# Technical Appendix for Force-Aware 3D Contact Modeling for Stable Grasp Generation

**Anonymous submission**

## Notations

Since the main paper involves physical laws in deductions, there are plenty of notations. We define them in Table. 1 to help understand the technique details and principles of the main paper. By default, for each sample, we use $i$ to represent the $i^{\text{th}}$ point sampled from the object surface, $j$ to represent the $j^{\text{th}}$ label, and $h$ to represent the $h^{\text{th}}$ hand part.

Additionally, there are several definitions to clarify. The affinity set $A_j$ for calculating the force labels is defined as:

$$A_j = \{i \in [1,n] | \text{argmin}_j d(\{\mathbf{p}_i, \mathbf{n}_i\}, \{\mathbf{c}_j, \mathbf{n}_j\}) = j\} \cap \{i \in [1,n] | C_i > C_{th}\}, \quad (1)$$

and the normal fixed distance is:

$$d(\{\mathbf{p}_1, \mathbf{n}_1\}, \{\mathbf{p}_2, \mathbf{n}_2\}) = \|\mathbf{p}_1 - \mathbf{p}_2\| + w(1 - \mathbf{n}_1^T \mathbf{n}_2) \quad (2)$$

where $w = 0.1$ in our experiment. The distance is used in calculating the contact maps and in part point clustering.

## Automatic Labeling Procedure

The labeling procedure consists of 3 parts: scene preparation, simulation and labeling. The input sample include a pair of interacting hand and object meshes, which are first decomposed to convex parts and stored in a MJCF file. The file was then read by the simulator (Todorov, Erez, and Tassa 2012) with different preset parameters, after which the force labels with the least center of mass (CoM) displacement is selected. Finally, the labels are obtained by selecting the label with the least acceleration under a displacement threshold.

### Preparation

The physics engine requires the input geometries to be composed of convex parts to determine collisions, so it is necessary to do convex decomposition. Considering the different geometric properties of hand and object meshes, we utilize different procedures.

Hand meshes are of fixed topology (MANO model (Romero, Tzionas, and Black 2022)), so the decomposition can be the same for all hands. To directly determine the hand part label of each labeled force, we decompose the hands in the same way as the part definition in the main paper. *i.e.*, if the blend-skinning weight matrix in MANO is $W \in \mathbb{R}^{778\times16}$, then the part label of the $i^{\text{th}}$ point is $h_i = \text{argmax}_j\{W_{ij}\}$.

After that, the hand mesh is partitioned according to the part labels and the resulting parts are nearly convex, so we directly take the convex hulls of the parts to get the convex decomposed result.

Object meshes are of different topologies, so we utilized a more general convex decomposition algorithm, *i.e.*, CoACD (Wei et al. 2022) to decompose.

The output of this step include a series of mesh files storing all the decomposed meshes, and a MJCF file storing the positions of all the decompositions.

### Simulation

The files from the previous step builds up the scene for simulation, yet we find it common in simulation when the object slips away from the hand when the GT label is intuitively a stable grasp. As stated in the main paper, this happens due to the unavoidable errors in hand surface deformation and imperfect approximation of the soft tissues of the hand, which are very complicated problems. Additionally, what we need is essentially one possible force label of the current hand grasp sample, and therefore keeping the object stable is enough.

Thus, we allow slight displacement and use parameter searching technique to find the parameter with the least displacement within $1\,\text{s}$ of simulation. The basic idea is very simple: if we can find a stable state with: 1) a small object CoM displacement; 2) reasonable simulation parameter settings, then the forces applied to the object at this state is one possible force combination of the current grasp. Based on this, we design the simulation procedure as follows:

For a certain set of parameters, the simulation is done by fixing the hand position and setting the object free. In each simulation timestep, the contact labels (including contact points and forces) and the object displacement are saved. The parameter to search for the simulation parameter is determined to be 'damp ratio' of the MuJoCo simulator, denoted by $r$, which directly affects the 'reference acceleration'. When penetrations are detected, a spring-damper model (Liu et al. 2013) is applied to generate compulsory force at each contact point. This parameter controls the stiffnesss of the models and thus the resulting forces.

The parameter searching is done in coarse-to-fine manner. The coarse searching points are uniformly distributed in

