SAGA: Stochastic Whole-Body Grasping with Contact
** Appendix **

Yan Wu*[1], Jiahao Wang*[2], Yan Zhang[1], Siwei Zhang[1], Otmar Hilliges[1], Fisher Yu[1], Siyu Tang[1]

[1] ETH Zürich, Switzerland
[2] Max Planck Institute for Informatics, Germany
yan.wu@vision.ee.ethz.ch, jiwang@mpi-inf.mpg.de,
{yan.zhang,siwei.zhang,otmar.hilliges,siyu.tang}@inf.ethz.ch, i@yf.io

In the Appendix, we first provide the body markers representation, architecture details, training and optimization setup, dataset pre-processing details, and the AMT user study evaluation details in Appendix A. In Appendix B, we illustrate the detailed baseline experiments and ablation study setup. We further provide additional experimental results and visualization results in Appendix C, and we discuss the existing limitations in Appendix D. Please see the **video** in our project page for more random samples of synthesized grasping poses and grasping motions.

A Method and Implementation Details

A.1 Body Markers Placement

To have informative yet compact markers setup, as illustrated in Fig. S1, we follow the placement of the markers in GRAB [10] MoCap system, having 49 markers for the body, 14 for the face, 6 for hands and 30 for fingers (see Fig. S1 (a)-(c)) on SMPL-X body surface. As hand poses are subtle and the palm is frequently in contact with the object, we additionally have 44 markers on two palms (see Fig S1 (d-1)) to further enrich the markers information for a better grasp. For the grasping ending pose generation in stage1, we use all these 143 markers for training and optimization. For the motion infilling network training in stage2, we only use a sparse set of palm markers with 10 markers on fingertips (see Fig. S1(d-2)).

A.2 Architecture Details

WholeGrasp-VAE We have visualized the WholeGrasp-VAE architecture in Fig. 3. This CVAE is conditioned on the object height information and the object geometry feature extracted with PointNet++ [9] encoder. In the encoder, taking the body markers' postions $\boldsymbol{M} \in \mathbb{R}^{N\times 3}$ and body markers contacts

* Equal contribution.

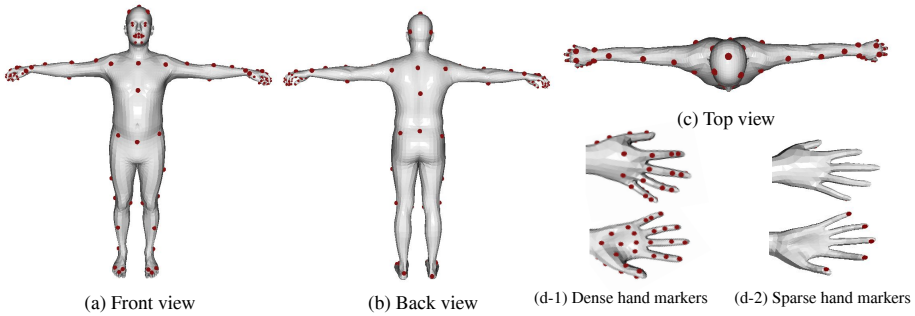

Fig. S1. Visualization of our body markers placement. We have 69 markers on the body surface, which are visualized as red spheres on SMPL-X body surface, in which 49 for the body, 14 for the face and 6 for hands. For the WholeGrasp-VAE training, we additionally have 30 markers on fingers and 44 markers on palms (d-1). For the MotionFill-VAE training, we only have sparse hand markers (d-2) with 10 markers on fingertips.

$C_{\boldsymbol{M}} \in \{0,1\}^N$ as inputs, where N is the number of markers, the body branch encodes the body feature $\mathcal{F}_B$; Taking the object contacts $C_{\boldsymbol{O}} \in \{0,1\}^{2048}$ as an additional feature of the object point cloud data, the object branch uses the PointNet++ to encode the object feature. Further, we fuse them into a joint 16-dimensional latent space $\boldsymbol{z}_s$. In the decoder, we individually decode the body markers' positions, markers' contacts, and object contacts. Note that we model the contacts learning as a two-class (*in-contact* or *out-of-contact*) classification task, and the decoder outputs the *in-contact* probability of each points. And the PointNet++ encoder architecture is given by: SA(256, 0.2, [64, 128]) $\rightarrow$ SA(128, 0.25, [128, 256]) $\rightarrow$ SA([256, 512]), where SA denotes the set abstraction level [9].

