InterHandGen: Two-Hand Interaction Generation via Cascaded Reverse Diffusion

Supplementary Material

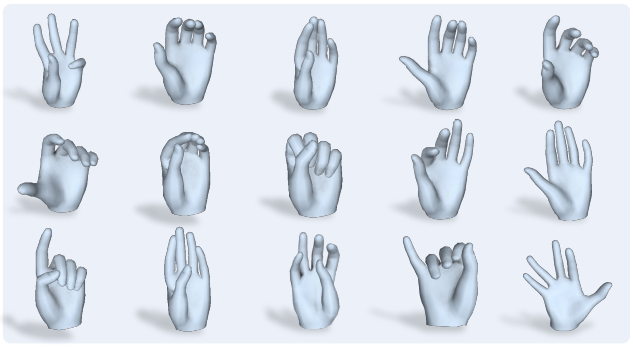
In this supplementary document, we first discuss the potential use of our method to build a universal hand prior (Section S.1) and show the additional qualitative results (Section S.2) of the experiments in the main paper. We then report additional experimental comparisons between parallel and cascaded generation approaches (Section S.3). Lastly, we report the implementation details (Section S.4).

S.1. Future Work: Universal Hand Prior

Due to the generality of our method, the proposed prior can be jointly trained with heterogeneous datasets to build a universal hand prior for all hand-related problems. Recall that our method learns the decomposed hand distributions using a single diffusion network via conditioning dropout. Since our network training (Algorithm 1 in the main paper) involves learning on both single-hand and two-hand training examples to model $p_\phi(\mathbf{x}_r)$ and $p_\phi(\mathbf{x}_r|\mathbf{x}_l)$, respectively, we can incorporate any existing single-hand datasets into the training as well. Taking a step further, we can also simultaneously apply dropout to the object condition $\mathbf{c}$ to model both object-conditional and unconditional (two-)hand distributions using a single diffusion network. Overall, our learning method based on the distribution decomposition along with conditioning dropout is naturally suited to build a multi-task prior trained with heterogeneous datasets (i.e., a single hand only, a single hand with an object, two hands, and two hands with an object).

While building a universal hand prior falls outside the scope of this work, we perform a toy experiment to showcase its possibility. We train our diffusion prior on two-hand dataset (InterHand2.6M [8]) along with *multiple single-hand datasets* [2, 19, 21, 22] and report the qualitative examples of two-hand and single-hand synthesis in Figures S1a and S1b, respectively. Sampling from our prior yields plausible single-hand and two-hand shapes. Importantly, this setting is shown to further boost the diversity of two-hand interaction synthesis (**from 3.59 to 4.39**) by exposing our prior to richer training examples. In Figure S1c, we also show the generation examples that could not be sampled using the prior trained on InterHand2.6M only. In particular, we collect the generated samples that are false positive with respect to the KNN manifold [13] modeled by the prior trained on InterHand2.6M only. As shown in the figure, these samples also model plausible two-hand interactions. One current limitation is that this universal prior does not necessarily improve the plausibility metric (e.g., FID, KID, precision) scores compared to individually trained priors. We hypothesize that existing datasets in each target domain such as InterHand2.6M [8] captures only the subset of the true distributions, and individual datasets share very little with each other to bring synergy to the joint learning. We leave building a more synergistic universal prior for future work.

(a) Two-hands sampled by our prior.



(b) Single-hands sampled by our prior.

(c) False positive samples with respect to the manifold [13] modeled by the prior trained on InterHand2.6M [8] only.

Figure S1. **Hands sampled by our prior trained on two-hand dataset [8] and additional single-hand datasets [2, 19, 21, 22].**

S.2. Additional Qualitative Results

S.2.1 Monocular Two-Hand Reconstruction

In Figure S2, we provide the qualitative comparison of our monocular two-hand reconstruction experiment in Section 4.3 in the main paper. In the figure, brown boxes highlight areas where shape penetration occurs, and blue boxes denote regions with inaccurate hand interaction (e.g., contact is absent where it should occur). While the baseline results of InterWild [7] contain several examples with penetration or inaccurate hand interaction, our approach can generate more plausible reconstructions. This indicates that leveraging our diffusion prior is effective in reducing ambiguity in an ill-posed monocular reconstruction problem.

