APPENDIX

In this appendix, we provide additional information about the dataset, implementation details, post-processing techniques. We also discuss on the current limitation as future research perspectives.

1 Dataset

1.1 Motion Data

Table 1 shows a detailed break down of our dataset in terms of different types of interactions. Our dataset consists of 3 hours of MoCap with over 500 motion sequences. In addition to the dataset, we explain the data structure and provide the demonstration code on how to use the dataset [1].

Table 1: Distribution of the COUCH dataset with different types of interaction.

Interaction Type	Minutes	%
Right Hand	36.3	17.3
Left Hand	29.4	14.0
Both Hand	60.5	28.9
No Contact	36.5	17.4
Free Interaction	31.9	15.2
Locomotion	15.1	7.2

1.2 Data Processing

SMPL Fitting. We segment the human in captured RGB images by running Detectron V2 [9] followed by manual correction with [8] on the segmentation masks. These masks are then used to segment multi-view depth maps and lift human point clouds from 2D to 3D. We use FrankMocap [7] to initialize the SMPL pose from the images and then apply instance specific optimization [4] to fit the SMPL model to the segmented human point cloud. For more accurate fitting, we additionally obtain the SMPL shape parameters of each subject from 3D scans using [5].

Synchronization with the IMUs. The fitted SMPL model provides us with accurate contacts with the scene, however, the fitted motion sequence is prone to occlusion and drastic body movements, as a result, the fitted motion can be jittery at times. On the other hand, the pose captured with the IMUs is smooth over time, but it might not accurately capture the contacts. To this reason, we synchronize the Kinect captured data with the body sensors by incorporating the SMPL fitted poses into the IMU pose sequences. After synchronization, we

optimize the joint rotations $\boldsymbol{j}_i^r$ to achieve temporal smoothness via the objective

$$L_{\text{temp}}(\boldsymbol{j}_i^r) = \sum_{i=1}^{T-1} \|\boldsymbol{j}_{i+1}^r - \boldsymbol{j}_i^r\|^2 + \sum_{i=1}^{T-1} \|\ddot{\boldsymbol{j}}_i\| \tag{1}$$

where $\ddot{\boldsymbol{j}}_i$ represents the acceleration of the body joints in frame i approximated by central difference.

We additionally use the binary contact labels of the toes and the heels detected by the IMU sensors to remove foot-sliding on the motion data. To remove the foot-sliding, we compute the average joint positions over the duration of the contacts grouped by the positive contact labels. This computation is performed for all four foot joints. This forms a sequence of target joint positions of the feet $\tilde{\boldsymbol{f}}_i \in R^{4\times3}$. We then optimize the objective function

$$L_{\text{slide}}(\boldsymbol{j}_i^r) = \sum_{i=1}^{T} \|\tilde{\boldsymbol{f}}_i - \boldsymbol{f}_i\|^2, \tag{2}$$

where $\boldsymbol{f}_i$ represents the foot joint positions at frame i. The resulting motion sequence is temporally smooth and has accurate contacts registered with the chair models.

Object Processing. To obtain object segmentation, we pre-scan objects using a 3D scanner [2,3]. We then use multi-view object keypoints, marked by manual annotators on the images, to fit the pre-scanned chair meshes to the given frame. The segmentation masks are then obtained by projecting fitted object meshes to the images. Since the chairs remain static during the capture, we average over the 6D pose of the fitted chair model during each capture session to obtain the final transformation of the chair.

2 Training Details

2.1 ControlNet

As shown in Figure 1, the contact network is a two-layer LSTM architecture. Each layer has a hidden dimension of 512. The pose and the control signals (hand trajectories, and the local phases) are each encoded through a two-layer fully connected network with of shape {128, 128} before passing through the LSTM. We apply scheduled sampling on hand trajectories for better model performance. For the local phases, we always use the ground truth. Each of our training samples is in a sequence of 60 frames. The ControlNet is trained for 150 epochs with an Adam optimizer. The initial learning rate is 1e-3 and a cosine learning rate scheduler was used to decay the learning rate gradually to 5e-6. The full training of a subject-specific model takes approximately 1 hour on an NVIDIA V100 GPU.

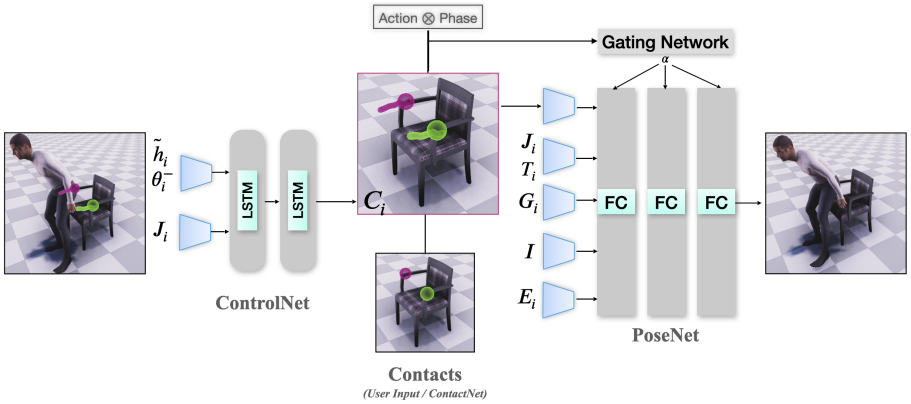

Fig. 1: Our method that combines the ControlNet and the PoseNet.

