D-Grasp: Physically Plausible Dynamic Grasp Synthesis for Hand-Object Interactions

SUPPLEMENTARY MATERIAL

Sammy Christen[1] Muhammed Kocabas[1,2] Emre Aksan[1]
Jemin Hwangbo[3] Jie Song[1†] Otmar Hilliges[1]
[1]Department of Computer Science, ETH Zurich [2]Max Planck Institute for Intelligent Systems, Tübingen
[3] Department of Mechanical Engineering, KAIST

The supplementary material of this paper includes **a video** and this document. We provide more detailed descriptions of our method in Section A and implementation details (physics simulation, baselines, metrics, and the learning algorithm) in Section B. Furthermore, we present additional qualitative results, as well as more detailed quantitative results in Section C. Lastly, we discuss potential societal impacts in Section D and provide a glossary for the notations used in this paper in Section E.

A. Method Details

We presented our method in Section 3. Importantly, we functionally separate the 6DoF global motion synthesis module from the grasping policy. We achieve this by explicitly separating the information flow in the feature extraction layers $\phi(\cdot)$ and $\psi(\cdot)$, similar to [4]. We show in Section 4.4 that this enables solving the complex *dynamic grasp synthesis* task. We now provide more details on the feature extraction layers.

A.1. Grasping Feature Extraction Details

We detail our method's grasping policy in Section 3.2. In this section, we provide additional details on how we extract the features of the goal space presented in Section 3.2.1. Hence, we need to extract object-relative features from the label $\mathbf{D}$ in order to be invariant to the object 6D pose during the grasping phase. Since collisions with the object occur when learning a grasp, it is crucial to have a representation that is flexible with respect to the object's pose, even when its position changes. We therefore focus on explaining the goal components $\mathbf{G} = [\widetilde{\mathbf{g}}_x|\widetilde{\mathbf{g}}_q|\mathbf{g}_c]$.

Relative target positions: The term $\widetilde{\mathbf{g}}_x$ measures the 3D distances between the hand's current and the target joint 3D positions $\mathbf{x}_h$ and $\overline{\mathbf{x}}_h$, respectively. Hence, to get the 3D target positions $\overline{\mathbf{x}}_h$, we utilize the label's information about the (global) 6D poses of the object $\overline{\mathbf{T}}_o$ and the hand $\overline{\mathbf{T}}_h$, as well as the target joint configuration $\overline{\mathbf{q}}_h$. Specifically, we use forward kinematics to compute the global target pose of the hand, which we then convert into the object-relative coordinate frame using $\overline{\mathbf{T}}_o$. This provides us with the 3D target positions $\overline{\mathbf{x}}_h$ for all the joints. We then apply the same procedure to the current state of the environment, using the object's current 6D pose $\mathbf{T}_o$, the hand's current 6D pose $\mathbf{T}_h$ and the hand's current joint configuration $\mathbf{q}_h$. This gives us the 3D joint positions of the current hand configuration $\mathbf{x}_h$ in the object-relative frame. Next, we measure the distance between the current and target joint positions:

$$\mathbf{g}_x = \overline{\mathbf{x}}_h - \mathbf{x}_h.$$

Our final step consists of transforming $\mathbf{g}_x$ into wrist-relative coordinates, finally providing us with $\widetilde{\mathbf{g}}_x$.

Relative target rotations: The term $\widetilde{\mathbf{g}}_q$ represents the angular distances between the current and target rotations for the joints and the wrist. For the local joint rotations, we can directly compute the distance between the current joint rotations $\mathbf{q}_h$ and the target joint rotations $\overline{\mathbf{q}}_h$. For the orientation of the wrist, we follow the abovementioned procedure to achieve invariance to the object pose. Hence, we convert the global 6D hand target pose $\overline{\mathbf{T}}_h$ into an object-relative target pose using $\overline{\mathbf{T}}_o$. We apply the same conversion to the current 6D hand pose $\mathbf{T}_h$ using the object's current 6D pose $\mathbf{T}_o$. We then compute the angular distance between the current and target object-relative poses. Finally, we transform the computed distance into wrist-relative frame for consistency.

