 100 mesh sequences of 30 frames from our test set and randomly sample 8192 points from each frame. Then we conduct optimization with 2 choices of loss functions, *i.e*. Chamfer loss (HuMoR, Ours) or 3D keypoints loss (HuMoR*, Ours*). For both losses, we also enabled prior loss to regularize the shape and motion. As shown in Tab. 2, our method beats HuMoR in both cases.

	PA-MPJPE ↓	MPJPE ↓	PVE ↓	Accel ↓
HuMoR	70.7	46.0	45.4	10.5
Ours	32.6	30.0	27.6	4.9
HuMoR*	25.7	27.1	26.1	7.3
Ours*	**16.2**	**14.5**	**11.4**	**4.5**

Table 2. **Quantitative comparisons with HuMoR.**

Comparison to NPMs We provide the comparisons to NPMs [49] on depth completion task. We choose 100 sequences and use the pretrained model of NPMs to perform completion from partial depth. Specifically, given a depth image sequence of 30 frames, we project the depth values into a 256^3-SDF grid to generate the inputs for NPMs, and then optimize the latent codes frame-by-frame with the default setups. Note that NPMs runs 10 times slower than H4D and uses twice of the memory. The Tab. 3 shows that our model outperforms NPMs, either w/ or w/o (NPMs*) encoders for code initialization, on both metrics.

	IoU ↑	CD ↓
NPMs*	79.3%	0.104
NPMs	85.5%	0.042
Ours	**87.7%**	**0.037**

Table 3. **Quantitative comparisons with NPMs.**

4. Visualization of Principal Components

The linear motion model in our framework totally has 90 principal components, the first 4 components are for global rotation (pelvis joint) and the rest 86 for other body joints. We start from the same rest pose and visualize some principal components in Fig. 9 and 10. Specifically, for each shown component, we select different scaling factors (before each row) and multiply them with this component to show the motion results. As shown, PC0 roughly controls the global rotation around the vertical axis; PC4 and PC6 affect the opening and closing of the upper arm and forearm respectively; PC7 is related to the bending of the legs; and PC9 tells the movement of arms and legs at the same time. In general, positive and negative scaling factors of components correspond to opposite directions of motion, and the absolute value affects the magnitude of the motion.

5. Additional Qualitative Results

We show the additional qualitative examples on 4D reconstruction in Fig. 11, shape and motion recovery in Fig. 13, temporal completion in Fig. 14 and 15, spatial completion in Fig. 16, future prediction in Fig. 12 and motion retargeting in Fig. 17.

6. Run-time

In Tab. 4, we show the per sequence run-time of our method and previous 4D representation methods on forward inference for 4D reconstruction and backward optimization for temporal completion. Note that we report the time cost to run a complete backward optimization process for a sequence (500 iterations). The length of the full sequence is $L = 30$, and all models run on a single NVIDIA 2080Ti GPU. Instead of the Neural ODE [13], we model the human