MotionFill-VAE In TrajFill, the root state at time t is given by $\boldsymbol{\Gamma}_t = (x_t, y_t, \cos\gamma_t, \sin\gamma_t)$, where x_t, y_t are the position of pelvis joint in the x-y (ground) plane, and γ_t is the body rotation around z-axis. Given $\boldsymbol{\Gamma}_0$ and $\boldsymbol{\Gamma}_T$, the TrajFill is built to learn the deviation $\Delta\boldsymbol{\Gamma}_{0:T+1} = \boldsymbol{\Gamma}_{0:T+1} - \overline{\boldsymbol{\Gamma}}_{0:T+1}$ from an initial trajectory $\overline{\boldsymbol{\Gamma}}_{0:T+1}$ which is a linear interpolation and one-step extrapolation of the given $\boldsymbol{\Gamma}_0$ and $\boldsymbol{\Gamma}_T$, and we use $\overline{\boldsymbol{\Gamma}}_{0:T+1}$ as the condition. Inside TrajFill, we use MLP structures for the encoder and the decoder. For the encoder, input trajectory features are passed through two residual blocks, which has the same hidden size as the input dimension ($8T$, where T is the time length of the input). After that, two linear branches project the features into the 512-dimensional latent space. The decoder includes two residual blocks with hidden sizes equal to $8T$ and $4T$, respectively. We get the final output of TrajFill by adding the last residual block output and the initial rough trajectory $\overline{\boldsymbol{\Gamma}}_{0:T+1}$.

In LocalMotionFill, following the same input processing step as in [12], we build a 4-channel local motion image $\boldsymbol{I} \in \mathbb{R}^{4\times(3N+n)\times T}$, where N, n are the number of markers and the dimension of foot-ground contact labels. The first

channel of $\boldsymbol{I}$ is a concatenation of foot-ground contacts $C_{F_{0:T}} \in \{0,1\}^{n\times T}$ and the normalized local markers $\boldsymbol{M}^l_{0:T} \in \mathbb{R}^{3N\times T}$.The other three channels of $\boldsymbol{I}_l$ are the normalized root local velocities $\boldsymbol{v}^l_{0:T}$. To incorporate the condition information $(\boldsymbol{M}_0, \boldsymbol{M}_T, \boldsymbol{v}_{0:T})$ into the LocalMotionFill, similarly, we build a condition image $\boldsymbol{I}_c$ which essentially is the masked input image $\boldsymbol{I}$. We use the same CNN-based encoder and decoder network as in [12] to learn the infilled motion image.

A.3 Dataset Processing

GRAB. We use GRAB (https://grab.is.tue.mpg.de/license) dataset to train both the WholeGrasp-VAE and MotionFill-VAE for grasping ending pose generation and motion infilling, respectively.

For WholeGrasp-VAE training, considering the different body shape pattern of male and female, we suggest training the male model and the female model separately. Following GrabNet [10], for training, we take all frames with right-hand grasps. And out of the 51 different objects in GRAB dataset, following the same split of object class in GrabNet [1,10] we take out 4 validation objects (*apple, toothbrush, elephant, hand*) and 6 test objects (*mug, camera, toothpaste, wineglass, frying pan, binoculars*), and the remaining 41 objects are used for training. We center the object point cloud and the body markers at the geometry center of the object.

For MotionFill-VAE training, we only utilize sequences where humans are approaching to grasp the object. GRAB dataset captures the motion sequences where the human starts with T-pose, approaches and grasp the object, and then interacts with the object. For MotionFill-VAE training, we clip those approaching and grasping sequences. Since most of these approaching motion sequences only last for about 2s in the GRAB dataset, we clip 2-second videos from each sequence by ensuring that the last frames are at stable grasping poses. If the sequence is shorter than 2s, we pad the first frame to have the two-second clip.

