H4D: Human 4D Modeling by Learning Neural Compositional Representation

Boyan Jiang1∗ Yinda Zhang2∗ Xingkui Wei1 Xiangyang Xue1 Yanwei Fu1
1Fudan University 2Google

Abstract

Despite the impressive results achieved by deep learning based 3D reconstruction, the techniques of directly learning to model 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 a dynamic 3D human over a temporal span with the SMPL parameters of shape and initial pose, and latent codes encoding 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 shapes, and thus shown critical impacts to many human-centric tasks, such as pose estimation [16, 24, 30, 32, 34, 42] and body shape fitting [9, 18, 33, 48, 58]. 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 [26, 44] often rely on computational expensive architectures to learn the compact latent spaces and reconstruct 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 [30, 32, 66] are mostly derived from the SMPL parametric model [37], which typically employs one shape parameter and a series of pose parameters to model dynamic sequences. Although they produce plausible results, their motion representations are not compact or only support a small time span, e.g. ± 5 frames [30].

∗ 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.
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 [60], we encode each dynamic human sequence with SMPL parameters representing shape and initial pose, and a compact latent code representing temporal motion, which can then be used to reconstruct the input sequence through a decoder. At the core of this decoder, a simple yet effective prior model extended from SMPL [37] is designed to provide coarse but long-term estimation of the 3D human geometry and motion. This can ensure more complete and plausible outputs compared to the prior arts of free-form reconstructions [26, 44], 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 by 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 the SMPL parameters and latent codes for each aspect from densely sampled point clouds, which are 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 temporally tracked, i.e., the point trajectories like in previous work [26, 44]. 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 detailed motion and geometry.

**Contributions** We propose H4D, a compact and compositional representation for 4D human captures, which combines a linear prior model with the residual encoded in a learned auxiliary code. The framework is learned via 4D reconstruction, and the latent representation can be extracted from either nonregistered 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 sequences 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 objects based on various representations, such as 3D voxels [17, 22, 63], point clouds [1, 20, 50, 51], meshes [12, 23, 29, 36, 62] and implicit functions [11, 14, 19, 27, 43, 46]. However, the deep representation 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 [43, 44], point clouds [53], and compositional property [26]. 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 the recurrent network for efficient inference.

#### Human Body Estimation

For human shape and pose estimation [9, 16, 24, 32–34, 42, 48] or motion prediction [2, 3, 7, 10, 39, 40], most of works are based on SMPL or its extension [37, 47, 54]. Specifically, HMMR [30] learns to encode temporal information by reconstructing a small number of past and future frames. Zhang et al. [66] propose the first autoregressive model for predicting 3D human motion from image sequences. VIBE [32] leverages GRU to regress SMPL parameters, and designs an adversarial learning framework to predict temporal transitions. Though produce plausible motion, the motion representations in these methods are either implicit [32], coupled with geometry [30, 32, 66], or limited to a short temporal range [30]. In contrast, we formulate the motion with a prior model based on PCA [28, 61] (its non-linear extension PGA [21] is also applicable) 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, 29, 30, 32, 33, 42] are limited to the unclothed body 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 [8, 14, 18, 55–57, 64, 67], 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 the canonical body mesh [4–6, 35, 38, 65], which achieves a good balance between the expressiveness and prior regularization. Most related to us, CAPE [38] 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.
3. Method

This section introduces our H4D representation, which is learned through reconstruction task (Fig. 2). Given a 3D human model performing motion in a time span (mesh sequence of 30 frames), we sample a point cloud of 8192 points from each as the input sequence to the network. Note that we do not assume the temporal correspondences among frames (e.g. point trajectories) are available, which are critically required by previous 4D representations [26, 44].

The input sequence is fed into a compositional encoder to extract SMPL parameters representing shape and initial body pose, together with latent codes representing 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 [37] (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 shape in the canonical pose, respectively. The final sequence is obtained by deforming the refined canonical shape using the linear blending weight according to the refined per-frame poses.

3.1. Compositional Encoder

To keep the representation compositional, we train four separate encoders to extract SMPL parameters representing the shape \(c_s\) and initial pose \(c_p\), and latent codes representing motion \(c_m\) and auxiliary information \(c_a\). The shape and pose encoders are implemented as PointNet-based [51] 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 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 can process sequences of unordered point clouds without temporal correspondences.