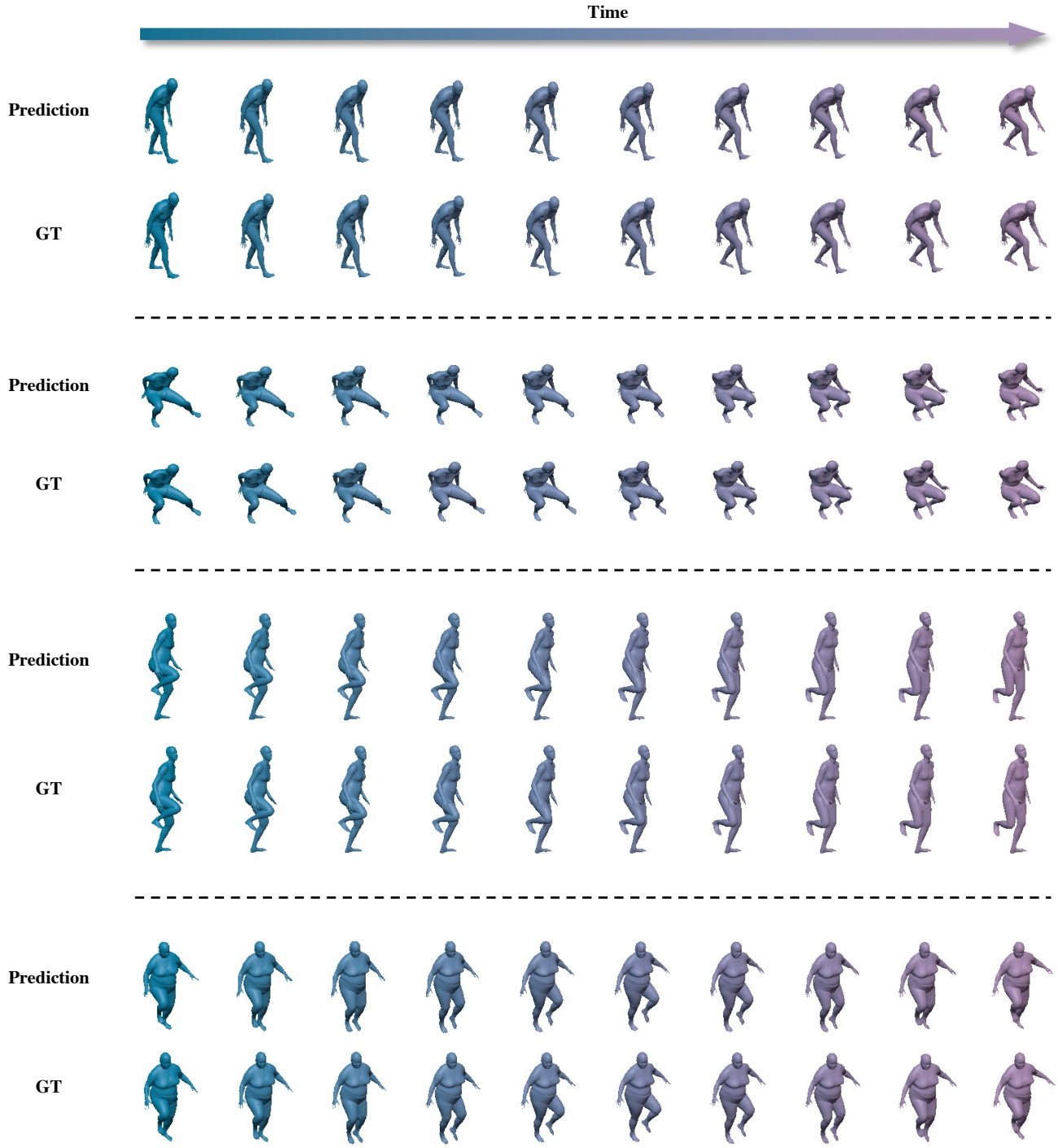


Figure 8. **Results of the novel motions from AMASS dataset.** To investigate the generalization ability of our method, we choose 4 motion sequences from the AMASS dataset, and use our model trained on the CAPE dataset to fit them by using the backward algorithm.

motion using the linear model and GRU-based compensation networks. As can be seen, our model runs faster in both cases, especially in backward optimization.

7. Limitations and Future work

We now discuss a few limitations of our approach that point to future work. **First**, our motion model can recon-

	Forward (s)	Backward (min)
OFlow [48]	1.106 (0.814)	17.600
4D-CR [25]	14.469 (5.861)	14.117
4D-CR-SMPL [25]	0.209	16.817
Ours	**0.175**	**7.303**

Table 4. **Comparisons about the run-time.** We show per sequence run-time of our method and baselines on forward inference and backward optimization. The numbers in the parentheses mean time without Marching Cubes.

struct discretized frame w.r.t each input time step, but not the arbitrary time in the continuous whole time span like 4D-CR [25] or OFlow [48], which will be useful in some scenarios requesting higher temporal resolution from inputs. Incorporating a network that takes time value scalar as input, *e.g*. Neural ODE [13], temporal MLP, would be a solution. **Second**, we currently conduct all the experiments on the sequences of 3D data, *e.g*. point clouds or meshes. On the one hand, this is due to the lack of 4D human datasets with color images, *e.g*. pairs of video and 3D human sequences (with clothing and hair). And on the other hand, the focus of this work is to propose a compositional representation and effectively power various 4D human related applications based on point cloud. Combining our representation with techniques such as neural rendering [37, 47] or photometric-based optimization [36, 38] for image-based full human 4D reconstruction would be a promising future direction. **Third**, we adopt the same clothing representation used in previous work [6, 34, 40], *i.e*. per-vertex offsets upon the body in canonical pose, and extend it to apply to temporal sequences. However, as discussed in CAPE [40], some loose garments such as skirts and coats are difficult to represent with offsets due to the limited capacity. Modeling clothes and hair as separate layers from the body with meshes or implicit surface is a feasible way, and we leave it to future work. **Forth**, since we have a compact motion representation that uses one single motion code to provide global control upon the whole sequence, future works also include high-level inference applications such as using the motion code learned in an unsupervised fashion to perform action classification with a simple linear classifier.

8. Broader Impact and Social Impact

Learning a compact representation for 3D data is a widely interested problem. However, less attention has focused on the 4D cases, though it is important for various applications to understand time-varying objects, *e.g*. Robotics, VR/AR. This work focuses on 4D human modeling and proposed H4D, a compact and compositional representation, which uses low-dimensional latent codes to encode key factors of dynamic humans. We make some attempts and demonstrate our representation has rich capacity and is amenable to many applications. We hope these explorations could provide insights for future research directions. For instance, using our representation for video-based full human reconstruction; exploiting the compositional property to control the outputs for generative tasks; and improving the 4D human representation and make up for the discussed limitations of our method. Broadly, our approach can serve as an important core tech in achieving the Metaverse. It may enable everyone to produce their own Avatar with their motions, potentially benefiting the Social Welfare.