AMASS. We pretrain our motion infilling model MotionFill-VAE on the AMASS (https://amass.is.tue.mpg.de/license.html). We down-sample the sequences to 30 fps and cut them into clips with same duration. To be consistent with GRAB dataset, we clip 2-second sequences from the AMASS for the grasping motion infilling task. We also evaluate our motion infilling network by conducting experiments on the general motion infilling task with different time lengths (see Appendix C.2), and for that, we clip the AMASS dataset into 4-second and 6-second sequences. Similar to [13], we reset the world coordinate for each clip. The origin of the world coordinate is set to the pelvis joint in the first frame. The x-axis is the horizontal component of the direction from the left shoulder to right shoulder, the y-axis faces forward, and the z-axis points upward. For training, we use all the mocap datasets except EKUT, KIT, SFU, SSM synced, TCD handMocap, and TotalCapture. For testing, we use TCD handMocap, TotalCapture, and SFU.

Algorithm 1 WholeGrasp-Opt: grasping pose optimization

Input: Sampled markers $\hat{\boldsymbol{M}}$, markers-object contacts $\hat{C}_{\boldsymbol{M}}$, $\hat{C}_{\boldsymbol{O}}$.
Output: Body mesh $\boldsymbol{B}_T(\boldsymbol{\Theta}_T)$ and the queried markers $\boldsymbol{M}_T$.
Require: Optimization steps $(N_1, N_2, N_3) = (300, 400, 500)$.

```
for i = 1 : N_1 do
    Optimize t, R to minimize E_fit in Eq. 5.
for i = N_1 : N_1 + N_2 do
    Optimize t, R, β, θ_b to minimize E_fit in Eq. 5.
for i = N_1 + N_2 : N_1 + N_2 + N_3 do
    Optimize θ_b, θ_h, θ_e to minimize E_opt in Eq. 4-6.
return Θ_T = [β, t, R, θ_b, θ_h, θ_e]
```

A.4 Implementation Details

We implement our experiments using PyTorch v1.6.0 [7]. In the following, we introduce the training details of WholeGrasp-VAE and MotionFill-VAE, and optimization details of GraspPose-Opt and GraspMotion-Opt respectively.

WholeGrasp-VAE training. In Sec. 3.2, we have introduced the WholeGrasp-VAE training losses. Note that for the object and markers contact map reconstruction, due to the class in-balance, we employ the weighted binary cross-entropy loss, and we empirically set the weights for *in-contact* class for objects and markers as 3 and 5 respectively. And we set the object and markers contact map reconstruction weight $\lambda_{\boldsymbol{M}}, \lambda_{\boldsymbol{O}}$ in Eq. 1 as 1. For the VAE training, we adopt the linear KL weight annealing strategy [5] to avoid posterior collapse issue in VAE training. And we empirically set $\lambda_c = 1$ and $\lambda_{KL} = 0.005e$ in $\mathcal{L}_{train} = \mathcal{L}_{rec} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_c\mathcal{L}_c$, where e is the epoch number, and we train the WholeGrasp-VAE for 40 epochs.

MotionFill-VAE training. In the experiments for Table. 3, for local motion infilling (given the starting pose, ending pose and trajectory), we train our LocalMotionFill model on AMASS training set; for the entire MotionFill-VAE ("Traj + local motion infilling"), we first pretrain our TrajFill module and LocalMotionFill module on the GRAB and AMASS training set respectively, and we further finetune the entire MotionFill-VAE on the GRAB training set. We empirically set the hyper-parameters in $\mathcal{L}_M = \mathcal{L}_{rec} + \lambda_{KL}\mathcal{L}_{KL}$ and Eq. 8 as follows: $\{\lambda_{KL}, \lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{1, 0.05, 1, 1, 0.5\}$.

