Supplementary Material -– ContactGen: Generative Contact Modeling for Grasp Generation

Shaowei Liu[1†] Yang Zhou[2] Jimei Yang[2] Saurabh Gupta[1*] Shenlong Wang[1*]
[1]University of Illinois Urbana-Champaign [2]Adobe Research
https://stevenlsw.github.io/contactgen/

A. Interactive 3D Visualization

High-resolution qualitative results with interactive 3D hand and object meshes can be visualized at our project webpage https://stevenlsw.github.io/contactgen/.

B. Human Studies Setup

We presented 4 views for each grasp. We gathered responses from 10 participants and posed two questions for each sampled grasp: (1) The generated hand grasp is natural and realistic, what is your opinion? (2) The generated hand grasp is stable, what is your opinion? Participants rated these questions on a five-point scale, ranging from strongly disagree (0) to strongly agree (5).

Figure 1: Human studies interface. Participants were asked to rate the quality of each grasp based on its naturalness and the stability of holding the object using a five-point scale ranging from strongly disagree (0) to strongly agree (5).

C. Implementation details of hand SDF model

We train the piecewise hand SDF model following [4]. We use the same network architecture and the same loss function as [4]. Each part decoder consists of four fully-connected layers with 32 neurons each, employing LeakyReLU activation with a negative slope of 0.1 for each layer. We use Mano shape [8] as the fixed shape code shared by all part decoders. For each hand sample, we conducted uniform sampling of 7,000 points on the hand mesh surface, an additional 7,000 near-surface points generated by applying isotropic Gaussian noise with a mean of zero and a standard deviation of $\sigma = 0.01$ to each sampled surface point, along with 1,400 randomly selected off-surface points as per Gropp et al.'s approach [2]. For each on-surface sampled point, we first compute its barycentric coordinates relative to the mesh and corresponding skinning weights weighted by the neighborhood hand mesh vertices. We pick the top 2 highest skinning weights as the part label of the sampled point. The network is trained from scratch. We train it for 100 epochs with a learning rate $1e-4$ and Adam optimizer [3]. Once the network was trained, given the provided pose and shape code, we could compute the SDF with respect to a given query point. Subsequently, by employing the Marching Cubes algorithm [5], we could reconstruct each part under a specified pose and shape, a visual representation of which is presented in Fig. 2. For more detailed, high-resolution visualizations of each reconstructed 3D part model, please refer to our project page accessible at https://stevenlsw.github.io/contactgen/.

D. Network architecture

Our ContactGen CVAE comprises a common backbone and three sets of encoders and decoders for each aspect of the ContactGen. To extract features, we utilize the PointNet++ [7] SSG segmentation network as the backbone. This network consists of three sequential set abstraction layers and three feature propagation layers, forming the architec-

[†]Work started at an internship at Adobe Research.
[*]Equal advising, alphabetic order

Figure 2: Hand part visualization. We visualize each hand part from the piecewise hand SDF model output under the given pose of Fig. 2 in the paper. Each part is obtained by running Marching Cubes algorithm [5] at the top of corresponding SDF output.