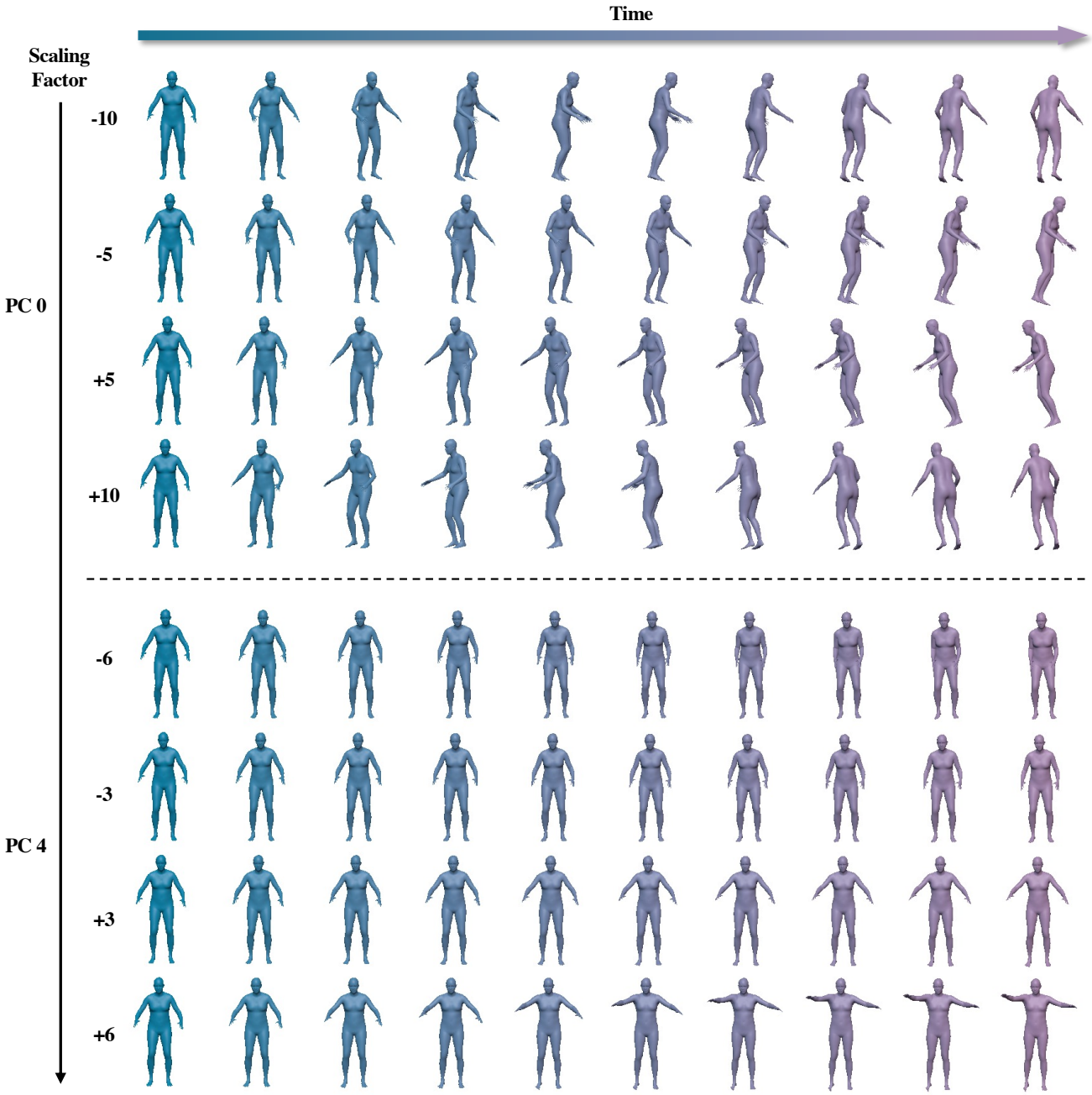


Figure 9. **Visualization of principal components (1).** PC0 and PC4 is the first principal component of global rotation and other body joint rotations respectively. The number before each row is the scaling factor for the corresponding component (multiply it with the eigenvector and show the result motion).

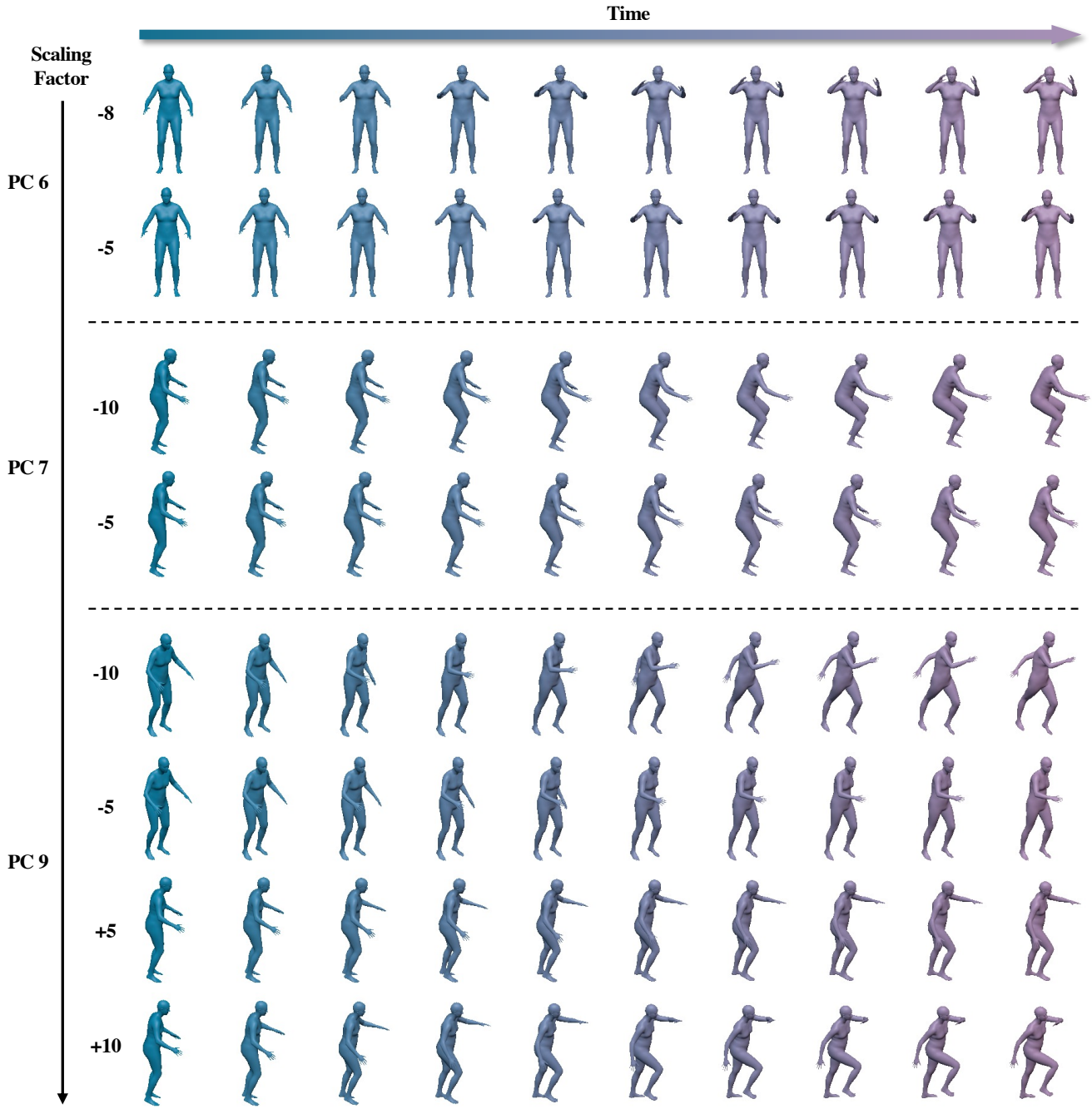


Figure 10. **Visualization of principal components (2).** Here are three principal components of body joint rotations, which in general control the movements of forearms (PC6), the bending of the legs (PC7), and motion similar to running (PC9) respectively.