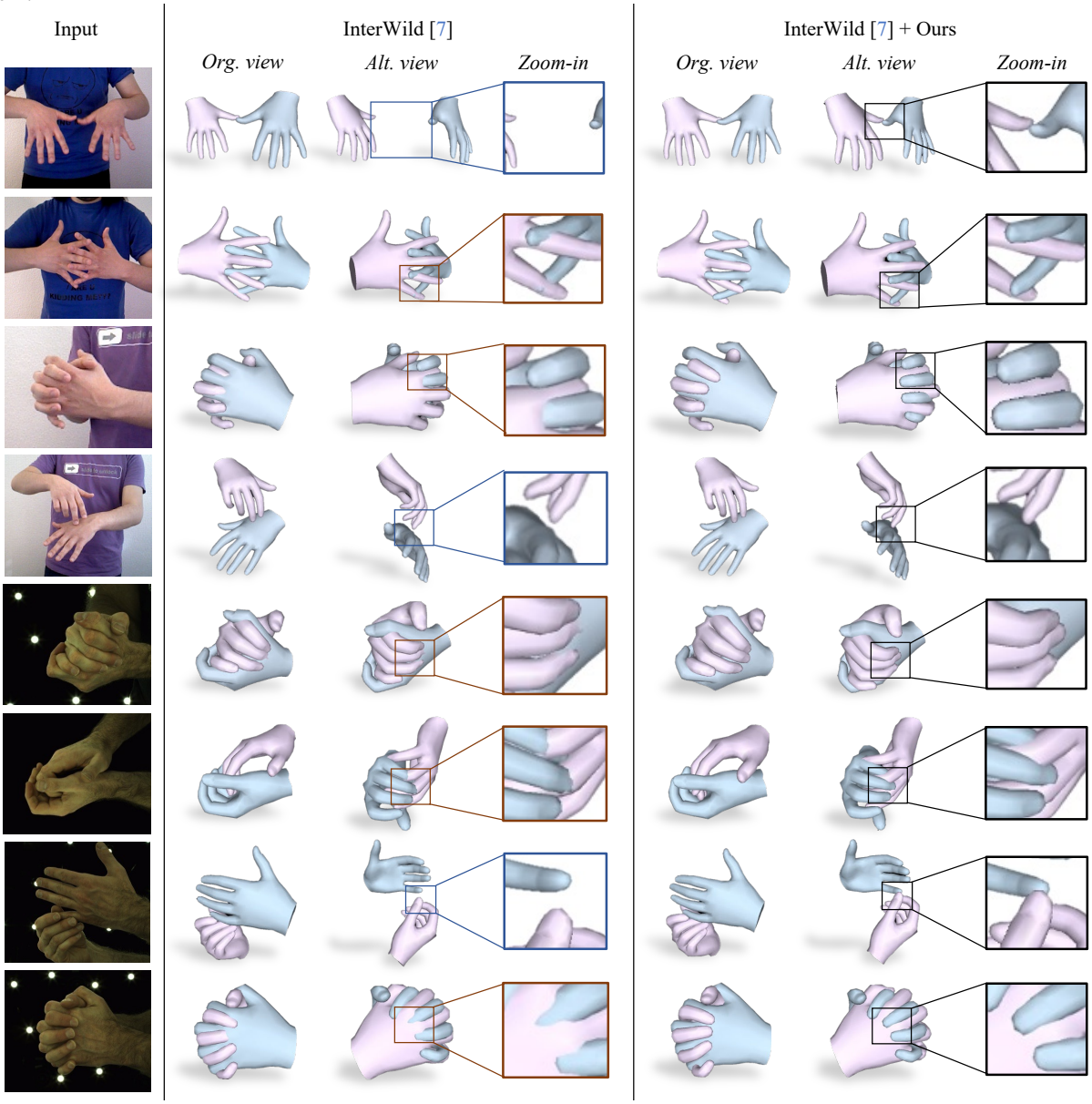


Figure S2. **Qualitative results of our monocular two-hand reconstruction experiment in Section 4.3.** The top four rows show results from the HIC dataset [17], while the bottom four rows show results from the InterHand2.6M dataset [8]. Brown boxes highlight areas where shape penetration occurs, and blue boxes denote regions with inaccurate hand interaction (e.g., contact is absent where it should occur). Utilizing our generative prior leads to more plausible reconstructions.

S.2.2 Two-Hand Interaction Synthesis

In Figure S3, we additionally show the qualitative comparison of two-hand interaction synthesis experiment in Section 4.1 in the main paper. In the figure, brown boxes denote regions with implausible two-hand interaction (e.g., where penetration or unnatural hand articulation occurs). Compared to the baselines, our method can produce more realistic two-hand interactions with less penetration. Especially, our method is shown to plausibly generate complex and tight two-hand interactions, for example, fingers of two hands crossing one another.