†Corresponding author

Target contacts: The contact goal vector $\mathbf{g}_c = (\overline{\mathbf{g}}_c, \mathbb{I}_{\mathbf{f},\overline{\mathbf{g}}_c>0})$ is the concatenation of two vectors, namely the desired contacts $\overline{\mathbf{g}}_c$ and the term $\mathbb{I}_{\overline{\mathbf{g}}_c>0}$. To get the desired contacts for each hand joint from the grasp label, we measure the distance between all of a joint's vertices of the created meshes (Section 3.1) and all the vertices of the object mesh, which can be computed from the grasp label $\mathbf{D}$. Hence, for each joint j, the desired contacts are then determined as follows:

$$\mathbf{g}_{c,j} = \mathbb{I}\left[\sum_{i=1}^{I}\sum_{o=1}^{O}\mathbb{I}[\|\overline{\mathbf{v}}_\mathrm{i} - \overline{\mathbf{v}}_o\|^2 < \epsilon] > 0\right]. \quad (9)$$

If the distance between any vertex $\overline{\mathbf{v}}_\mathrm{i}$ of a joint j and an object vertex $\overline{\mathbf{v}}_o$ is below a small threshold ϵ (in our case 0.015m), we determine that the finger part should be in contact and hence the contact label should be equal to 1, otherwise 0.

The component $\mathbb{I}_{\overline{\mathbf{g}}_c>0}$ is a one-hot encoding vector indicating which of the desired contacts $\overline{\mathbf{g}}_c$ are active. Please note the redundancy in $\mathbf{g}_c$, which may be further improved in future work.

A.2. Motion Synthesis

Ours As described in Section 3.3, we use a closed-loop control scheme to move the hand from its current 6D pose $\mathbf{T}_h$ to the estimated hand pose $\widehat{\mathbf{T}}_h$. In particular, we compute the distance between the current and estimated target 6D object pose $\Delta\widehat{\mathbf{T}}_h = (\widehat{\mathbf{T}}_o - \mathbf{T}_o)$. This term is then added to the current 6D pose of the hand and weighted by a factor β:

$$\mathbf{T}_\mathrm{pd} = \mathbf{T}_h + \beta\Delta\widehat{\mathbf{T}}_h. \quad (10)$$

The term $\mathbf{T}_\mathrm{pd}$ is then sent to the PD-controller of the simulation. The output of the PD-controller are torques that generate a motion to guide the hand to the estimated target pose $\widehat{\mathbf{T}}_h$ by recomputing $\Delta\widehat{\mathbf{T}}_h$ after each simulation update. Note that in the *motion synthesis* phase, this module replaces the control of the first 6DoF of the grasping policy.

Ours+Learned Policy For the learned variant of the motion synthesis module, we propose a feature layer $\psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D})$ and a motion policy $\boldsymbol{\pi}_m(\mathbf{a}_m|\psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D}))$. Intuitively, it is not necessary for the motion policy to know about the proprioceptive information of the hand, such as joint angles and angular velocities. Therefore, we only extract features which are relevant to the global control of the 6D hand pose $\mathbf{T}_h$. The feature extraction layer $\psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D})$ receives the state $\mathbf{s}$ and the 6D target pose $\mathbf{T}_g$ of the object. The output of this layer is the following:

$$\psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D}) \equiv (\mathbf{T}_h, \dot{\mathbf{T}}_h, \mathbf{T}_o, \dot{\mathbf{T}}_o, \mathbf{g}_{o,x}, \mathbf{g}_{o,q}), \quad (11)$$

where the first four terms include information about the 6D poses and respective velocities of the hand and object. Crucially, the features $\mathbf{g}_{o,x}$ and $\mathbf{g}_{o,q}$ entail information about the object's current and target pose. The term $\mathbf{g}_{o,x}$ is the Euclidean distance between the object's current and target position $\mathbf{g}_{o,x} = \mathbf{T}_{o,x} - \mathbf{T}_{g,x}$ in global coordinates. Similarly, $\mathbf{g}_{o,q}$ computes the angular distance between the object's current and target pose $\mathbf{g}_{o,q} = \mathbf{T}_{o,q} - \mathbf{T}_{g,q}$. For motion synthesis, we use the following reward function:

$$r_m = \alpha_x r_{m,x} + \alpha_q r_{m,q}. \quad (12)$$

The position reward $r_{m,x} = e_\mathrm{mpe}$ measures the distance between the current and target object position (Eq. 13). The angular reward is the geodesic distance between the object's current and target orientation $r_{m,q} = e_\mathrm{geo}$ (Eq. 14). We weigh the two components with factors α_x and α_q.