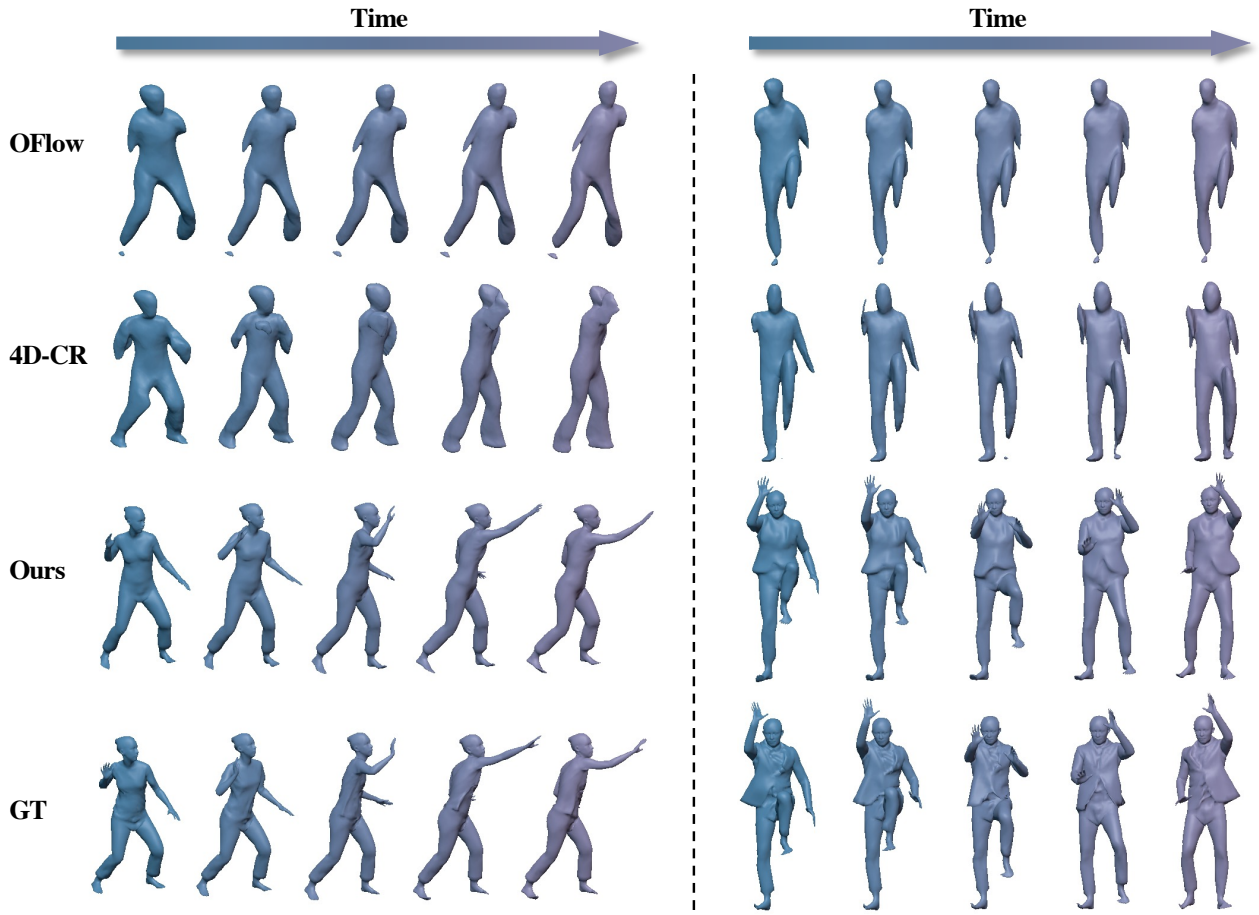


Figure 11. **4D Reconstruction.** Our method produces accurate motion sequence with fine-grained geometry (blazer, long trousers and hairstyle), while the results of baseline methods suffer from incomplete geometry with missing arms or hands, and are overly smooth.