GraspPose-Opt optimization. In Sec. 3.2, we have illustrated the GraspPose-Opt optimization losses design to recover SMPL-X body mesh from sparse markers and refine the body pose for more perceptually realistic human-object interactions. We empirically set the hyper-parameters in Eq. 5-7 $\{\alpha^o_{cont}, \alpha^m_{cont}, \alpha^{\boldsymbol{O}}_{colli}, \alpha^{\boldsymbol{B}}_{colli}, \alpha_{\boldsymbol{\theta}}\} = \{15, 15, 100, 200, 0.0005\}$. As it can be difficult for the optimization process to converge by jointly optimizing the high-dimensional SMPL-X parameters, which include the body global configuration $\boldsymbol{t}$ and $\boldsymbol{R}$, shape parameters $\boldsymbol{\beta}$, body pose parameters $\boldsymbol{\theta}_b$, and the more local hand pose $\boldsymbol{\theta}_h$ and eye pose

$\boldsymbol{\theta}_e$ parameters, similar as in MOJO [13], we suggest a multi-stage optimization mechanism by optimizing in a *global-to-local* fashion to facilitate a gradual convergence. And the detailed multi-stage training process can be found in Alg. 1. We use Adam [4] optimizer, and the initial learning rates for these three stages are set as 0.016, 0.012, 0.008 respectively.

GraspMotion-Opt optimization. In Sec. 3.4, we have shown the GraspMotion-Opt optimization losses design to recover smooth SMPL-X body motions from sparse markers sequence. Additionally, we introduce more details about our motion smoothness loss and foot skating loss design.

– **Cross-frame smoothness loss.** To encourage a temporarily smooth whole-body motion, following [12], we enforce smoothness on the smooth motion latent space $S_{1:T-1} = AE(\boldsymbol{M}_{1:T} - \boldsymbol{M}_{0:T-1})$ encoded by a pretrained autoencoder. Also, we explicitly enforce smoothness on the hand vertices, and the overall smoothness loss is given by:

$$E_{smooth} = \alpha_s^B \sum_{t=1}^{T-2} |S_{t+1} - S_t|^2 + \alpha_s^h \sum_{t=0}^{T-1} |\mathcal{V}^h_{\boldsymbol{B}_{t+1}} - \mathcal{V}^h_{\boldsymbol{B}_t}|^2 \tag{S.1}$$

– **Foot skating loss.** Following [12], we reduce the foot skating artifacts by optimization based on the foot-ground contact labels $\hat{C}_F$.

$$E_{skat} = \alpha_{skat} \sum_{t \in T_c} \sum_{|v_t^{foot}| \geq \sigma} ||v_t^{foot}| - \sigma| \tag{S.2}$$

where T_c means the timestamps with foot-ground contact, v_t^{foot} represents the velocity (location difference between adjacent timestamps t and $t+1$) of vertices on the left toe, left heel, right toe, and right heel, at time t. σ is a threshold and we use $\sigma = 0.1$ in our experiments.

The overall optimization loss is given by $E_{basic} + E_g + E_{smooth} + E_{skat}$, where E_{basic} and E_g are formulated in Eq. 10 - 11. We optimize the overall loss in two stages, where we first fit SMPL-X body mesh to the predicted markers sequences by minimizing the markers fitting loss (E_{fit} in Eq. 10), and then we refine the recovered body mesh sequences by minimizing the overall loss. We present the detailed optimization procedure in Alg. 2. We also use Adam [4] optimizer. For the first frame, the initial learning rate is 0.1, and for the other frames the initial learning rate is 0.01. Stage 1 optimization takes 100 steps and the learning rate becomes 0.01 after step 60 and decreases to 0.003 after step 80. The second stage optimization takes 300 steps, and the initial learning rate is set to 0.01 and decays to 0.005 after 150 steps.

Algorithm 2 GraspMotion-Opt: grasping motion optimization

Input: Sampled body markers sequences $\hat{\boldsymbol{M}}_{0:T}$, mutual markers-object contacts $\hat{C}_{\boldsymbol{M}}$, $\hat{C}_{\boldsymbol{O}}$ and dynamic foot-ground contact $\hat{C}_{F_{0:T}}$; Body shape parameters $\boldsymbol{\beta}$
Output: Smoothed whole-body grasping motion $\boldsymbol{B}_{0:T}(\boldsymbol{\Theta}_{0:T})$.
Require: Optimization steps $(N_1, N_2) = (100, 300)$.

```
for i = 1 : N_1 do
    Optimize [t, R, θ]_{0:T} to minimize Σ_{t=0}^{T} E_fit in Eq. 10.
for i = N_1 : N_1 + N_2 do
    Minimize E_basic + E_g + E_smooth + E_skat in § 3.4
return Θ_{0:T} = {β, [t, R, θ]_{0:T}}
```