| Category | Notation | Domain | Description |
|---|---|---|---|
| General variables | $n$ | $\mathbb{Z}^+$ | The number of sampled points |
| | $n_{kp}$ | $\mathbb{N}$ | The number of chosen keypoints. |
| | $\mathcal{R}$ | set | The non-negative real nubmer set. $\mathcal{R} = \mathbb{R}^+ \cup \{0\}$ |
| Physical variables | $\mathbf{a}$ | $\mathbb{R}^3$ | Translational acceleration of the object |
| | $\boldsymbol{\alpha}$ | $\mathbb{R}^3$ | Rotational acceleration of the object |
| | $\mathbf{p}_{CoM}$ | $\mathbb{R}^3$ | Object center of mass. |
| | $F_i$ | $\mathcal{R}$ | The normal force value of point $i$ |
| | $\mathbf{F}$ | $\mathcal{R}^n$ | $\mathbf{F} = [F_1, F_2, \ldots, F_n]^T$ |
| | $\mathbf{F}_{fi}$ | $\mathbb{R}^3$ | The friction vector of point $i$ |
| | $\mu$ | $\mathcal{R}$ | The friction coefficient for all contact surfaces. Set to 1 in our experiments. |
| | $\gamma_i, \delta_i$ | $[-1, 1]$ | The relative friction along axes $\mathbf{b}_i$ and $\mathbf{t}_i$. $\mathbf{F}_{fi} = \mu\gamma_i\mathbf{b}_i + \mu\delta_i\mathbf{t}_i$ |
| Force labels | $N_j$ | $\mathcal{R}$ | The normal force value of label $j$ |
| | $\mathbf{c}_j$ | $\mathbb{R}^3$ | The position of labeled contact point in label $j$ |
| | $A_j$ | set | The affinity point set of label $j$. The definition is Eq. (1) |
| Geometric variables | $\mathbf{p}_i$ | $\mathbb{R}^3$ | The position of point $i$. |
| | $\mathbf{n}_i$ | $\mathbb{R}^3$ | The surface normal of sampled point $i$; $\|\mathbf{n}_i\| = 1$. |
| | $\mathbf{b}_i, \mathbf{t}_i$ | $\mathbb{R}^3$ | The two tangential axes perpendicular to $\mathbf{n}_i$; $\mathbf{n}_i, \mathbf{b}_i, \mathbf{t}_i$ form a unit orthogonal basis |
| | $d$ | $\mathbb{R}$ | Distance or object center of mass displacement. |
| Machine learning variables | $\mathbf{v}_F$ | $\{0, 1\}^s$ | The one-hot vector representation of the force in point $i$. |
| | $\mathcal{L}$ | $\mathbb{R}$ | The loss in training and optimization. |
| | $w$ | $\mathbb{R}$ | The coefficient applied to the loss term |

Table 1: Notation table of the main paper

logarithm space to cover a wider range of parameters:

$$r_i = \exp\left\{\frac{N-i}{N-1}\log(r_{\min}) + \frac{i-1}{N-1}\log(r_{\max})\right\}. \quad (3)$$

Suppose the optimal parameter in Eq. (3) is $r_{i_0}$, then the fine searching is done linearly by:

$$r'_i = \frac{N'-i}{N'-1}r'_{\min} + \frac{i-1}{N'-1}r'_{\max}; \quad (4)$$

$$r'_{\min} = \begin{cases} r_{i_0-1}, & \text{if } i_0 \geq 2; \\ 0.1r_{i_0}, & \text{otherwise} \end{cases}; \quad (5)$$

$$r'_{\max} = \begin{cases} r_{i_0+1}, & \text{if } i_0 \leq N-1 \\ 1.2r_{i_0}, & \text{otherwise.} \end{cases} \quad (6)$$

After this step, the simulation results from the parameter with the least displacement is returned as the resulting simulation sequence. In the labeling procedure, the parameters are $N = 25, N' = 20, r_{\min} = -3, r_{\max} = 3$.

## Labeling

For the optimal sequence, the final displacement $\mathbf{p}_T$ is directly used as part of the label. Then the frame with the least acceleration under displacement threshold $d_{th} = 5\,\mathrm{cm}$ is used as the force label. Specifically, we calculate acceleration by $a_i = \|\mathbf{p}_i + \mathbf{p}_{i+2} - 2\mathbf{p}_{i+1}\|$. Then the frame with the least acceleration is used as the label frame, where all force labels are output as the label for this sample.

Finally, the label consists of the local contact frame $O_{local} \in \mathbb{O}(3)$, the local force vector $\mathbf{F}_j$, the contact point $\mathbf{c}_j$ and the contact hand part label $h_j$. The first dimension of the local force vector is exactly the normal contact force $N_j$. We can obtain the global force vector by $O_{local}\mathbf{F}_j$ but in experiments we only need the normal force for training.

## Detailed Simulation Settings

In simulation using MuJoCo, the `timestep` is set to $1\,\mathrm{ms}$ and the total simulation time is $T = 1\,\mathrm{s}$, so the simulation takes 1000 steps. The impedance parameters are less strict than the default parameters to mimic the softness of hand tissues. Following the notations from official MuJoCo documentation, the parameters are $d_0 = 0.7, d_{width} = 0.9$, `width` $= 0.006$, `midpoint` $= 0.5$, and `power` $= 2$. The surface friction coefficients are set to the same for hands and all objects. The `margin` parameter, which indicates the minimum distance of two bodies to be considered in contact, is set to $3\,\mathrm{mm}$. The tangential, torsional, and rolling friction coefficients are $1$, $0.5$, and $0.01$ respectively. The object is connected to the worldbody with a free joint, where the damping ratio is set to 1. All other parameters are set to default.

## Label statistics

In Fig. 2, we plot the density curve of all labeled contact normal force values is shown in fig. 3 and the separate curves