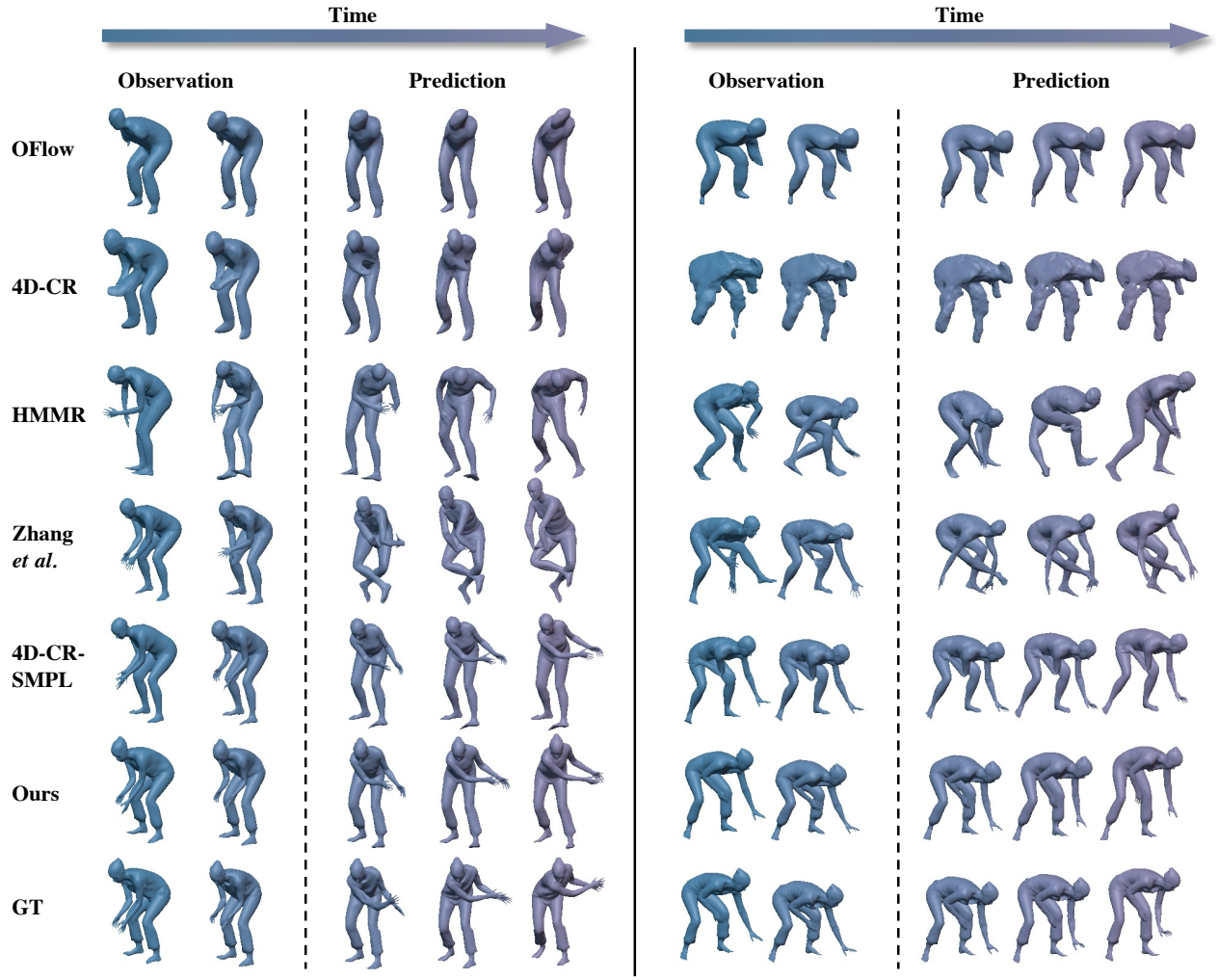


Figure 12. **Future Prediction.** Here are two different sequences on the future prediction task (split by solid line). Each sequence has $L = 30$ frames and we are aiming to extrapolate 10 future temporal frames based on 20 past observed frames. The meshes on the left of dotted line are reconstruction results of the observations, and the meshes on the right are the predictions for future time steps.