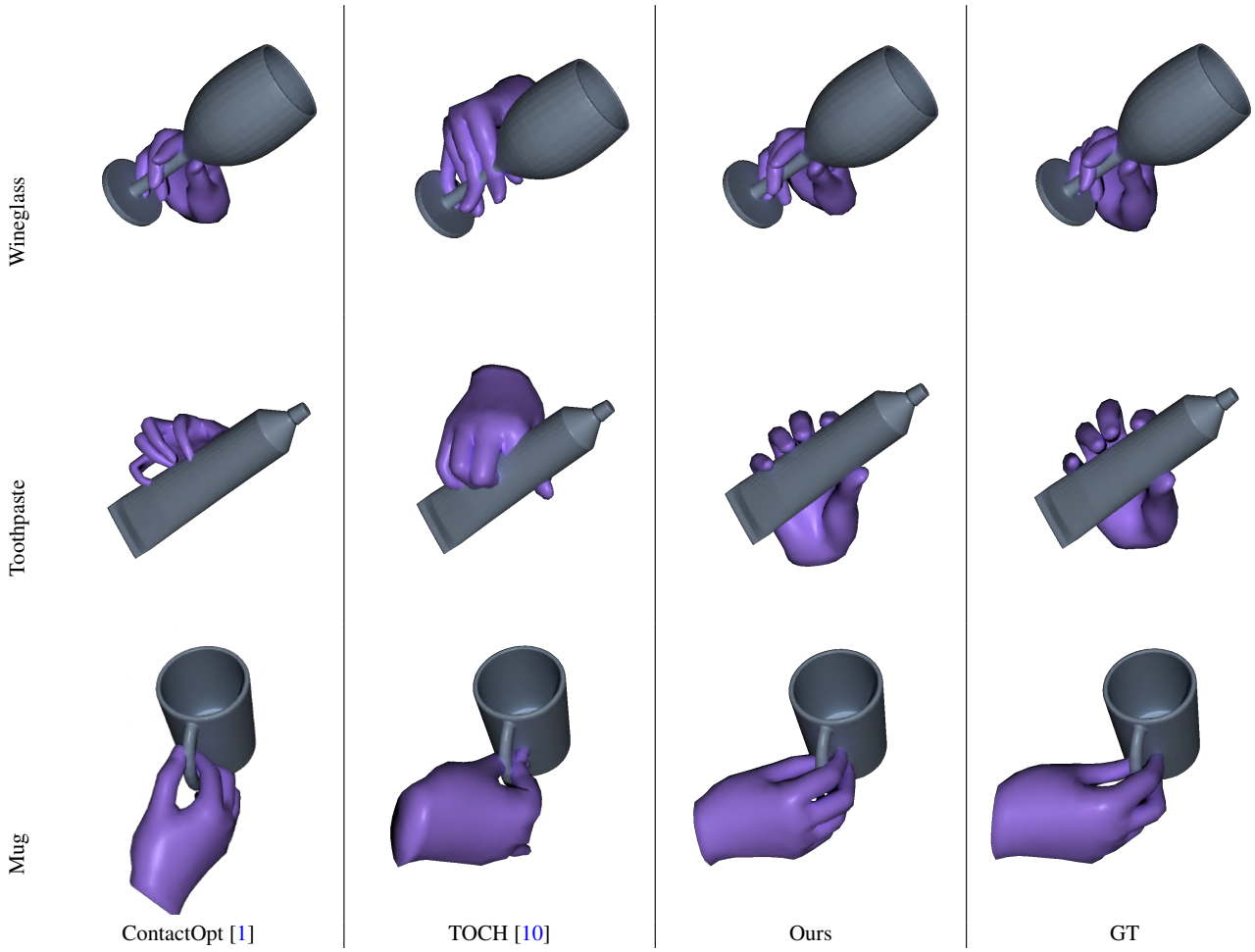

Figure 3: Contact Representation commparison against ContactOpt [1] and TOCH [10] on GRAB dataset [9]. Given the object and GT contact, we verify whether each method could recover the GT hand grasp. It can be seen both ContactOpt and TOCH exhibit failures in certain cases, whereas our method manages to achieve the closest reconstruction to the ground truth.

ture: SA(512, 0.2, [64, 128]) → SA(128, 0.4, [128, 256]) → SA([256, 512]) → FP(512, 256) → FP(256, 128) → FP(128, 64). Each encoder is implemented as a basic PointNet [6], comprising a shared MLP (64, 128, 256) applied to each point's feature and max pooling across points. Pooled features are then directed to another MLP (64, 256) to generate latent distribution parameters. The MLP incorporates LeakyReLU activation with a negative slope of 0.2. Following sampling of the latent code from the distribution, it is concatenated with each point's feature and sent to the respective decoder for prediction of each component map. The decoder architecture also employs the PointNet [6] approach, with the max pooling and MLP removed to yield pointwise predictions for each map. To capture hand-part features, we establish an embedding layer for each part with a feature dimension of 64. We feed the corresponding embedded feature of the part map into the network. For the contact map output, we pass the decoder's output through a Sigmoid layer to normalize the result within the [0, 1] range. For the part map output, we apply the argmax opera-

tion to determine the predicted hand part label. Finally, for the direction map output, we normalize each point's output to create a unit vector.

E. Contact Representation comparison

As discussed in Tab. 1 of the paper, we conducted a comparison between our proposed contact representation and the methods ContactOpt [1] and TOCH [10]. The aim was to assess their effectiveness in recovering accurate ground truth hand grasps. In Fig. 3, we offered a qualitative juxtaposition of each approach. From the visual comparison, ContactOpt and TOCH both exhibit inaccuracies in accurately recovering the actual hand pose, primarily because their representations lack completeness. In contrast, our approach offers a comprehensive representation of contact, enabling a full recovery of the ground truth hand pose.

References

[1] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1471–1481, 2021. 2, 3

[2] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Sandro Lombardi, Bangbang Yang, Tianxing Fan, Hujun Bao, Guofeng Zhang, Marc Pollefeys, and Zhaopeng Cui. Latenthuman: Shape-and-pose disentangled latent representation for human bodies. In *2021 International Conference on 3D Vision (3DV)*, pages 278–288. IEEE, 2021. 1

[5] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 1, 2

[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[8] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):245, 2017. 1

[9] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020. 2

[10] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Toch: Spatio-temporal object-to-hand correspondence for motion refinement. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 1–19. Springer, 2022. 2, 3