3.2. Linear Motion Model

We take the predicted \(c_p\) and \(c_m\) to reconstruct a coarse estimation of motion. Inspired by Urtasun et al. [61], we employ the parameter space of SMPL model and pre-learn a linear model for the motion to ensure robustness. Each input temporal sequence can be represented as \(\Phi = [\theta_1, \ldots, \theta_L], L = 30\), 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. \(\Psi = [\theta_2 - \theta_1, \ldots, \theta_L - \theta_1] \in \mathbb{R}^{72(L-1)}\), and run a Principal Component Analysis (PCA) [28] to reduce the dimension. The input motion now can be reconstructed through the linear model: \(\hat{\Phi} = [\bar{\theta}_1, \alpha^T \cdot \mathcal{M} + \mu_{\Psi} + \theta_1]\), 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 \(\mu_{\Psi}\) are the top \(K\) principal components and mean of \(\Psi\).
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 bases 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 the compositional encoder: \(\{c_{p_t}\}_{t=0}^{L-1} = \{c_{p_t}, c_{m}^T \cdot \mathcal{M} + \mu_{\psi} + c_{p_t}\}\), where \(c_{p_t}\) is the pose parameter for frame \(t\).

3.3. Motion Compensation Network

The LMM is effective in representing motion 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 output of Motion-Comp network, 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 of 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 [38] 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 with 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 hairstyles) 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 non-linear 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 [25], XNect [41] and Predicting Human Dynamics [66].

Loss Functions Since our reconstructions are registered with SMPL topology, we use per-vertex L1-loss with the ground truth mesh as the objective function. 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 LMM for supervision. In the second stage, we use added loss on both meshes before and after the Shape-Comp network. The detailed formulation of the loss functions can be found in Supp. Mat.

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 on 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. Finally, we provide an ablation study to test the impact of each component in our framework on the reconstruction quality.

Dataset We use the CAPE dataset [38, 49] 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 meshes in the canonical pose and pose parameter of each frame. Overall, the dataset provides good di-
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 be overly smooth and often have incomplete geometry, e.g. missing hands. We uniformly sample 5 frames (out of 30 frames) for visualization.

4.1. Representation Capability

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

Implementation

We use PyTorch to implement the model, and train with the Adam optimizer [31]. 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 $10^{-5}$ and dropped to $10^{-5}$ after 200K iterations with batch size 4 due to the limitation of GPU memory. We use 4 NVIDIA GeForce RTX 2080Ti GPU cards.

Evaluation

To measure the difference between the prediction and ground truth 3D shape, we use Chamfer Distance (CD) and Volumetric IoU (IoU) [43] for free-form geometry and Per Vertex Error (PVE) for SMPL registered shape. To measure the accuracy of motion, we 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 joints and 21 keypoints on the face, feet and hands. Please refer to [30, 32, 43] for more details about these metrics. For temporal sequences, we take the mean score of all frames.
I. Comparison with Previous 4D Representation Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>4D Reconstruction</th>
<th>Motion Retargeting</th>
<th>Motion Completion</th>
<th>Future Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU ↑</td>
<td>CD ↓</td>
<td>IoU ↑</td>
<td>CD ↓</td>
</tr>
<tr>
<td>OFlow [44]</td>
<td>61.5%</td>
<td>0.199</td>
<td>30.7%</td>
<td>0.470</td>
</tr>
<tr>
<td>4D-CR [26]</td>
<td>62.9%</td>
<td>0.165</td>
<td>47.3%</td>
<td>0.296</td>
</tr>
<tr>
<td>Ours</td>
<td>73.3%</td>
<td>0.093</td>
<td>70.7%</td>
<td>0.100</td>
</tr>
</tbody>
</table>

II. Comparison with Human Body Estimation Methods (Forward)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Shape and Motion Recovery</th>
<th>Motion Retargeting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
</tr>
<tr>
<td>HMMR [30]</td>
<td>87.8</td>
<td>102.1</td>
</tr>
<tr>
<td>VIBE [32]</td>
<td>45.3</td>
<td>53.4</td>
</tr>
<tr>
<td>4D-CR-SMPL</td>
<td>59.2</td>
<td>68.5</td>
</tr>
<tr>
<td>4D-CR-SMPL'</td>
<td>49.8</td>
<td>57.7</td>
</tr>
<tr>
<td>Ours</td>
<td>38.4</td>
<td>44.9</td>
</tr>
</tbody>
</table>