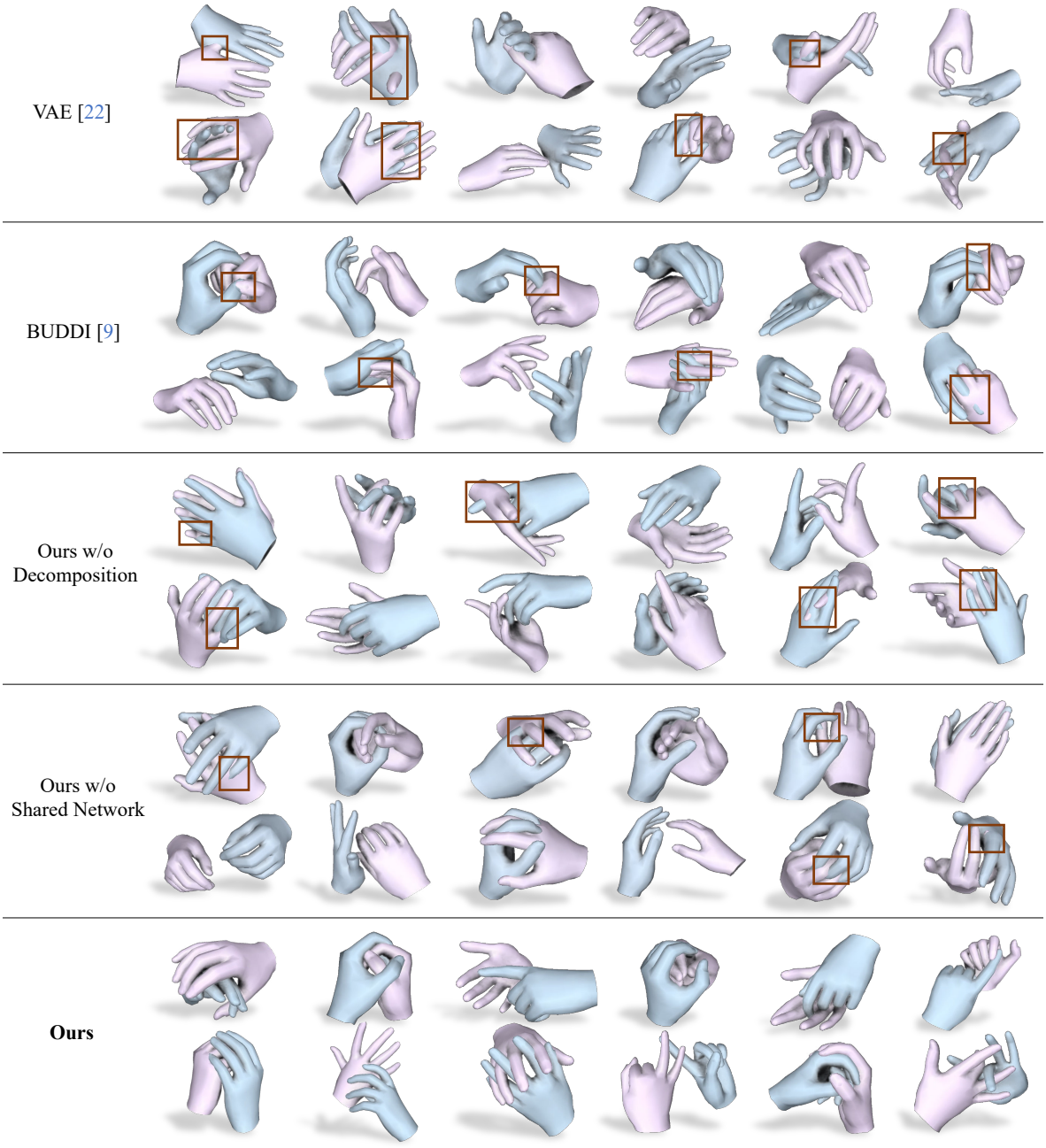


Figure S3. **Qualitative results of two-hand interaction synthesis experiment in Section 4.1.** Brown boxes denote regions with implausible two-hand interaction (e.g., where penetration or unnatural hand articulation occurs). Our method can produce more plausible two-hand interactions with less penetration.

S.2.3 Object-Conditioned Two-Hand Interaction Synthesis

In Figure S4, we also report the qualitative comparisons of object-conditional two-hand synthesis experiment in Section 4.2 in the main paper. Similar to the previous figures, brown boxes denote implausible regions with penetration or unnatural hand articulation. Our approach consistently demonstrates its capability to generate more plausible two-hand interactions, that are also closely adhering to the conditioning object.