A.5 Amazon Mechanical Turk (AMT) User Study

We perform user study via AMT, and the user study interface is presented in Fig. S2. We perform user study on both ground truth motion sequences from GRAB dataset and our randomly generated sample sequences. We test our pipeline with 14 unseen objects from both GRAB test set and HO3D [2] dataset, and we generate 50 random grasping motion sequences for each object, where objects are randomly placed. Each sequence is scored by 3 users and we take the average score, and the score range from 0 to 5 (from *strongly disagree* to *strongly agree*). The average perceptual score for each object class is presented in Table S1.

Table S1. Perceptual score results of both ground truth (GT) grasping motion sequences and our synthesized sequences for grasping various unseen objects. Due to the lack of ground truth whole-body grasping motions for objects in HO3D dataset, we only evaluate ground truth sequences for objects from GRAB dataset.

GRAB			HO3D		Average Score	
Object	Score		Object	Score (Ours)	GT	Ours
	GT	Ours				
			Cracker box	3.15		
Binoculars	3.83	2.98	Sugar box	3.45		
Camera	3.92	3.60	Mustard bottle	3.49		
Toothpaste	4.22	3.47	Meat can	3.43	4.04	3.15
Mug	4.38	2.82	Pitcher base	2.68		
Wine glass	3.85	2.87	Bleach cleanser	3.56		
Frying pan	4.16	1.7	Mug	2.61		
			Power drill	2.86		

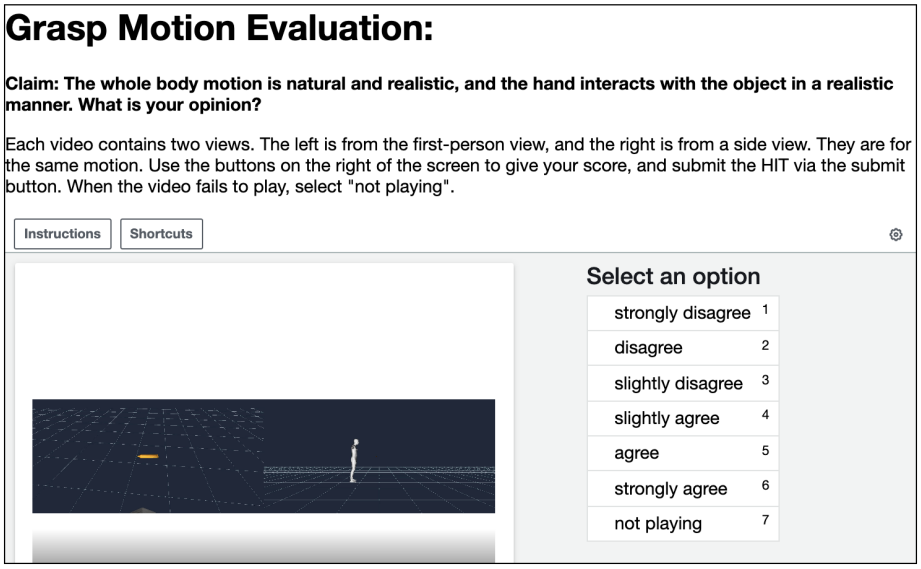

Fig. S2. AMT user study interface. We present both the first-view video (on the left side) and third-view video (on the right side) to the user for the grasping motion quality evaluation.

B Baselines Implementation Details

In Sec. 4.1 and Sec. 4.2, we conduct several baseline experiments as comparisons with our WholeGrasp-VAE and MotionFill-VAE and also some ablation studies. In this section, we illustrate more implementation details about our baselines and ablation studies.

B.1 Baselines to WholeGrasp-VAE

In Sec. 4.1, we extend the GrabNet [10] to the whole-body grasping pose generation task (GrabNet-SMPLX) as a comparison with our WholeGrasp-VAE design, and we also study the effectiveness of the multi-task WholeGrasp-VAE by comparing with the single-task design (WholeGrasp-single). In the following, we provide detailed experimental setup of these two experiments.