In general, we propose a learning based variant because we believe it could come in as a viable solution when the control of the global hand pose becomes more complex. In the current work, we directly control the 6D pose of the hand. In such a setting, an IK-based solution is expected to outperform a learning-based variant. In the future, one could extend our method to include a biomechanical model of a full arm. This would add inherent constraints to the hand movements and hence increase the complexity of controlling the hand successfully. On the upside, this may lead to more natural movements during the *motion synthesis* phase. Hence, in such a setting a learning-based variant may outperform an IK-based solution.

B. Implementation Details

B.1. Physics Simulation

To train our method, we use a physics simulation as described in Section 3.1. We chose RaiSim [7], since it allows modeling non-convex meshes and efficient parallel training. We first create a controllable hand model (Fig. 5). Similar to [13], we compute the argmax of the skinning weights to assign each of the vertices to a body part. We then group the vertices accordingly and create a mesh for each body part. We limit the joint range in a data-driven manner. Specifically, we estimate the joint limits by parsing the DexYCB dataset and acquiring the maximum joint range, similar to [12]. Since the data may not contain the full range of possible joint displacements, we increase this limit by a slack constant. In practice, we found that approximating the collision bodies with primitive shapes (i.e., the simple objects and the hand meshes) led to an order of magnitude increase in training speed. This is because the simulation time increases roughly quadratically with the number of collision points. Therefore, for more complex object meshes, we apply a decimation technique to reduce the number of vertices (Fig. 6). For the simpler meshes, we use primitive shapes and mesh alignment as an approximation. For training and evaluation, we therefore use the simplified meshes (except for the interpenetration metric, see Section B.3).

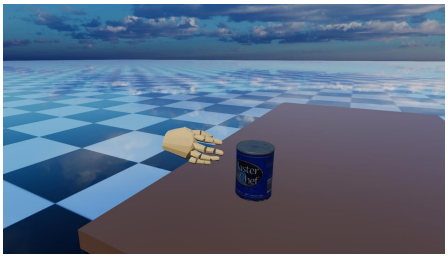

Figure 5. **Physics Simulation.** We create a controllable hand model and deploy it in the RaiSim physics-engine [7] to provide us with information about contacts and dynamics.

Figure 6. **Mesh Decimation.** We use mesh decimation to reduce the number of vertices of the object mesh. On the left is the original object mesh, on the right the decimated mesh. This helps to speed up the physics simulation during training.

B.2. Learning Algorithm

We train policies by using our own implementation of the widely used PPO algorithm [11]. We use the parameters summarized in Tab. 5 for training. We create a parallelized training scheme with a worker per grasp label for data gathering (amounting to e.g. 376 parallel environments for DexYCB). We then train a single policy over all objects, containing all grasps from the training set. For the GraspTTA [8] and ContactOpt [5] experiments, we double the amount of workers, such that they roughly correspond to the batch size of the DexYCB experiment (i.e., 400 workers with 2 workers for each label). Each training cycle utilizes a single GPU and 100 CPU cores and takes up to 24-72 hours of training.

B.3. Metrics Details

This section contains an extended description of the metrics depicted in Section 4.2.1.

Success Rate: We define the success rate as the primary measure of physical plausibility. It is measured as the rate of sequences which maintain a stable grasp, i.e., where the object does not slip and fall down for a period of a 5s window. We lower the surface in the simulation for this purpose. A success rate of 0.0 indicates no success, 1.0 means all sequences were successful.

Hyperparameters PPO	**Value**
Epochs	1e4
Steps per epoch	1.2e6
Environment steps for grasping	195
Environment steps for full task	300
Batch size	376
Updates per epoch	16
Simulation timestep	2.22e-3s
Simulation steps per action	13
Discount factor γ	0.996
GAE parameter λ	0.95
Clipping parameter	0.2
Max. gradient norm	0.5
Value loss coefficient	0.5
Entropy coefficient	0.0
Optimizer	Adam [9]
Learning rate	5e-4
Hidden units	128
Hidden layers	2
Weight Parameters	**Value**
w_x	-2.0
w_q	-0.1
w_c	1.0
$w_{\text{reg},h}$	0.5
$w_{\text{reg},o}$	1.0
$w_{x,j}$	1.0
$w_{x,\text{tip}}$	4.0
λ	5.0
α_x	-2.0
α_q	-0.25

Table 5. Hyperparameters of our method. The parameter "steps per epoch" is reported for the DexYCB training set with a batch size of 376. This number varies according on the amount of grasp labels available in the training set.