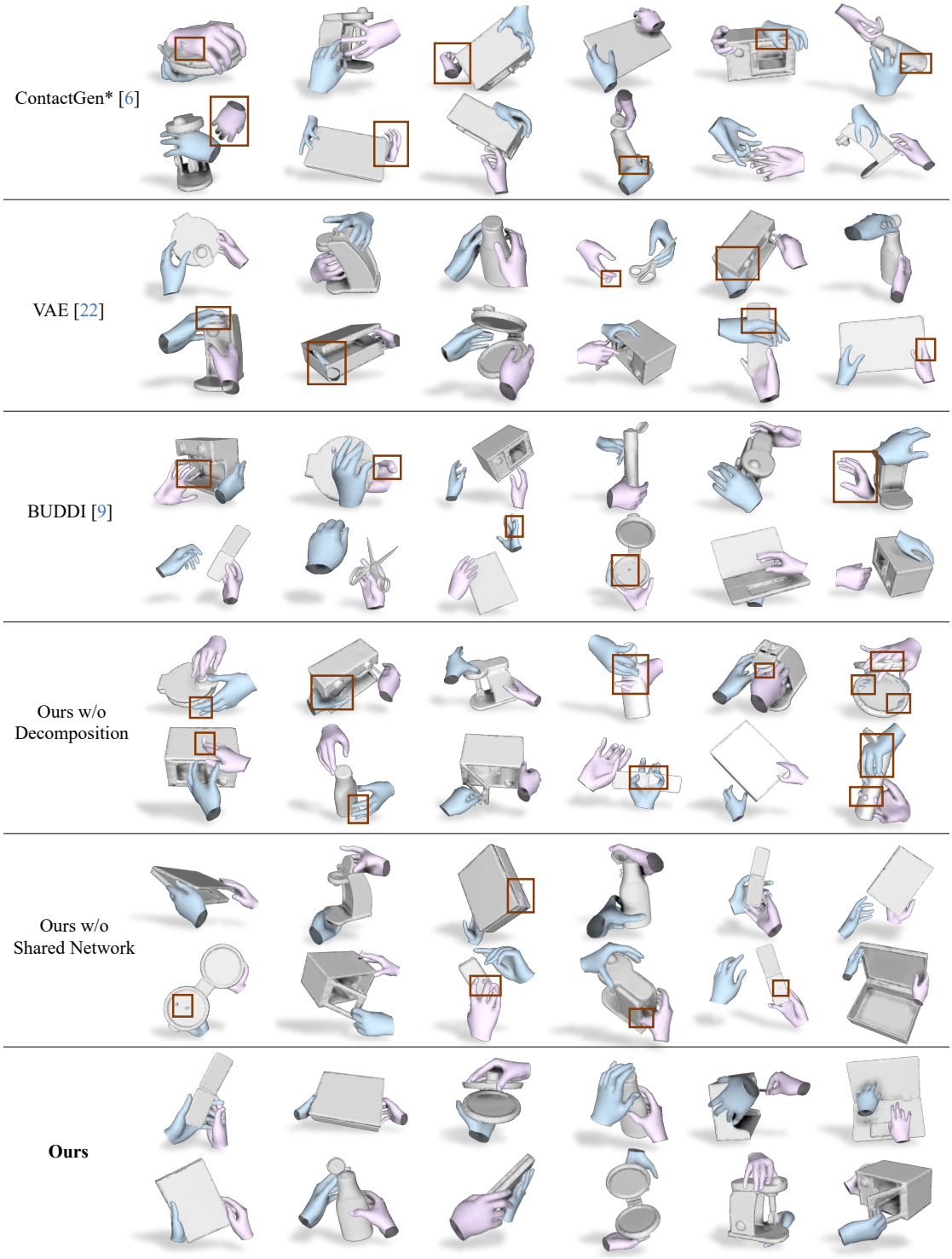


Figure S4. **Qualitative results of two-hand interaction synthesis experiment in Section 4.2.** Brown boxes denote implausible regions with penetration or unnatural hand articulation. Our approach can generate more realistic bimanual interactions.

S.3. Parallel vs. Cascaded Generation

We additionally show the experimental comparisons between our cascaded generation approach and the parallel two-human generation approach of ComMDM [14] modified for two-hand generation. Directly following [14], we added the ComMDM communication block to two parallel single-hand diffusion networks having shared parameters. We increased the number of attention layers by one to achieve better results, while the other hyperparameters remain the same as in [14]. As shown in Tab. S1, our cascaded approach leads to better generation quality due to (1) the reduced dimensionality of the generation target and (2) the conditioning on clean (rather than noisy) instances of another hand.

Table S1. **Comparisons between the parallel and cascaded generation approaches.**

Method	FHID (↓)	Precision (↑)	Diversity (↑)
Parallel (ComMDM [14])	2.19	0.75	2.68
Cascaded (Ours)	**1.00**	**0.86**	**3.59**

S.4. Implementation Details

We now report the implementation details for the reproducibility of the proposed method. Note that we also plan to publish our code after the review period.

S.4.1 Evaluation Protocol

Two-hand feature backbone. We modify PointNet++ [10] to regress (1) two hand poses in axis-angle representation, (2) relative root rotation in 6D rotation representation [20], and (3) relative root translation given a two-hand shape represented as a point cloud. Our network architecture mainly follows the architecture of the original PointNet++ encoder, except for the output dimension of the last fully connected layer modified to 108 (in order to match the concatenated dimension of our estimation targets). We train our network on InterHand2.6M [8] dataset for 200 epochs with a batch size of 32. Other training details (e.g., learning rate, batch size) remain unchanged from the original PointNet++ model. The test MPJPE of the resulting model is $1.49mm$.

Object-conditional two-hand feature backbone. The network architecture and training details are the same as those of our two-hand feature backbone, except that the network regresses (1) two-hand root rotations and translations in the object-centric coordinate space (not the relative root transformation between two hands) and that (2) the object feature is additionally incorporated to estimate two-hand poses. In particular, we use the PointNet++ [10] embedding module in our object-conditional diffusion model (refer to Section 3.6) to extract the object feature and feed it to the first fully connected layer of our two-hand pose regression network.