– **GrabNet-SMPLX.** GrabNet proposed to synthesize diverse hand grasps by directly learning the hand model parameters. And a similar idea can be extended to the full-body grasp synthesis by learning the compact SMPL-X body model parameters, which can include the body global configurations t and R, the shape parameters β, and the full-body pose $\theta = [\theta_b, \theta_h, \theta_e]$. Fig. S3 (a) shows the schematic architecture of the GrabNet-SMPLX baseline. Different from the original GrabNet, instead of encoding the object shape using

basic point set features [8], we employ the PointNet++ [9] in the GrabNet-SMPLX baseline, which is consistent with our WholeGrasp-VAE.

– **WholeGrasp-single.** As visualized in Fig. S3 (b), we build a single-task WholeGrasp-VAE, namely WholeGrasp-single, where we only learns the body markers positions. We employ the same multi-stage optimization algorithm as in GraspPose-Opt to fit SMPL-X body mesh to sampled markers. Recall that in our GraspPose-Opt, with the predicted body and object contact map, we design a contact loss accordingly (see Eq. 6) to refine the mutual contacts between the human body and object. However, due to the lack of contact map prediction in the single-task WholeGrasp-single, we cannot directly leverage this contact loss term to refine the hand pose. Instead of simply ignoring the contact refinement loss term in this test-time optimization step, we build a strong baseline by pre-defining a fixed hand contact pattern and designing a heuristic contact loss accordingly. Concretely, we firstly compute the average contact probability for each hand vertices over all the GRAB dataset, and we denote this hand contact prior probability as $C_{\mathcal{H}}$. Heuristically, we encourage those hand vertices that have a high prior contact probability (greater than 0.7) and also are closed enough to the object (less than 2cm) to contact with the object, and we formulate this heuristic contact loss baseline E^h_{cont} as:

$$E^h_{cont} = \alpha^h_{cont} \sum_{h \in \mathcal{V}^h_{\boldsymbol{B}}} \mathbb{1}(C_h > 0.7)\mathbb{1}(d(h, \boldsymbol{O}) < 0.02) * C_h d(h, \boldsymbol{O}) \tag{S.3}$$

where $\mathcal{V}^h_{\boldsymbol{B}}$ and $\boldsymbol{O}$ denote the hand vertices and object point cloud respectively, and $d(x, \mathcal{Y}) = \min_{y \in \mathcal{Y}} ||x - y||_2^2$. Therefore, the overall optimization loss for the single-task WholeGrasp-single experiment is given by:

$$E^{single}_{opt}(\boldsymbol{\Theta}) = E_{fit} + E^o_{colli} + E^h_{cont} + E^g_{cont}. \tag{S.4}$$

where $E_{fit}, E^o_{colli}, E^g_{cont}$ have the same formulations as in our GraspPose-Opt.

B.2 Baselines to MotionFill-VAE

In Sec. 4.2, we compare our method with the convolution autoencoder network (CNN-AE) in [3], LEMO [12], and RouteNet and PoseNet from Wang *et al.* [11]. For RouteNet and PoseNet, we remove the scene encoding branch from [11] and adopt the same route encoding branch and pose encoding branch architecture design. We use the same body representation as ours in all these experiments.

C Additional Results

C.1 Ablation Study on GraspPose-Opt optimization losses

In Sec. 4.1 and Table 2, we have studied the effectiveness of our proposed GraspPose-Opt optimization loss design in Eq. 4 for optimizing human-object

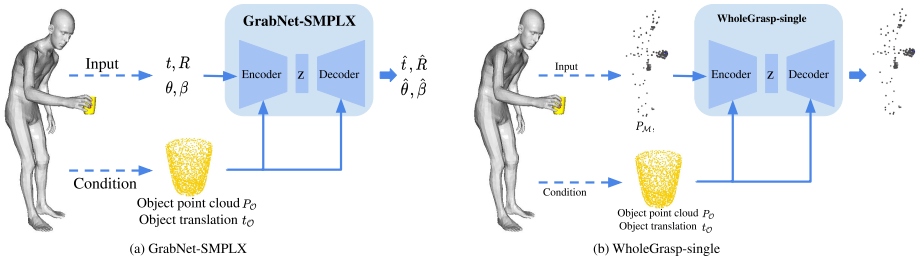

Fig. S3. (a) The schematic architecture of GrabNet-SMPLX baseline. Similar as in GrabNet, we build a CVAE model to directly generate SMPL-X parameters; (b) The Schematic pipeline of the WholeGrasp-single baseline, which merely learns positions information of body markers.