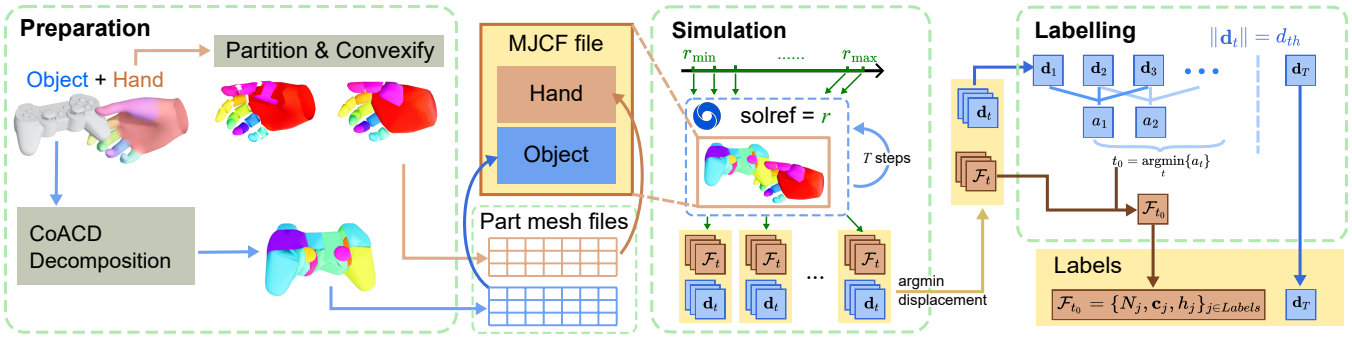



Figure 1: The labeling procedure

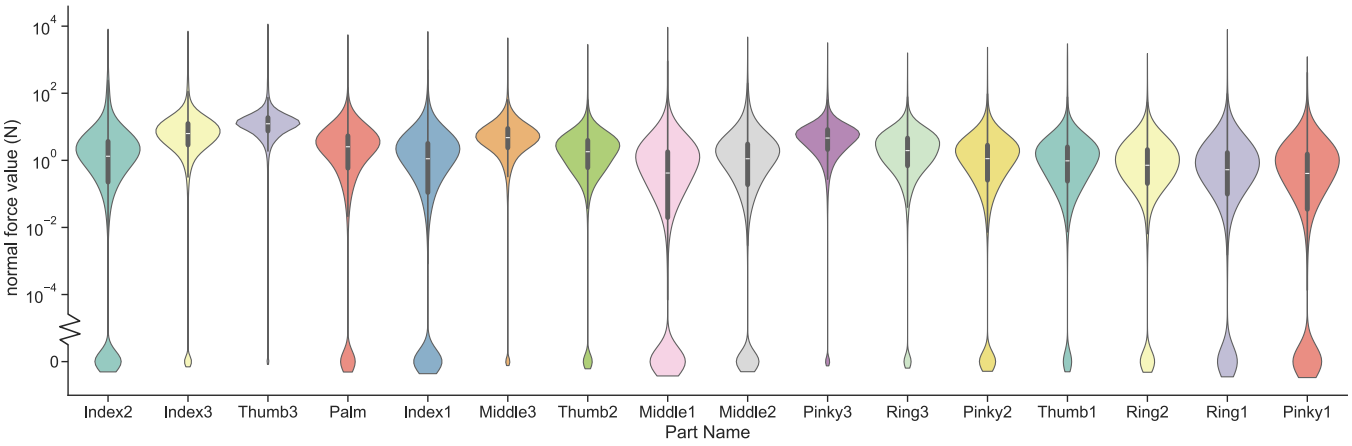



Figure 2: Distribution of all part forces in violin plot. For each violin, the outer bound marks the density curve of the data, the thick line in the middle stands for the 2nd and 3rd quarters of the data, and the white point marks the average.

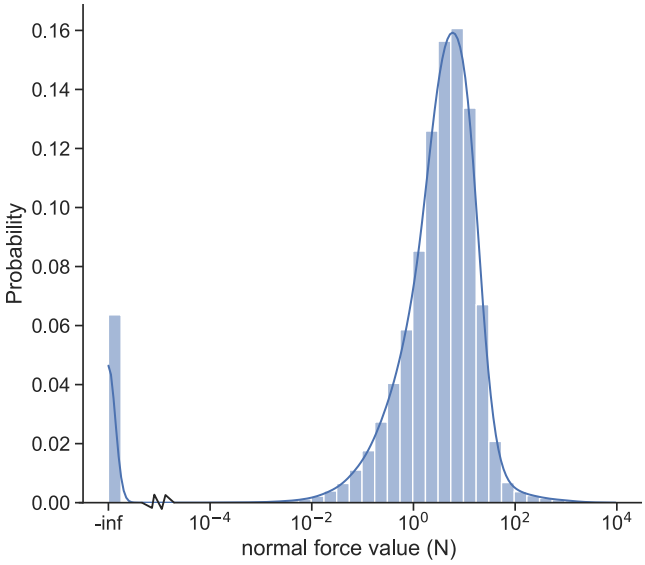



Figure 3: Distribution of all labeled forces.

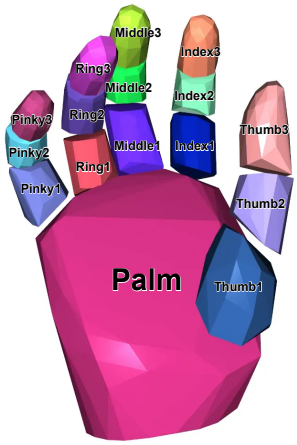



Figure 4: Hand part names

for forces on all hand parts are shown in Fig. 4. While the force labels of all parts spread a broad range just like the overall distribution shown in the main paper, we can still find some interesting conclusions by comparing the average: The tip of thumb ('Thumb3') has the largest average force, and the

5 finger tips provides the top-5 largest forces. This indicates that we human usually interact with objects by finger tips. The thumb, usually opposing the rest four fingers, provides the largest average forces in daily interactions.