Evaluation metrics. We mainly follow the implementation details of the existing human pose and motion generation work [11, 16] for computing Fréchet Distance [3], Kernel Distance [1], diversity [11, 16] and precision-recall [13]. One important difference is that we adapt our own two-hand backbone network for feature extraction. For measuring penetration volume, we first voxelize two hand meshes with $1mm$ grids and count the number of voxels that are occupied by both hands similar to HALO [5].

S.4.2 Network Training and Inference

Training. We train our diffusion network for 80 epochs using an Adam optimizer with an initial learning rate of 2×10^{-4}. We additionally use a learning rate scheduler to decay the learning rate by 10% every 20 epochs. We set the batch size as 256 and 64 for unconditional and object-conditional diffusion networks, respectively. For diffusion noise scheduling, we use linear scheduling from $\beta_1 = 10^{-4}$ to $\beta_T = 0.01$ [4]. We set the maximum value of diffusion time as $T = 256$ and the probability of conditioning dropout as $p_{uncond} = 0.5$. Note that, for unconditional two-hand synthesis, only the relative root transformation between two hands is meaningful in modeling plausible interactions. Thus, we supervise the root transformation of the interacting hand generation ($p_\phi(\mathbf{x}_r|\mathbf{x}_l)$) with the ground truth transformation of $\mathbf{x}_r$ relative to $\mathbf{x}_l$, while not imposing supervision to the root transformation of the anchor hand generation ($p_\phi(\mathbf{x}_r)$). For object-conditional two-hand synthesis, we supervise both generation cases with the ground truth root transformations relative to the conditioning object.

Inference. For network inference, we use DDIM [15] sampling with 32 denoising steps. We set the classifier-free guidance weight as $w_{cfg} = 0.1$. For anti-penetration guidance weight w_{pen}, we use a multiplicative scheduling starting from 4 at $t = 0$ with a rate of 0.9. This strategy is adopted to avoid using a high weight for anti-penetration guidance in the early stages of the denoising process, where samples may still exhibit high levels of noise.

Mirroring transformation Γ [12]. We adopt the same mirroring transformation function $\Gamma(\cdot)$ used in MANO [12]. $\Gamma(\cdot)$ multiplies the input instance by the transformation matrix $\mathbf{T}$, which is defined as:

$$\mathbf{T} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Note that, for MANO hand shapes represented as MANO parameters, applying $\Gamma(\cdot)$ to the root rotation parameter is sufficient, as the local hand deformations are also mirrored along the MANO kinematic chain starting from the root

pose (please refer to [12] for more details on the MANO model).