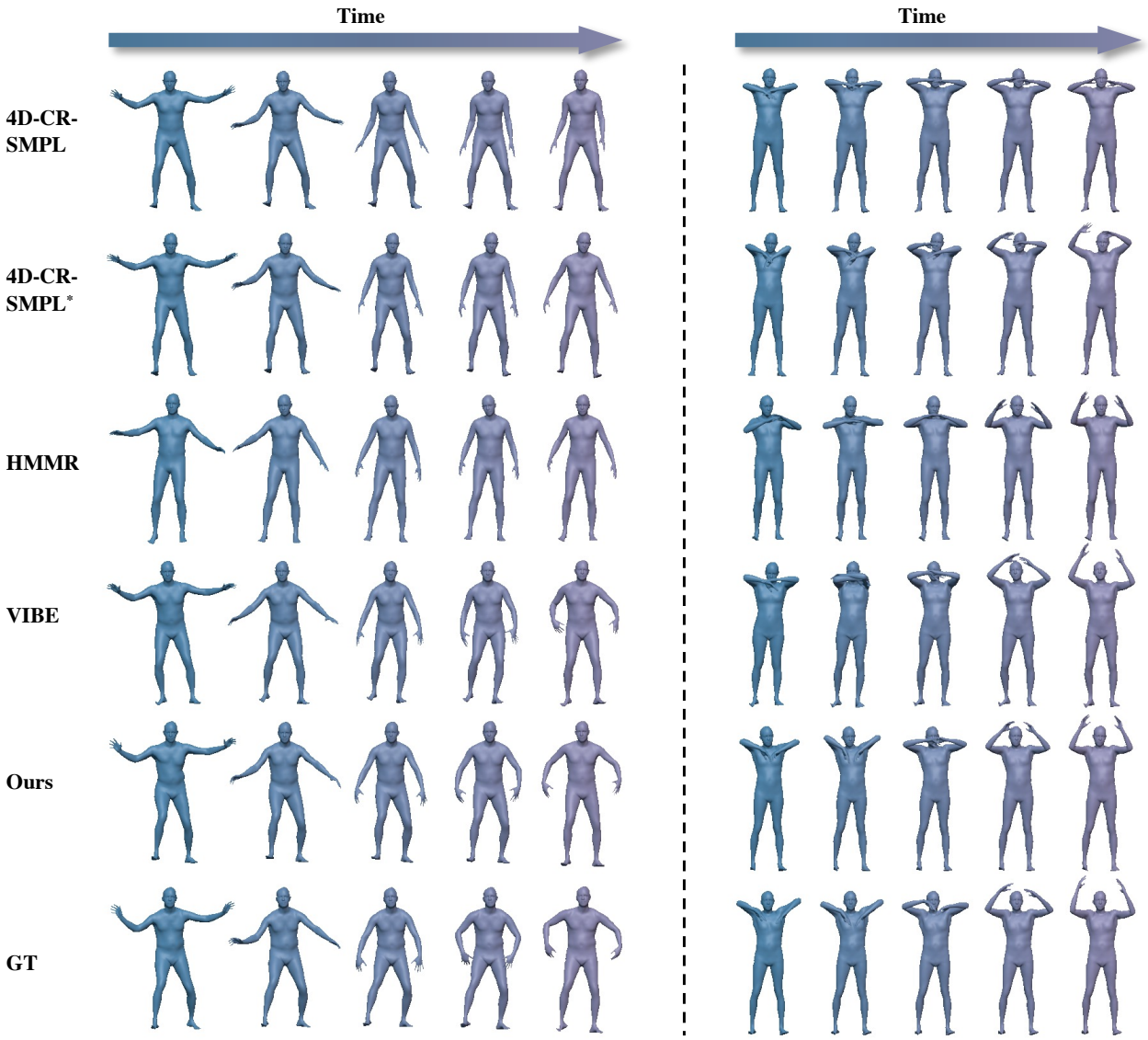


Figure 13. **Shape and Motion Recovery.** Different from the 4D reconstruction task, the goal here is to recover accurate SMPL motion sequence from the input point cloud sequence. We uniformly sample 5 frames (out of 30 frames) for visualization. Our model in general performs best among all the methods.

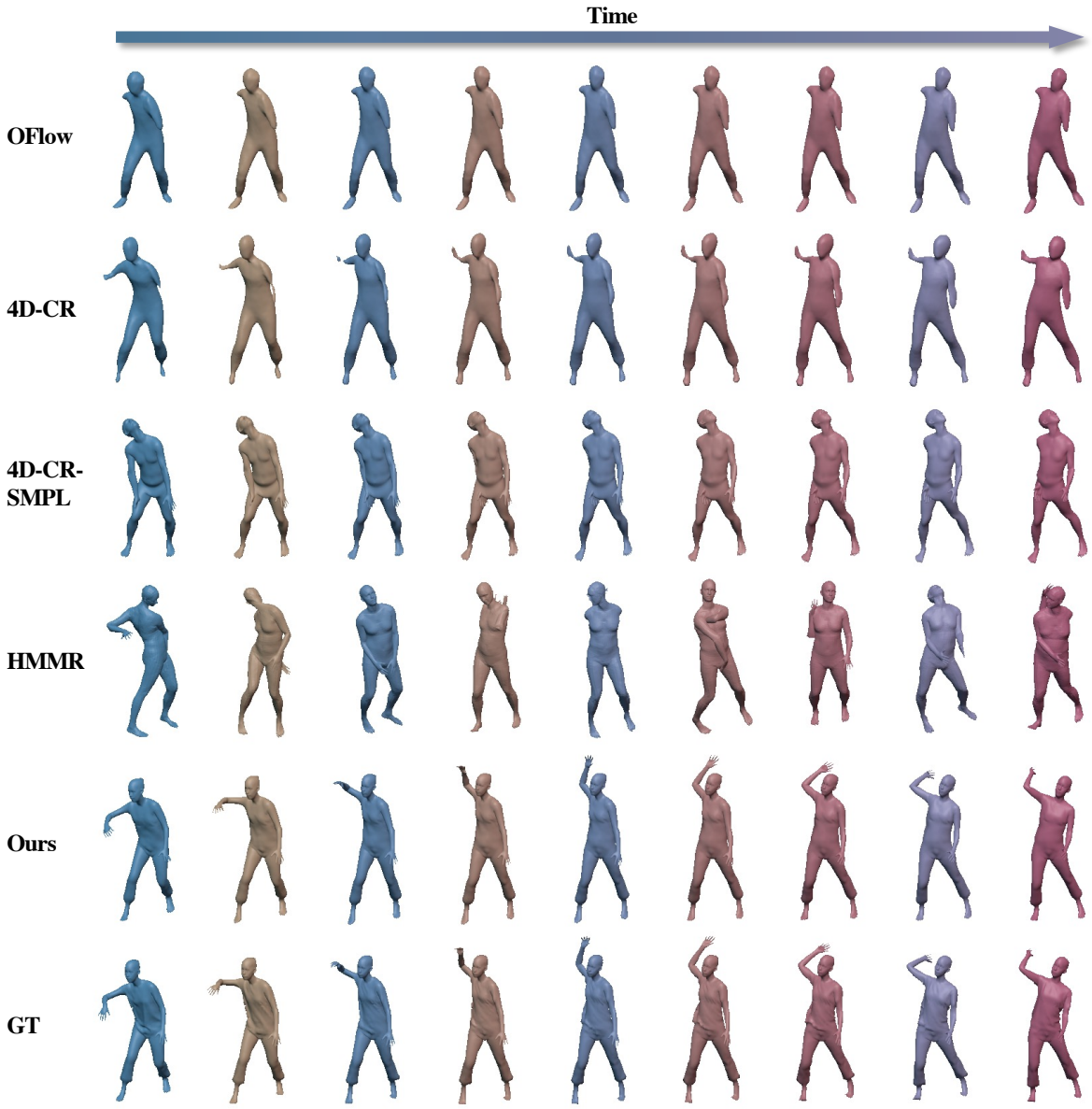


Figure 14. **Temporal Completion.** Given a sequence of $L = 30$ frames, we randomly select 15 frames as observation and perform the backward fitting algorithm to optimize the latent codes, and then reconstruct the full sequence to complete the missing frames. The meshes with yellow-red-ish color are completed unseen frames. We find 4D-CR-SMPL and HMMR produce unnatural pose results while our method successfully reconstructs and completes the full motion sequence, possibly because our linear motion model provides regularization and global temporal context for the output motion.