Interpenetration: We calculate the amount of hand volume that penetrates the object. To do so, we use the original MANO mesh [10] and the high-resolution object mesh. Hence, there is no physical simulation involved when measuring interpenetration. To ensure a fair comparison against the static baseline, we choose the last time step of the grasping phase for our method and hence omit the approaching phase from the evaluation.

Simulated Distance: Similar to the metric proposed in [8], we compute the mean displacement between the object and the hand's wrist. Instead of measuring the absolute displacement, we report the mean displacement in mm per second. We measure the displacement for a maximum window of 5s or stop whenever the object falls and hits the surface.

Contact Ratio: For the ablation study, we measure the ratio between the target contacts $\overline{\mathbf{g}}_c$ defined via the grasp label $\mathbf{D}$ and the contacts achieved in the physics simulation $\mathbb{I}[\mathbf{f} > 0]$. We average over the whole sequence, therefore both the ap-

proaching and grasping phase are contained in this metric.
MPE: This metric is used for the motion synthesis experiments. It is the mean position error between the object's 3D position and the object's target 3D position, defined as $\mathbf{g}_{o,x}$ (Section A.1):

$$e_{\text{mpe}} = \|\mathbf{g}_{o,x}\|^2 \tag{13}$$

Geodesic: This is the angular metric used in the motion synthesis experiments. In particular, the angular distance between the object's current orientation $\mathbf{T}_{o,q}$ and the object's target orientation $\mathbf{T}_{g,q}$. It is defined as follows:

$$e_{\text{geo}} = \text{acos}(0.5(\text{trace}(\mathbf{R}_o\mathbf{R}_g^\top) - 1)), \tag{14}$$

where $\mathbf{R}_o$ and $\mathbf{R}_g$ are the rotation matrices of the corresponding orientations of the object and the target 6D pose, respectively.

B.4. Baselines

Here we provide an extended description of the baselines.
***-PD:** Similar to [8], we place the object into the hand via the grasp label. We then attempt to maintain the grasp using PD-control in the physics simulation. To do so, the hand's 6DoF global pose $\mathbf{T}_h$ and the joint configuration $\mathbf{q}_h$ are initialized with the grasp label reference directly, hence $\mathbf{T}_h = \overline{\mathbf{T}}_h$ and $\mathbf{q}_h = \overline{\mathbf{q}}_h$.
***-IK:** We employ an offline optimization to correct for imperfections (i.e., minor distances or penetrations) in the label by utilizing the information about the target contacts $\overline{\mathbf{g}}_c$ (Section A.1) and the closest points on the object surface. In particular, for the finger parts that we deem to be in contact, we replace the original 3D keypoints from the grasp label $\overline{\mathbf{x}}_h$ by the closest vertex points on the object surface. We then run an optimization to yield a corrected target pose. The reconstructed samples are then passed to the PD-control. We found this technique to be effective for motion capture data, but not for the labels from GraspTTA [8] or ContactOpt [5], likely because both methods already inherently optimize for contact. Hence, we omit it for the latter methods in the main text.
Flat-RL: We employ an RL baseline that does not separate the grasping from the motion synthesis phase, but trains the full dynamic grasp synthesis task end-to-end. In particular, this baseline uses the concatenation of the grasping policy's feature layer $\phi(\mathbf{s}, \mathbf{D})$ (Section 3.2.1) and the feature layer of the learned motion synthesis module $\psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D})$ (A.2). Hence, the policy in this case is $\boldsymbol{\pi}(\mathbf{a}|\phi(\mathbf{s}, \mathbf{D}), \psi(\mathbf{s}, \mathbf{T}_g, \mathbf{D}))$. For the reward function we use the combination of the reward used for the grasping policy (Eq. 4 in main paper) and the reward for the decoupled motion synthesis policy (Eq. 12). The weights of the different reward components are reported in Tab. 5.

B.5. Experimental Details

Here we provide a short overview of the different object sets and grasp labels used in each experiment.