interactions. In Fig. S4, we also present the visualization results of optimized hand poses using different loss designs to show the effects of our proposed loss terms. Since the hand pose can be highly sensitive to even tiny noises in markers positions, using only the basic markers fitting loss and foot ground contact loss, the recovered hand pose from markers can hardly interact with the object in a perceptually realistic way (see visualization result in Fig. S4 (a)). While the object collision loss E_{colli} helps to mitigate the hand-object interpenetration issue (Fig. S4 (b)), the optimized hand does not grasp the object steadily. Using our mutual human-object contact loss E^o_{cont}, the object surface areas with higher contact probability attract hand vertices, and we can yield realistic and plausible hand-object interaction (see visualization result in Fig. S4 (c)).

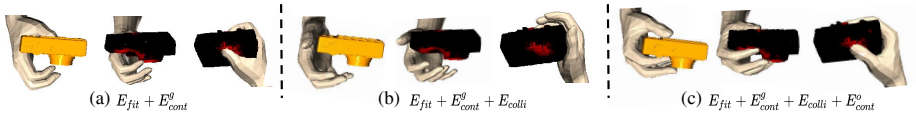

Fig. S4. Visualization results of ablation study on the GraspPose-Opt optimization loss design in Table 2. We present the optimized hand poses using different loss designs, and the red areas on the object surface indicate higher contact probability.

C.2 Additional Visualizations and Results on MotionFill-VAE

In Table 3, we have shown the quantitative results of our method compared with other state-of-the-art methods. In Fig. S5, we qualitatively present the diversity of the motions generated by our model which is finetuned on GRAB dataset [10]. The figure shows that our method can generate diverse trajectories as well as diverse local motions.

Limited by the short sequence length in the GRAB dataset, we only conduct the two-second motion infilling experiments with our MotionFill-VAE. Beyond

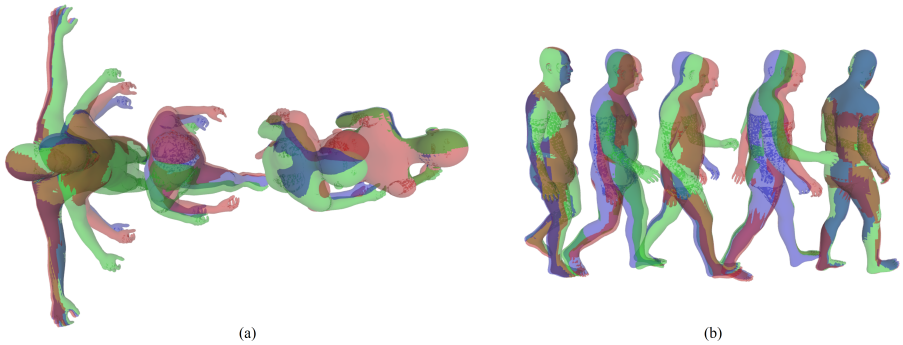

Fig. S5. Visualization on generated diverse motion sequences. (a) Three motion sequences on three different trajectories generated by TrajFill-CVAE, respectively. (b) Three motion sequences generated on the same ground truth trajectory. Different colors (red, green, blue) represent different motion sequences in each sub-figure. The diverse intermediate frames show the stochasticity of our TrajFill-CVAE and LocalMotionFill-CVAE.

generating two-second motion sequences given the starting pose and the ending pose, we show that our motion infilling model can be easily generalized to longer time lengths. Given the starting pose, ending pose, we train our MotionFill-VAE on AMASS dataset with 2s, 4s, 6s clips, respectively. In Fig. S6, we present the infilled motion sequences of 2 seconds, 4 seconds, and 6 seconds. The visualization results show that our motion infilling model is able to generate motions with different time lengths.