Fig. 5 shows the distribution of all simulation displacements in the labeling process, where 60%$\sim$ 70% are from the first bin, with a displacement near to 0. Note that another peak appears at $\sim 350\,\mathrm{cm}$, which indicates that the object is not held at all and falls freely. The displacement is slightly smaller than that of a real $1\,\mathrm{s}$ free fall ($\sim 490\,\mathrm{cm}$) due to the non-zero damping of the simulator parameter.

Fig. 6 shows some labeled samples and most of the cases are reasonable. The bottom 4 samples are not stable in the simulator which can be expected by imaging the way of grasping. These samples are therefore assigned lower weights in training.

## Experiment Settings and Extra results

### Implementation Details

In our experiments, object meshes are sampled 2048 points. For the training set, samples with only one hand in contact are used, and sampled every 10 frames to filter out similar poses. All samples with left hand contacts are mirrored to right according to (Taheri et al. 2020). The cVAE is trained with Adam optimizer (Kingma and Ba 2015) for 100 epochs. The learning rate is $10^{-2}$ and batch size is 64. Each epoch takes $\sim 6\,\mathrm{min}$. The partitioned SDF hand model is the one trained by (Liu et al. 2023) on FreiHAND dataset (Zimmermann et al. 2019). In testing, we apply random rotations to the object, but in simulations the gravity is always $(0, 0, -9.81\,\mathrm{m/s^2})$. In pose optimization, the second initialization stage takes 200 iterations with a learning rat of $0.05$, and the optimization stage lasts 1000 iterations with a learning rate of $0.005$.

The hyperparameters in training are $w_{rec} = 1, w_{stability} = 2$ and in optimizations are $w_c = 0.1, w_{pene} = 0.5, w_{kp} = 0.2, w_{reg} = 0.01$. $w_{KL}$ changes with steps following (Liu et al. 2023). Training and testing are done on 2 NVIDIA RTX4090 GPUs and an Intel Xeon W7-3445 CPU.

### Runtime and Memory

Our method is based on iterative optimization, which is the major source of time and memory consumption. As our method requires the identification of keypoints before optimization and extra losses during optimization, we study the runtime and GPU memory consumptions and compare our method with previous state-of-the-art optimization-based method (Liu et al. 2023). We also set up a baseline method where only the contact map and hand part map are predicted. There are also no keypoints in baseline method. We report the average runtime of each batch of 20 samples and the allocated GPU memory in Tab. 2. All experiments are run using a single NVIDIA RTX4090 GPU.

In Tab. 2, we can see our method takes similar time and memory as ContactGen. Our method utilizes similar global initlization step with an extra step to determine keypoints, so the result indicates that searching for the optimal keypoint combinations is not computationally heavy compared to the optimization process. The runtime increase compared to baseline method is mainly due to the initialization step, which is also iterative. The increased memory usage is mainly caused by the extra branch predicting forces and calculating stability losses, which is very small compared to the overall memory allocated (0.04GB vs. 1.69GB).

| Method | Runtime (s / batch) | GPU Memory Alloc (GB) |
|---|---|---|
| Baseline | 36.0 | 1.65 |
| ContactGen (Liu et al. 2023) | 41.2 | 1.69 |
| Ours | 41.7 | 1.69 |

Table 2: Runtime and memory consumption analysis

### Predicted Force Samples

We show more examples of generated forces in Fig. 7. The ground truth is obtained by running the labeling pipeline. The samples also shows that forces predicted by our method is closer to the ground truth, supporting the conclusion in the main paper.

### Generated Grasp Samples

In this section, we show more randomly picked generated grasp samples while comparing our result to previous methods (GrabNet (Taheri et al. 2020) and ContactGen (Liu et al. 2023)) in Fig. 8, 9, and 10, where we can find more evidence that our method favors grasps supporting the object from the bottom, thus resulting in more stable and plausible grasps statistically.

## References


Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Liu, S.; Zhou, Y.; Yang, J.; Gupta, S.; and Wang, S. 2023. Contactgen: Generative contact modeling for grasp generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 20609–20620.

Liu, T.; Bargteil, A. W.; O'Brien, J. F.; and Kavan, L. 2013. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)*, 32(6): 1–7.

Romero, J.; Tzionas, D.; and Black, M. J. 2022. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*.

Taheri, O.; Ghorbani, N.; Black, M. J.; and Tzionas, D. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *European Conference on Computer Vision (ECCV)*.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wei, X.; Liu, M.; Ling, Z.; and Su, H. 2022. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4): 1–18.

Zimmermann, C.; Ceylan, D.; Yang, J.; Russell, B.; Argus, M.; and Brox, T. 2019. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 813–822.