Grasping Objects When using grasp predictions from an external grasp synthesis method [8] (Section 4.3), we train with the objects used in DexYCB [2]. During evaluation, we report results on both the HO3D subset as done in [8] and the objects from DexYCB. For the experiment with ContactOpt [5], we train and test on the HO3D objects (except for 019 pitcher base, which is not contained in the dataset). Note that since the models for grasp synthesis and the image-based pose estimates have no notion of physics in terms of where an object is positioned in space (in contrast to the data from DexYCB), we apply a small modification to the simulation to ensure a fair comparison. We place the object on a surface and allow the hand to approach from any direction, even penetrating the surface. We achieve this by disabling the collision response between the surface and the hand. In future work, an optimization could filter out poses that require approaching from beneath a surface. Also note that since we only have access to a single grasp reference and not a sequence for GraspTTA and ContactOpt, we start each sequence at a predefined distance away from the object in the mean MANO hand pose.
For the evaluation of our method in this experiment, we remove the surface (i.e. table) after the *grasping* phase. The metrics are being measured from the moment the table is removed. For the baselines, we directly start the sequence in the target pose of both the hand and object (without a table present).

Motion Synthesis For the experiment presented in Section 4.4, we included a representative subset of YCB [1] objects. Namely, we used 2 cylindric objects (002 master chef can and 007 tuna can), 2 box-shaped objects (004 sugarbox and 061 foam), and 2 more complex objects (019 pitcher base and 052 extra large clamp) for training and evaluation. We use our train-split of DexYCB in this experiment. Furthermore, we filter out the failed grasps from the experiment in Section 4.3 and train and evaluate only on the stable grasps. Using unsuccessful grasps in this case would not produce any viable motions, since the objects cannot be grasped correctly to initiate the *motion synthesis* phase. Each sequence starts with the *grasping* phase, where only the grasping policy $\boldsymbol{\pi}_g$ is active. This ensures that a stable grasp on the object can be reached before moving the object globally. In the subsequent *motion synthesis* phase, both the grasping policy and the motion synthesis module are acting simultaneously.

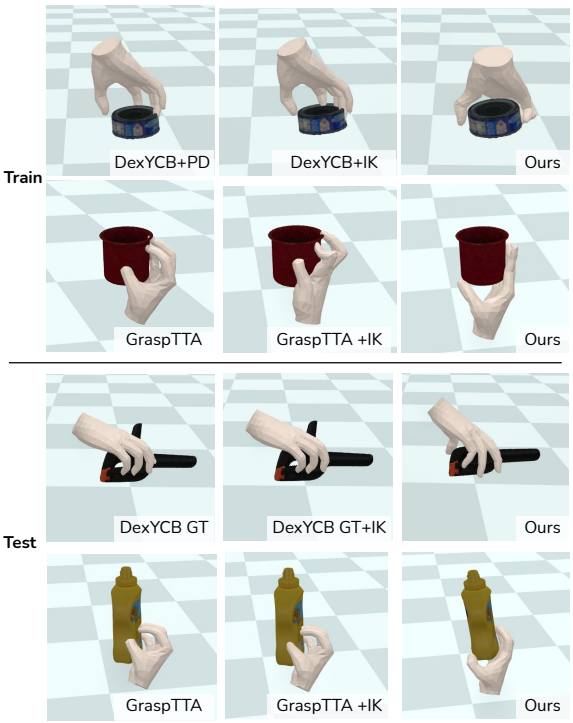

Figure 7. **Additional Qualitative Grasps.** We provide additional qualitative examples of grasps. Rows 1-2: Comparison of the grasps on the training-sets of DexYCB [2] and the generated grasps from [8]. Rows 3-4: Comparison on the test-sets of DexYCB [2] and the generated grasps from [8]. As shown, our method produces more physically plausible grasps, i.e., with less interpenetration and more realistic contacts than the baselines.

Ablations For the experiment presented in Section 4.5, we use a subset of YCB [1] objects and train per-object policies with grasp labels extracted from DexYCB [2]. In particular, we included one cylindric object (002 master chef can), one box-shaped object (004 sugarbox) and one complex object (052 extra large clamp) for training and evaluation.

C. Additional Results

All-Object vs. Per-Object Policies We experimented both with a single policy trained over multiple objects and single policies trained per object. A comparison on the DexYCB train/test-split is shown in Table 6. We observe that the single object policies (Ours PO) can be trained faster (i.e. ∼3000 vs. 10'000 epochs) and yield a better overall performance on the training labels, likely due to overfitting. On the other hand, the more general all-object policies (Ours AO) take longer to train, however, the generalization performance on unseen grasp labels is better. The performance on the training data is lower compared to the per-object policies. This result indicates that an all-object policy helps to prevent the policy from overfitting to single grasp references.