2.2 PoseNet

The PoseNet adopts the mixture-of-expert structure [6]. It consists of different feature encoders of structures shown in Table 2. The gating network and the prediction networks are both three-layer fully-connected networks, with hidden dimensions of 128 and 512 respectively. The number of experts is set to 10. The PoseNet is trained for 150 epochs with an Adam optimizer. The initial learning rate is 1e-4 and a cosine learning rate scheduler was used to decay the learning rate gradually to 5e-6. The full training of a subject-specific model takes approximately 6 hours on an NVIDIA V100 GPU.

Table 2: Details on different encoder networks of the PoseNet.

Networks	Architecture
Encoder for **C**	{128,128,128}
Encoder for {**J**, **T**}	{512, 512, 512}
Encoder for **G**	{128,128,128}
Encoder for **I**	{512, 512, 512}
Encoder for **E**	{256,256,256}

2.3 ContactNet

The ContactNet encodes the scene **I** through a three-layer fully connected network of shape {512, 512, 64}. The latent vector $\boldsymbol{z}$ of the VAE is of size 6. The weight of the Kullback-Leibler divergence β is 0.1. We use the Adam optimizer with a learning rate of 1e-3 and train ContactNet for 150 epochs. The full training of a subject-specific model takes approximately 10 minutes on an NVIDIA V100 GPU.

3 Contact Projection and Trajectory Fitting

To ensure the ContactNet predicts contacts that land exactly on the surface of the object, we perform a post-processing step, when the distance of the network predicted contact to the surface is less than a set threshold of 10 cm, we simply project the contact onto the nearest point on the chair surface. When the distance is greater than 10 cm, we simply neglect the predicted contact. The ControlNet predicts the future hand trajectories, and it would be possible to fit the predicted pose to the predicted hand position from the hand trajectories at each frame to further improve the satisfaction of the contact constraints. Note, in the evaluation of the main paper we do not apply such fitting technique.

4 Controlling with a series of Contacts.

One useful application of our approach is to automatically generate a motion sequence with a series of desired contacts in the context of animation, character control, when executing a set of complex actions. For instance, the person can be instructed to first sit with their hands on the armrest, then lift the arms to support the head before bringing the hands back to the armrest, see Figure 2) and the supplementary video. Our approach can be adapted for this task by iteratively providing the new goal locations for the hands as input after the present locations are reached.

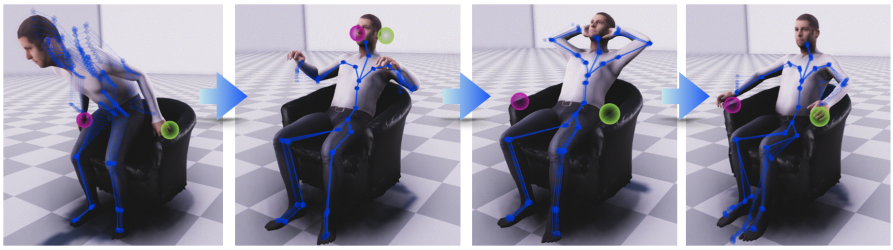

Fig. 2: COUCH can also be extended by specifying a series of contacts for to automatically synthesize more complex interactions. The past poses are indicated by blue skeletons.

5 Limitations and Future Direction

We observe the synthesized motion can slightly intersect with the chair. A solution to this problem would be to apply a post-processing step to avoid such collision. In order to generalize to more different chair shapes, it would be useful

to investigate better ways of encoding the scene geometry while trying to avoid over-fitting.

Different shaped person can intersect with the same object very differently even when performing the same motion. The COUCH dataset captures human interaction with different body shapes. With the dataset, it is possible to study how to build subject-variant motion synthesis model and how to effectively condition on the body shapes. These are challenges in motion synthesis that have not been tackled.

Our work on controllable human-chair interaction. It would be useful to extend the scope of interacted objects, especially considering the cases when the objects are non-static, when performing motions such as lifting a box, or opening a door. Another possible direction would be to further apply contact-based control in these interactions.

References

1. http://virtualhumans.mpi-inf.mpg.de/couch/
2. https://www.treedys.com/.
3. Agisoft metashape. https://www.agisoft.com/
4. Alldieck, T., Magnor, M., Bhatnagar, B.L., Theobalt, C., Pons-Moll, G.: Learning to reconstruct people in clothing from a single RGB camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (jun 2019)
5. Bhatnagar, B.L., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: Combining implicit function learning and parametric models for 3d human reconstruction. In: European Conference on Computer Vision (ECCV). Springer (August 2020)
6. Eigen, D., Ranzato, M., Sutskever, I.: Learning factored representations in a deep mixture of experts. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings (2014)
7. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In: IEEE International Conference on Computer Vision Workshops (2021)
8. Sofiiuk, K., Petrov, I., Barinova, O., Konushin, A.: f-brs: Rethinking backpropagating refinement for interactive segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8623–8632 (2020)
9. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)