III. Comparison with Human Body Estimation Methods (Backward)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Motion Completion</th>
<th>Future Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
</tr>
<tr>
<td>HMMR [30]</td>
<td>146.5</td>
<td>141.6</td>
</tr>
<tr>
<td>Zhang et al. [66]</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4D-CR-SMPL</td>
<td>87.3</td>
<td>67.3</td>
</tr>
<tr>
<td>Ours</td>
<td>53.8</td>
<td>42.7</td>
</tr>
</tbody>
</table>

Table 1. Comparison to SoTA methods on various tasks. For evaluation, we adopt Volumetric IoU (IoU) and Chamfer Distance (CD) [43] for comparisons with free-form methods (block I), and several standard metrics following [30, 32] for SMPL-based methods (block II&III, the numbers are measured in mm). ∗ denotes the input point cloud sequence has temporal correspondence.

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 [59], 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 SMPL parameters and latent codes for each ($c^I_a, c^I_p, c^I_m, c^I_s$) and ($c^M_a, c^M_p, c^M_m, c^M_s$). We then conduct the motion retargeting by using ($c^a, c^p, c^m, c^s$) for linear motion model, $c^M_a$ for Motion-Comp network, and $c^I_a$ 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 the canonical pose and pose parameters provided by CAPE [38] dataset to generate motion retargeted ground truth sequences. We compare with free-form geometry based approach (OFlow and from the auxiliary code. As baselines, we compare to SoTA video-based human shape and pose estimation methods HMMR [30] and VIBE [32]. Originally designed for color images input, we replace their image encoder with the point cloud encoder in our setup. Moreover, as an additional baseline, we extend 4D-CR [26] 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 unclothed SMPL defined shapes, 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 estimation (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. We provide additional comparisons to the recent motion-based human body estimation method HuMoR [52] with the task of shape and motion recovery from point clouds via auto-decoding in Supp. Material.
4D-CR) on the full geometry (in Tab. 1 (I)) and SMPL based approach (HMMR, VIBE, 4D-CR-SMPL) on the unclothed body 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 the compositional encoder is more effective in extracting correct information from inputs and facilitating motion recovering.

**Motion Completion** Our representation can also fulfill fitting tasks in the auto-decoding fashion, in which the SMPL parameters and 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 SMPL parameters and latent codes, reconstruct the full sequence, and then measure the geometry accuracy on the other 15 frames. Note that we use Chamfer loss with the additional prior terms borrowed from IPNet [8] on uniformly sampled points instead of PVE to simulate the case in real applications, 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 the qualitative results are in Supp. Material. Zhang et al. [66] uses a similar motion model to HMMR [30], 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 noise added to the observed frames in Fig. 6 (b). The error of 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 the 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 and quantitative comparisons to NPMs [45] in Supp. Material.

**Future Prediction** Our representation also supports future prediction. Specifically, we run a fitting algorithm on the first 20 frames, generate the SMPL parameters and latent codes, and then reconstruct the 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 [30], Zhang et al. [66] and 4D-CR-SMPL [26]), our method still achieves better performance. Moreover, we empirically found that, though given
pose prior terms 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 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 produces reasonable prediction of future motion. The meshes on the left and right are reconstructions of the observations and predictions for the future time steps, respectively.

4.3. Ablation Study

In this section, we perform an ablation study and show the quantitative results to demonstrate the effect of the major designs in our method. The visualization example can be found in Supp. Material. **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 the linear motion model (“-LMM”), which directly updates the initial pose code, or removing Motion-Comp network (“-Motion c_a”), which only relies on the linear motion model (LMM) for motion recovery. In either case, the motion accuracy drops consistently as measured by all metrics, indicating the necessity of combining the prior model with learned compensations.

**Shape Model** We then verify if the auxiliary code helps to recover the 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 [44] and 4D-CR [26], 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 framework is designed for learning our representation, which encodes the input point cloud sequences into the SMPL parameters of shape and initial pose, and latent codes of motion and auxiliary information. Extensive experiments on 4D reconstruction, shape and motion recovery, motion retargeting, completion and prediction validate the efficacy of the proposed approach.

Acknowledgements

This work was supported in part by NSFC Project (62176061), Shanghai Municipal Science and Technology Major Project (2018SHZDZX01). The corresponding authors are Xiangyang Xue, and Yanwei Fu.
References


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


[34] 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


[53] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. Ad-