	Models	Success ↑	SimDist [mm/s] ↓	Interpenetration ↓
Train	Ours PO	**0.81**	$\mathbf{3.7 \pm 5.8}$	1.94
	Ours AO	0.7	5.8 ± 7.4	**1.75**
Test	Ours PO	0.42	12.4 ± 10.4	**1.19**
	Ours AO	**0.63**	$\mathbf{8.0 \pm 8.1}$	1.77

Table 6. **Policy Type Comparison**. We compare a single policy trained over multiple objects (Ours AO) and single policies trained per object (Ours PO). We find that the all-object policies lead to better generalization performance on the DexYCB dataset [2].

Additional Qualitative Grasping Results We provide additional qualitative results in Fig. 7. Specifically, we include examples on the training sets of DexYCB [2] and the generated grasps [8]. Moreover, we present additional examples for both test-sets. As can be observed, our method can correct for interpenetration and achieve more realistic grasps.

Quantitative Grasping Result Details We present the results of the empirical evaluation per object in Tables 7-12. It allows us to analyze the results in more detail. For the grasp evaluation experiment (Section 4.3), we find that the main difficulty for our learned policy are thin objects which are hard to pick up from the surface, e.g., grasping a pair of scissors from a table. This is indicated by the relatively low success rates in Tables 7 and 8 for the "037 scissors" and "040 large marker" objects. Grasping these objects requires very fine-grained finger motion or creating a distinct motion to pick them up, which involves sliding the object along the surface to overcome static friction. We find that this issue is mitigated partially in the experiment with generated grasp labels (Tables 9 and 10), because the deactivated collisions of the hand with the surface (see Section B.5) help to achieve stable grasps.

For the baselines, we occasionally observe a configuration that leads to high success rates despite noisy pose references. Specifically, if the interpenetration is large (e.g. GT-IK in Table 8 for "021 bleach cleanser" or "024 bowl"), the objects can become entangled within the hand mesh and will therefore not be able to fall down. Thus, the success rate metric should always be interpreted in combination with the other metrics.

For the experiment with HO3D images, we find that the performance of our method is equally good across all objects and conditions (Tables 11 and 12). This is likely due to the high-quality reference grasps that are produced by [5]. While our approach can correct interpenetration and noisy poses to some degree, it is conditioned on the reference pose, and hence performs best when provided

with grasp targets that roughly approximate a real human (i.e., physically plausible) grasp on the object. We conclude that especially for generalization to unseen objects, good grasp references are important.

Generalization to Unseen Object Details In Table 13, we report the detailed results of the generalization experiment. We observe a large variance across the different object sets. For example, the success rate of our method on test set 1, which comprises easier geometries, reaches up to 0.83. On the other hand, our method only achieves a 0.33 success rate on the test set 6, which contains the complex objects "037 scissors" and "040 large marker". Generally, we find that our method is able to outperform the static baselines across the different test sets. As a future extension, it would be interesting to scale the method to even larger datasets. Such ambitions are supported by different works for dexterous robotic manipulation tasks [3, 6], which have recently demonstrated the ability of large scale training with regards to object types in order to achieve generalization across objects.

D. Societal Impact

While the dynamic grasps generated by our method are not yet indistinguishable from real ones, we can extrapolate to a more mature version of this work, opening-up many potential applications, e.g., in AR/VR, HCI or robotics. These applications may lead to negative societal impact, where so-called deep-fakes are the obvious nefarious use of such methods. However, it is also possible that due to the computational complexity and resulting real-world cost of implementing even positive applications, there may be negative implications for already underprivileged populations. For example, a service robot that may learn to cooperate with humans may not be affordable for many that have need for such advanced care technologies.

In going forward with the development of technologies related to this paper, one must carefully balance the potential positive uses and the undesired side-effects. Since we have no control over whether such technologies will be developed at all, by whom and for which purposes, we argue that openly discussing the technical details, properties and limitations is one way to ensure that a) such technologies are well understood and therefore counter measures to nefarious use would be easier to implement and b) that as many individuals as possible can have access to related technologies. To this end we will release all source code for research purposes.

E. Glossary

We include a glossary in Tables 14 and 15 to provide an overview of the many notations used in this paper.