D Limitations

Although our method can generate realistic grasping poses and grasping motions for most of the unseen objects in our test set, we observe some failure cases where the synthesized human fails to grasp the object in a realistic way. We have the similar observation as mentioned in GrabNet [10], the frying pan is the most challenging object to grasp. As visualized in Fig. S7, though the generated humans are in contact with the pan, they typically fail to grasp the pan handle, resulting in perceptually unrealistic results and low perceptual score in Table S1.

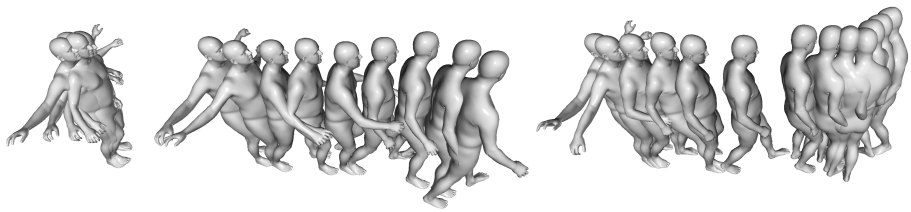

Fig. S6. Visualization on generated motion sequences with different time lengths (2s, 4s, 6s). (a) 2-second motion sequences. (b) 4-second motion sequences. (c) 6-second motion sequences. We train these three MotionFill-VAE models using training data with different time lengths on AMASS dataset [6]. The visualization results show that our motion infilling model can be easily generalized to different time horizons.

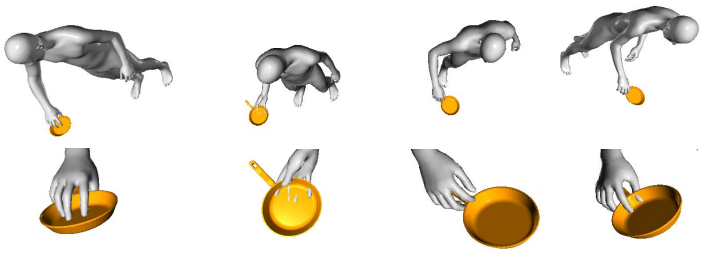

Fig. S7. Grasping pose random samples for grasping the frying pan. Generating realistic grasping poses for frying pan pan is challenging. Although the generated humans are still in contact with the pan, they typically fails to grasp the handle of the pan, resulting in perceptually unrealistic results.

References

1. Brahmbhatt, S., Ham, C., Kemp, C.C., Hays, J.: ContactDB: Analyzing and predicting grasp contact via thermal imaging. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
2. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3d annotation of hand and object poses. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3196–3206 (2020) 6
3. Kaufmann, M., Aksan, E., Song, J., Pece, F., Ziegler, R., Hilliges, O.: Convolutional autoencoders for human motion infilling. In: 2020 International Conference on 3D Vision (3DV). pp. 918–927. IEEE (2020) 8
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 5
5. Lucas, J., Tucker, G., Grosse, R.B., Norouzi, M.: Understanding posterior collapse in generative latent variable models. In: DGS@ICLR (2019) 4
6. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision. pp. 5442–5451 (Oct 2019) 11
7. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019) 4
8. Prokudin, S., Lassner, C., Romero, J.: Efficient learning on point clouds with basis point sets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 8
9. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems **30** (2017) 1, 2, 8
10. Taheri, O., Ghorbani, N., Black, M.J., Tzionas, D.: GRAB: A dataset of whole-body human grasping of objects. In: European Conference on Computer Vision (ECCV) (2020) 1, 3, 7, 9, 10
11. Wang, J., Xu, H., Xu, J., Liu, S., Wang, X.: Synthesizing long-term 3d human motion and interaction in 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9401–9411 (2021) 8
12. Zhang, S., Zhang, Y., Bogo, F., Pollefeys, M., Tang, S.: Learning motion priors for 4d human body capture in 3d scenes. In: IEEE/CVF International Conference on Computer Vision (ICCV 2021) (2021) 2, 3, 5, 8
13. Zhang, Y., Black, M.J., Tang, S.: We are more than our joints: Predicting how 3d bodies move. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3372–3382 (2021) 3, 5