S.4.3 Network Architecture

Hand embedding. For embedding noisy right-hand parameter $\mathbf{x}_t \in \mathbb{R}^{64}$ and conditioning left-hand parameter $\mathbf{x}_l \in \mathbb{R}^{64}$, we use two separate MLPs with the same network architecture. Each MLP consists of two fully connected layers, whose output feature dimensions are 2056 and 512, respectively. The first layer is followed by Swish activation. We denote the resulting embeddings for $\mathbf{x}_t$ and $\mathbf{x}_l$ by $emb_{\mathbf{x}_t}$, $emb_{\mathbf{x}_l} \in \mathbb{R}^{512}$, respectively.
Diffusion time embedding. For embedding diffusion time $t \in \mathbb{N}$, we use Sinusoidal embedding in DDPM [4] to extract a 512-dimensional feature. We then use an MLP (whose architecture is the same as the MLP used for hand embedding) to further extract the feature of t. We denote the resulting embedding for t by $emb_t \in \mathbb{R}^{512}$.
Object embedding. For embedding the object point cloud $\mathcal{O}$, we use a PointNet++ [10]-based architecture. We modify the original PointNet++ encoder by dropping the last layer and changing the final feature dimension from 256 to 512. Other implementation details remain unchanged from [10]. We denote the resulting embedding for $\mathcal{O}$ by $emb_{\mathcal{O}} \in \mathbb{R}^{512}$.
Transformer encoder. We perform channel-wise concatenation of $emb_{\mathbf{x}_t}$, $emb_{\mathbf{x}_l}$, emb_t, and (optionally) $emb_{\mathcal{O}}$ to consider each embedding as an input token to a transformer encoder. For the architecture of the transformer encoder, we use two self-attention blocks [18] with four attention heads. Each head consists of two fully connected layers, whose output feature dimensions are 2048 and 512, respectively. Each layer is followed by Layer Normalization, ReLU activation, and dropout with a rate of 0.1. After the self-attention modules, we use one fully connected layer to map the flattened output tokens into a global feature $emb_{glo} \in \mathbb{R}^{2056}$.
Output decoder. We use an MLP-based decoder to estimate the clean hand parameter $\mathbf{x}_r \in \mathbb{R}^{64}$ from emb_{glo}. The MLP consists of seven fully connected layers. The output feature dimension of all layers is 2056, except for the last layer whose output dimension is 64 to model the hand parameter. Each layer (except for the last layer) is followed by ReLU activation. Note that we use skip connections for all layers, in which the input feature is concatenated with the condition embeddings (i.e., $emb_{\mathbf{x}_l}$ emb_t and optional $emb_{\mathcal{O}}$). In the odd-numbered layers, we additionally concatenate the noisy hand embedding $emb_{\mathbf{x}_t}$ to the input feature.

S.4.4 Baseline Comparisons

Two-hand synthesis. For VAE [23] and BUDDI [9], we use the original network architectures with minor modifications to obtain better generation results on InterHand2.6M [8] dataset to perform fairer comparisons. For VAE, we empirically observed that increasing the feature dimension (from 128 to 256) and the number of encoder layers (from 4 to 5) improves the performance. For BUDDI, we increased the feature dimension of the self-attention blocks from 152 to 184 to obtain better generation results. For our method variations, we use the same implementation details except for the changes specified in Section 4.1.
Object-conditional two-hand synthesis. For BUDDI [9] and our method variations, we incorporate the object feature encoded by PointNet++ [10] as an additional token to the transformer encoder in a similar manner to our method. For VAE [23], we feed the object feature as an additional input to the second layer of both the encoder and decoder, similar to HALO [5]. For ContactGen [6], we extend the single-hand contact map to a two-hand contact map and optimize both hands accordingly.

References

[1] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 5

[2] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. Large-scale multiview 3d hand pose dataset. *Image Vis. Comput.*, 2019. 1

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 5

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 5, 6

[5] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *3DV*, 2021. 5, 6

[6] Shaowei Liu, Yang Zhou, Jimei Yang, Saurabh Gupta, and Shenlong Wang. Contactgen: Generative contact modeling for grasp generation. In *ICCV*, 2023. 6

[7] Gyeongsik Moon. Bringing inputs to shared domains for 3d interacting hands recovery in the wild. In *CVPR*, 2023. 2

[8] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *ECCV*, 2020. 1, 2, 5, 6

[9] Lea Müller, Vickie Ye, Georgios Pavlakos, Michael Black, and Angjoo Kanazawa. Generative proxemics: A prior for 3d social interaction from images. *CoRR*, abs/2306.09337, 2023. 6

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 5, 6

[11] Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. Modi: Unconditional motion synthesis from diverse data. In *CVPR*, 2023. 5

[12] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: modeling and capturing hands and bodies together. *ACM TOG*, 2017. 5, 6

[13] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018. 1, 5

[14] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *CoRR*, abs/2303.01418, 2023. 5

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 5

[16] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2022. 5

[17] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016. 2

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 6

[19] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, 2017. 1

[20] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 5

[21] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017. 1

[22] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019. 1

[23] Binghui Zuo, Zimeng Zhao, Wenqian Sun, Wei Xie, Zhou Xue, and Yangang Wang. Reconstructing interacting hands with interaction prior from monocular images. In *ICCV*, 2023. 6