[1] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*. 5, 12, 13

[2] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 5, 7, 12, 13

[3] Tao Chen, Jie Xu, and Pulkit Agrawal. A simple method for complex in-hand manipulation. In *5th Annual Conference on Robot Learning (CoRL)*, 2021. 2, 14

[4] Sammy Christen, Lukas Jendele, Emre Aksan, and Otmar Hilliges. Learning functionally decomposed hierarchies for continuous control tasks with path planning. *IEEE Robotics and Automation Letters*, 6(2):3623–3630, 2021. 9

[5] 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 (CVPR)*, pages 1471–1481, 2021. 1, 2, 5, 6, 7, 11, 12, 13, 16

[6] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021. 2, 14

[7] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018. 5, 10, 11

[8] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the International Conference on Computer Vision*, 2021. 1, 2, 5, 6, 7, 11, 12, 13, 15, 16

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations (ICLR)*, 2015. 11

[10] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 3, 6, 11

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5, 11

[12] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 211–228. Springer, 2020. 10

[13] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7159–7169, June 2021. 2, 10

	GT+PD			GT+IK			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
002 master chef can	18.0 ± 9.8	0.19	5.68	17.2 ± 10.0	0.20	12.96	1.5 ± 5.7	0.90	1.54
003 cracker box	18.2 ± 10.1	0.16	3.62	9.2 ± 10.7	0.47	9.16	8.4 ± 12.3	0.68	2.48
004 sugar box	15.6 ± 11.2	0.32	5.52	8.7 ± 9.7	0.50	11.64	0.1 ± 0.0	1.00	3.28
005 tomato soup can	12.1 ± 10.8	0.28	4.34	12.2 ± 10.1	0.33	10.72	1.4 ± 5.5	0.90	2.51
006 mustard bottle	4.4 ± 7.8	0.64	9.51	1.1 ± 1.5	0.76	16.50	3.2 ± 8.5	0.88	2.20
007 tuna fish can	17.4 ± 9.4	0.14	2.52	16.6 ± 9.6	0.19	5.26	1.8 ± 4.6	0.71	1.13
008 pudding box	15.1 ± 9.7	0.21	3.91	13.2 ± 11.2	0.39	7.19	2.4 ± 5.9	0.78	0.98
009 gelatin box	18.9 ± 9.4	0.15	2.23	18.3 ± 9.3	0.20	4.61	3.7 ± 8.6	0.85	0.98
010 potted meat can	13.6 ± 10.3	0.30	4.14	11.4 ± 10.0	0.39	8.81	1.8 ± 6.0	0.89	0.64
011 banana	16.7 ± 8.6	0.09	3.27	15.9 ± 10.0	0.20	4.97	5.9 ± 9.6	0.47	0.73
019 pitcher base	11.0 ± 11.1	0.40	7.47	11.2 ± 11.2	0.37	17.01	6.9 ± 10.8	0.68	3.16
021 bleach cleanser	5.1 ± 8.5	0.61	8.25	3.8 ± 7.0	0.61	15.64	0.2 ± 0.4	0.94	3.08
024 bowl	10.2 ± 10.7	0.41	3.15	7.3 ± 9.3	0.62	9.83	7.5 ± 10.7	0.62	2.08
025 mug	12.3 ± 10.3	0.35	3.24	9.7 ± 10.6	0.53	6.56	10.2 ± 12.2	0.59	1.61
035 power drill	0.8 ± 1.8	0.83	10.68	5.3 ± 9.0	0.65	15.76	6.4 ± 10.7	0.59	1.64
036 wood block	13.4 ± 11.6	0.42	7.31	11.3 ± 11.5	0.50	13.36	0.1 ± 0.0	1.00	3.56
037 scissors	10.6 ± 9.9	0.35	1.72	13.5 ± 9.4	0.22	3.03	19.3 ± 10.0	0.11	0.35
040 large marker	19.8 ± 5.8	0.04	1.38	21.5 ± 5.9	0.05	2.81	20.9 ± 7.3	0.05	0.09
052 extra large clamp	15.0 ± 9.9	0.28	1.97	12.3 ± 11.0	0.44	3.50	10.4 ± 11.7	0.44	1.70
061 foam brick	18.9 ± 7.3	0.12	1.93	17.2 ± 10.5	0.26	5.23	3.5 ± 8.1	0.84	1.30
Average	13.4 ± 9.2	0.31	4.59	11.8 ± 9.4	0.39	9.23	5.8 ± 7.4	0.70	1.75

Table 7. Detailed results for the DexYCB train set.