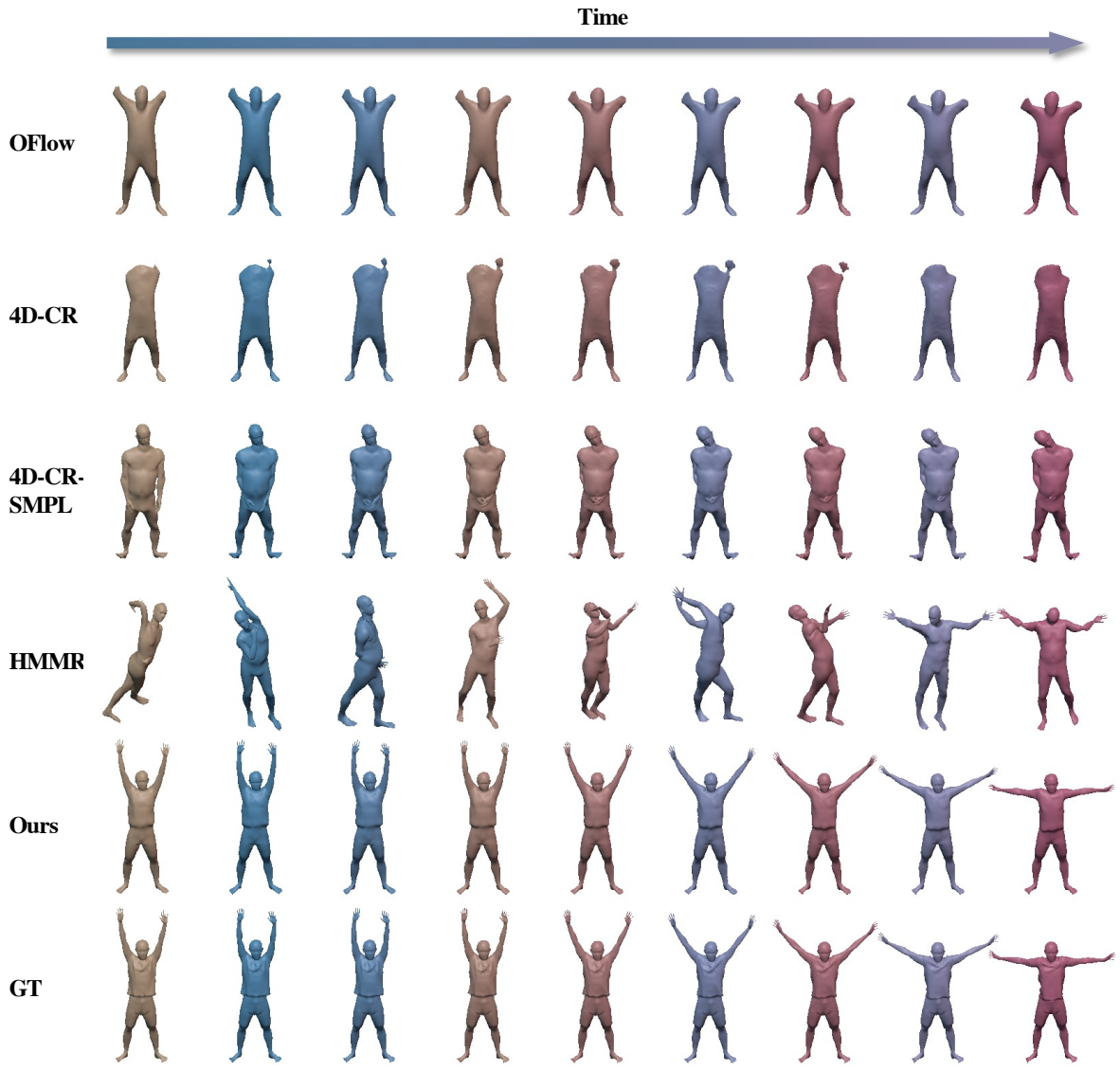


Figure 15. **Temporal Completion.** The meshes with yellow-red-ish color are completed unseen frames.