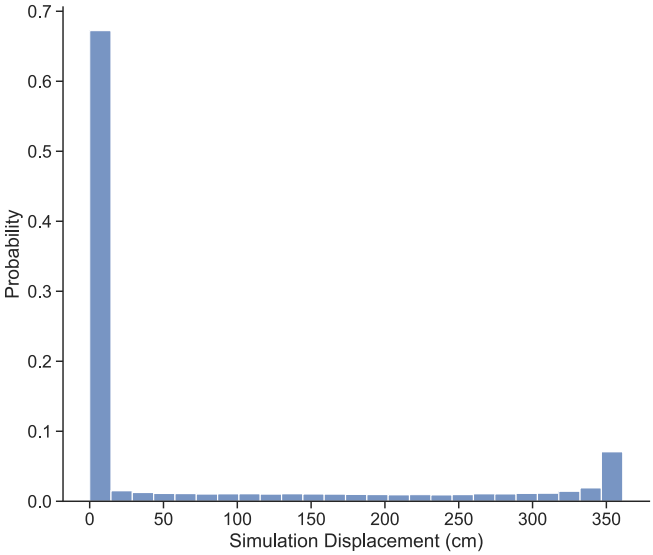



Figure 5: The histogram of simulation displacement

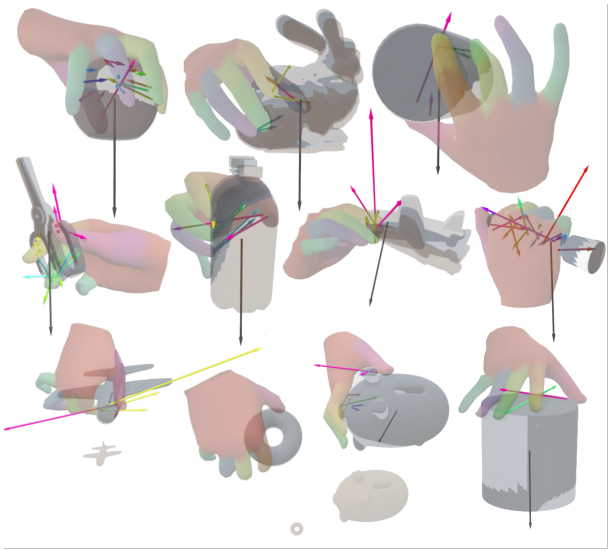


Figure 6: Some force label examples. The colorful arrows indicates the contact force directions and values, and the gray down arrows indicates the gravity. The bottom 4 samples are unstable while the rest are stable.

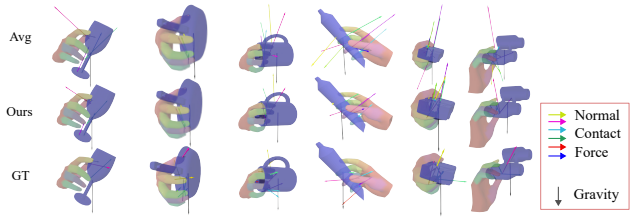
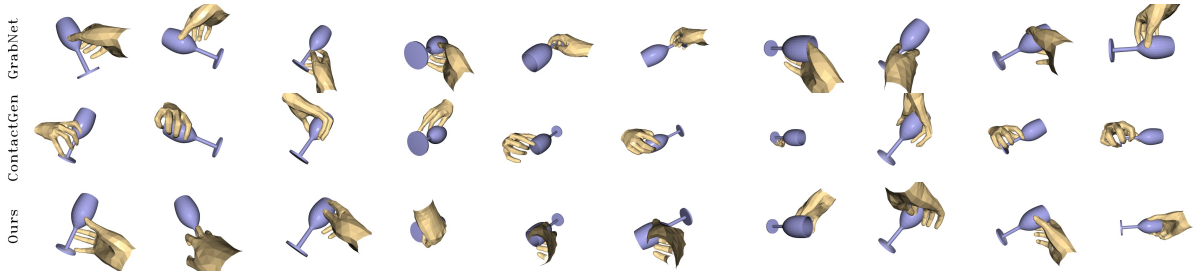
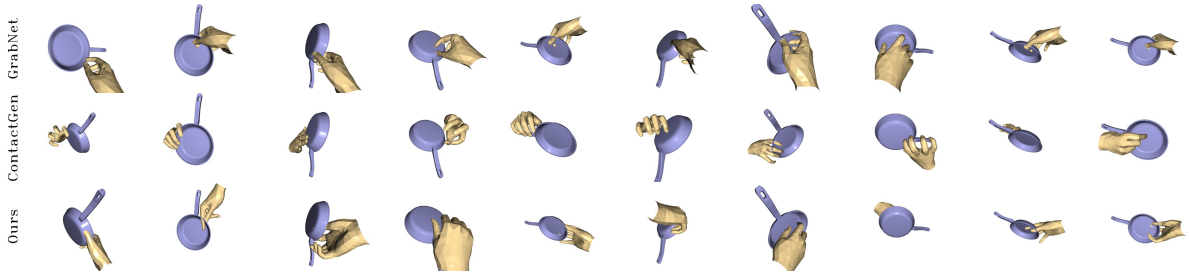


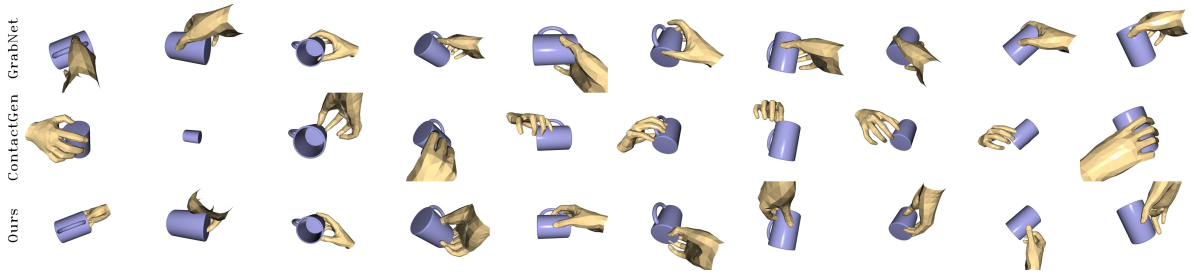

Figure 7: Some sampled force predictions. 'Avg' means the result of average predictor.