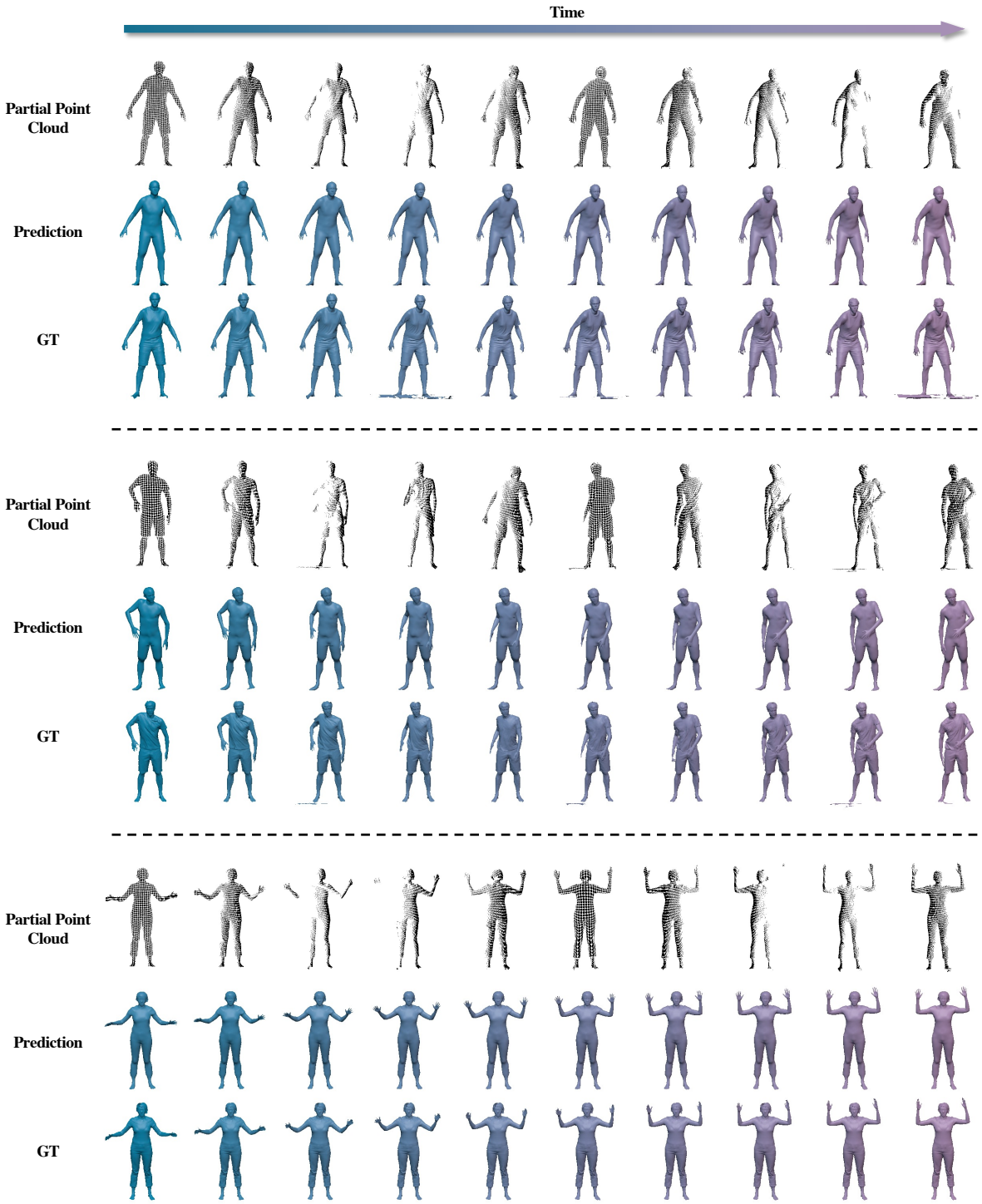


Figure 16. **Spatial Completion.** We use the partial point clouds back-projected from the depth images as observation, and reconstruct the motion sequence with complete geometry. Note that our depth images are rendered from the raw scanned mesh sequence of CAPE dataset, which simulates the real-world scenarios.

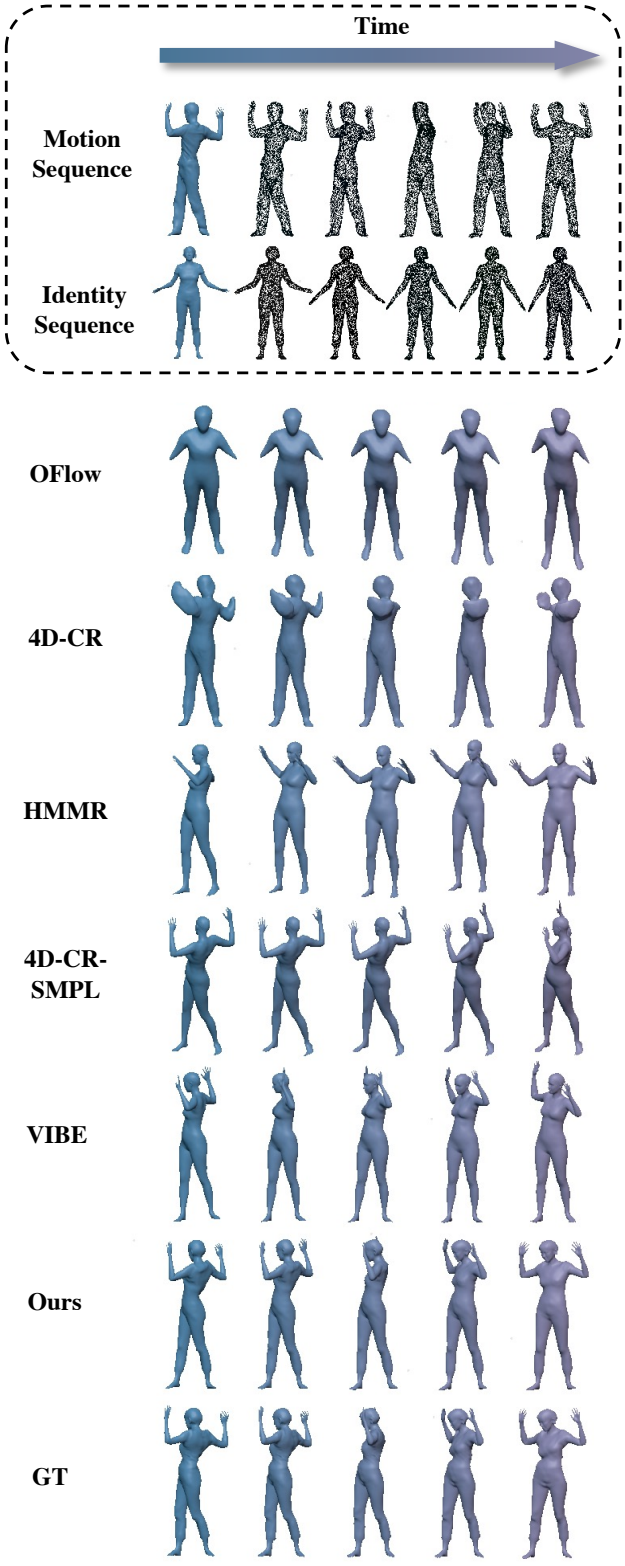


Figure 17. **Motion Retargeting.** Our goal is to transfer the human movements of the motion sequence (Row 1) to the people in the identity sequence (Row 2).