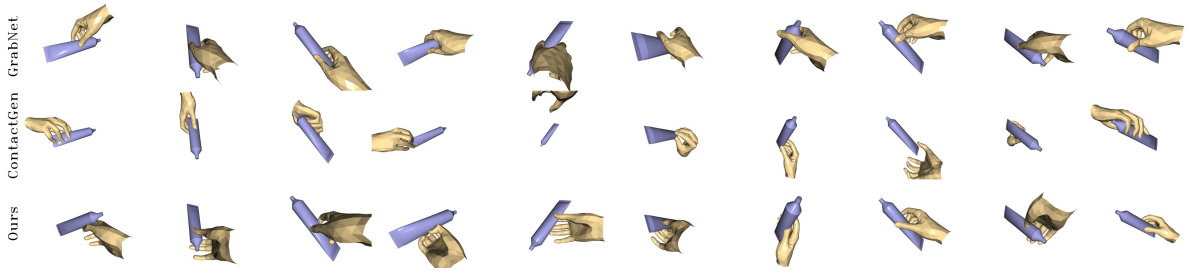

(a) Wineglass



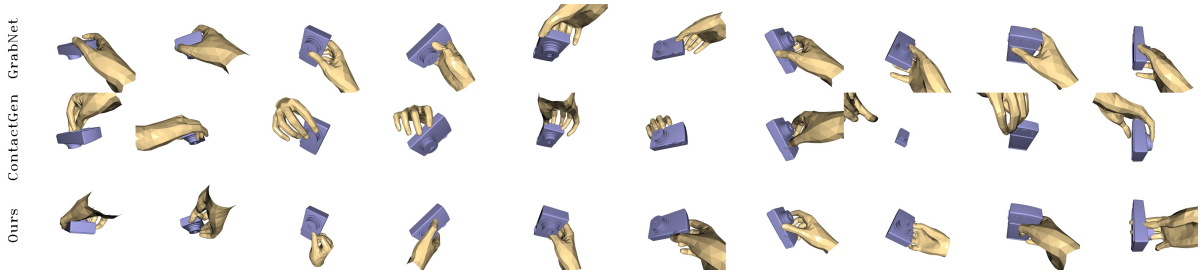

(b) Frying Pan



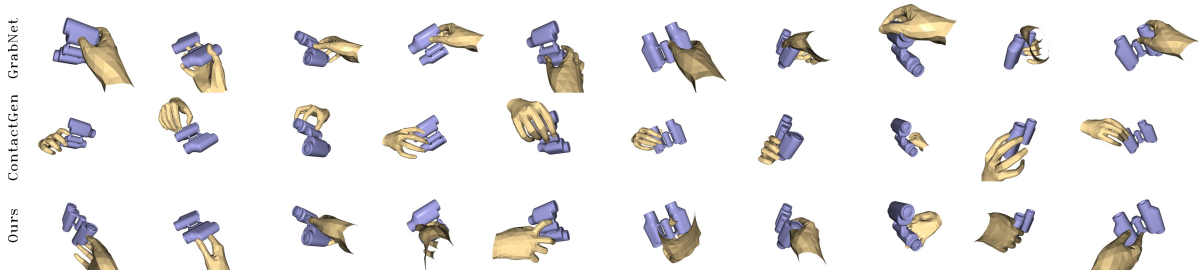

(c) Mug




(d) Tooth Paste

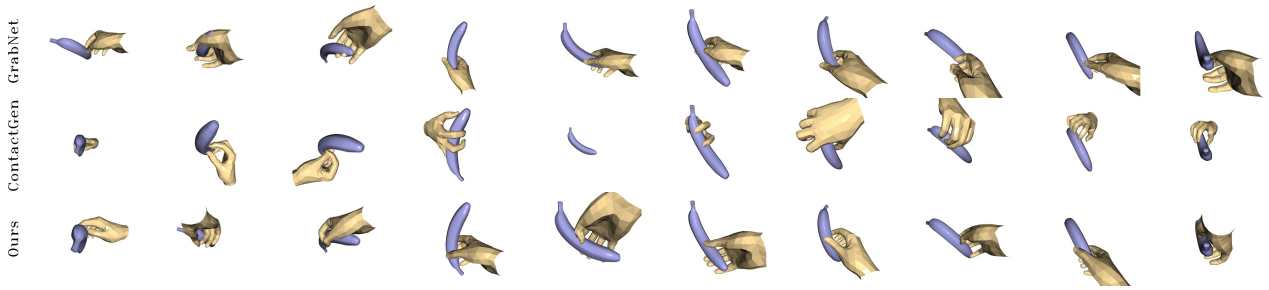



(e) Camera

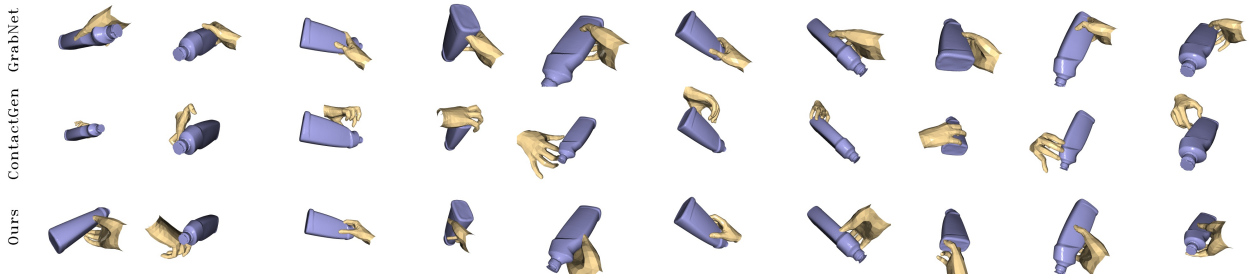



(f) Binoculars

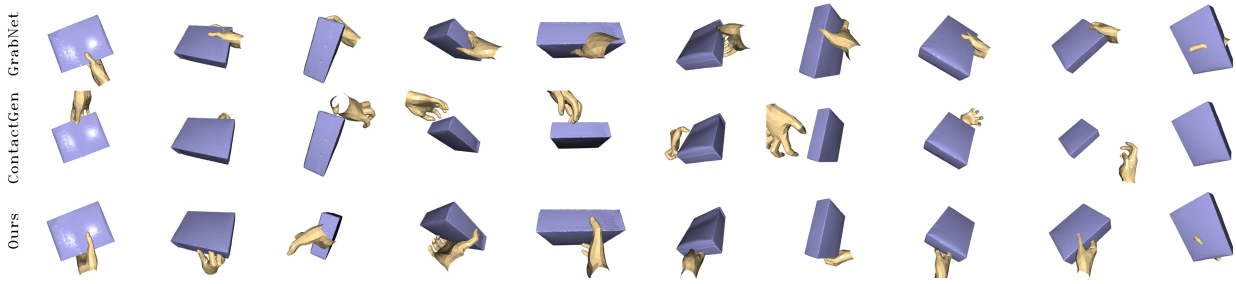
Figure 8: Extra Samples from GRAB dataset



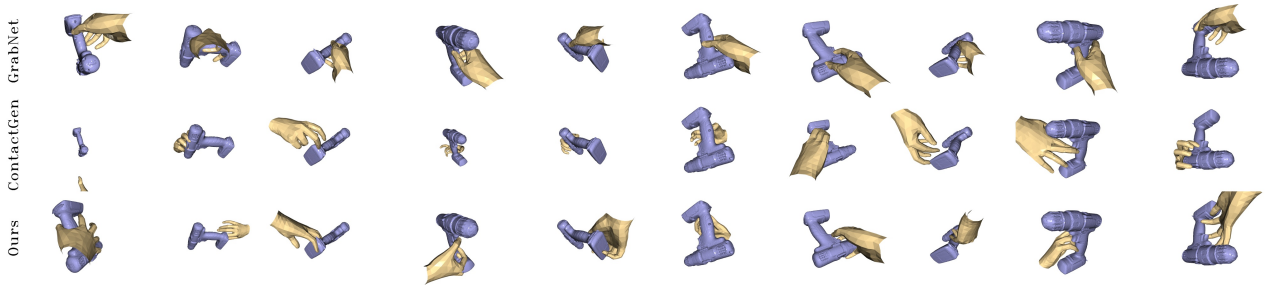

(a) Banana



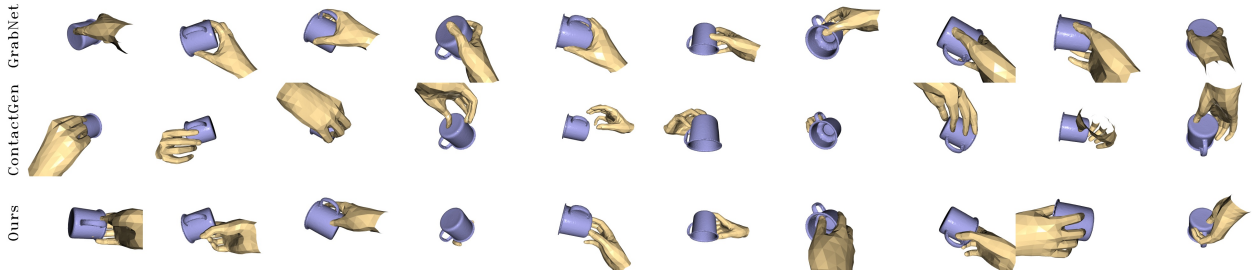

(b) Bleach Cleanser




(c) Cracker Box

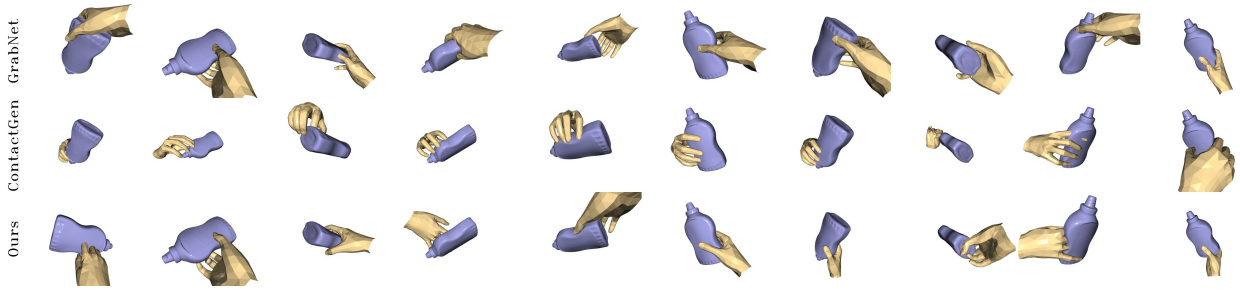



(d) Power Drill

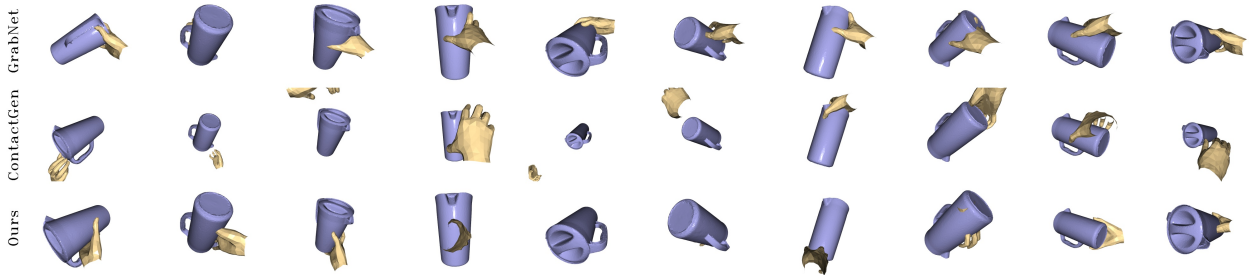



(e) Mug

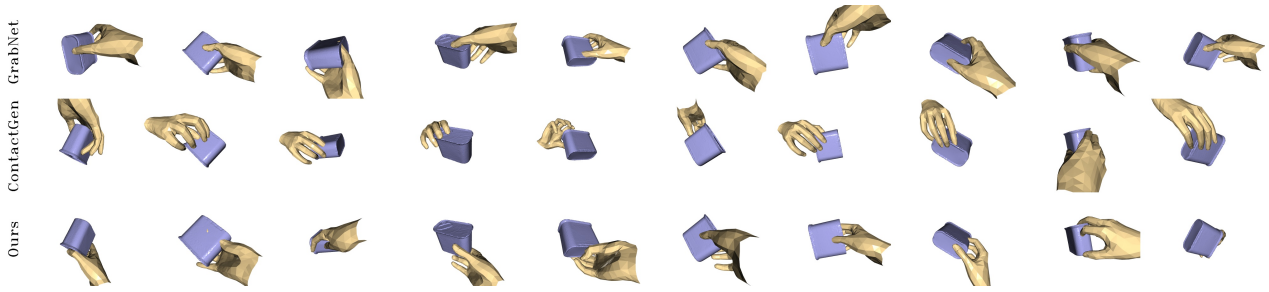
Figure 9: Extra Samples from HO3D dataset (1)



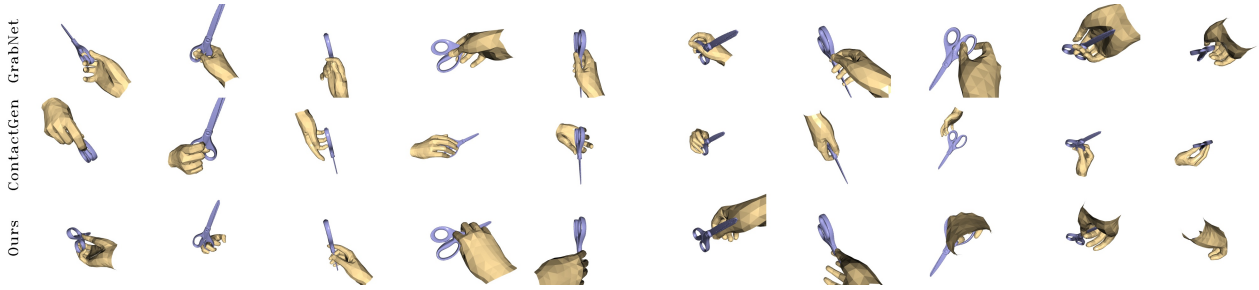

(a) Mustard Bottle



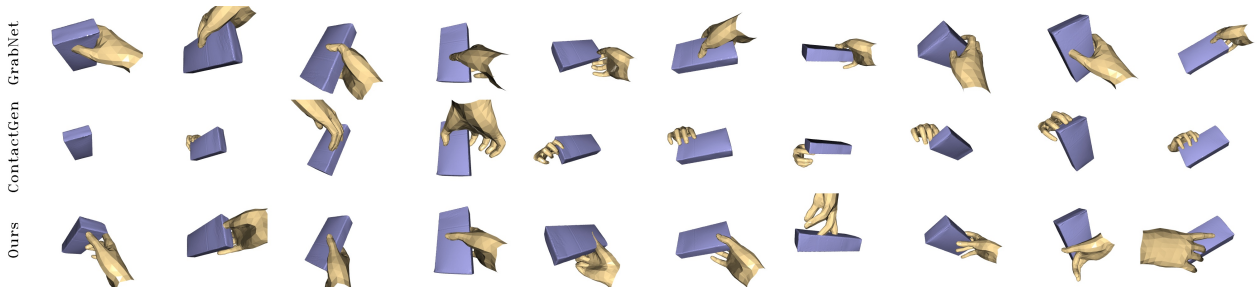

(b) Pitcher Base




(c) Potted Meat Can




(d) Scissors




(e) Sugar Box

Figure 10: Extra Samples from HO3D dataset (2)