	GT+PD			GT+IK			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
002 master chef can	20.4 ± 9.3	0.17	4.17	19.2 ± 8.6	0.17	10.73	0.6 ± 1.2	0.83	1.67
003 cracker box	14.7 ± 11.1	0.33	5.21	9.0 ± 11.5	0.67	13.10	8.9 ± 12.4	0.67	4.06
004 sugar box	14.1 ± 12.1	0.43	6.75	3.9 ± 8.5	0.86	16.04	3.9 ± 9.3	0.86	3.02
005 tomato soup can	7.6 ± 9.0	0.50	3.25	8.2 ± 8.4	0.25	7.56	14.4 ± 11.4	0.25	1.03
006 mustard bottle	9.5 ± 9.9	0.50	6.77	6.7 ± 9.9	0.63	13.19	7.0 ± 12.0	0.75	2.13
007 tuna fish can	16.4 ± 9.4	0.14	1.55	16.3 ± 8.5	0.14	3.55	0.1 ± 0.0	1.00	1.50
008 pudding box	18.5 ± 8.2	0.17	2.08	12.7 ± 10.5	0.33	4.19	4.2 ± 9.2	0.83	0.92
009 gelatin box	21.6 ± 9.7	0.14	2.00	15.3 ± 13.3	0.29	4.59	3.8 ± 8.5	0.71	0.86
010 potted meat can	9.9 ± 10.2	0.40	2.60	10.8 ± 10.8	0.40	7.00	0.1 ± 0.1	1.00	1.00
011 banana	13.8 ± 9.9	0.14	2.79	13.0 ± 10.8	0.29	4.36	10.8 ± 12.0	0.43	0.50
019 pitcher base	12.6 ± 12.5	0.50	7.83	13.1 ± 12.7	0.50	25.54	6.2 ± 9.1	0.50	3.08
021 bleach cleanser	5.8 ± 8.3	0.40	10.43	5.1 ± 9.3	0.80	15.40	10.2 ± 12.3	0.60	2.10
024 bowl	9.3 ± 11.1	0.50	5.83	0.3 ± 0.3	1.00	13.44	12.0 ± 11.9	0.50	1.56
025 mug	7.4 ± 9.2	0.50	3.00	0.5 ± 0.5	1.00	9.13	4.1 ± 9.0	0.83	1.33
035 power drill	0.2 ± 0.1	1.00	8.25	0.5 ± 0.4	0.83	15.02	5.6 ± 10.4	0.67	2.71
036 wood block	20.6 ± 9.1	0.17	4.88	12.9 ± 12.4	0.50	11.73	0.2 ± 0.1	1.00	4.17
037 scissors	8.1 ± 9.9	0.50	4.44	6.4 ± 8.6	0.50	6.55	19.0 ± 8.1	0.13	0.58
040 large marker	18.7 ± 3.2	0.00	1.43	13.5 ± 8.7	0.20	3.58	24.2 ± 0.1	0.00	0.00
052 extra large clamp	16.2 ± 8.9	0.14	2.48	5.1 ± 7.7	0.43	5.21	14.0 ± 12.1	0.43	1.88
061 foam brick	16.0 ± 10.8	0.29	2.43	10.0 ± 8.3	0.13	4.86	10.5 ± 12.0	0.57	1.30
Average	13.1 ± 9.1	0.35	4.41	9.1 ± 8.5	0.50	9.74	8.0 ± 8.1	0.63	1.77

Table 8. Detailed results for the DexYCB test set.

	Jiang *et. al* [8]+PD			Jiang *et. al* [8]+IK			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
002 master chef can	24.0 ± 1.5	0.00	6.39	24.0 ± 1.6	0.00	16.19	2.6 ± 7.5	0.90	2.70
003 cracker box*	22.8 ± 2.4	0.00	6.97	22.8 ± 2.5	0.00	8.18	6.0 ± 10.3	0.70	5.80
004 sugar box*	14.7 ± 8.5	0.10	5.60	15.2 ± 8.7	0.10	10.45	1.4 ± 5.3	0.95	4.03
005 tomato soup can	14.8 ± 9.1	0.10	5.62	14.6 ± 9.0	0.10	11.51	3.8 ± 8.7	0.85	5.76
006 mustard bottle*	5.6 ± 7.2	0.30	5.66	5.2 ± 7.3	0.50	10.61	0.8 ± 2.8	0.95	5.14
007 tuna fish can	20.9 ± 1.7	0.00	2.88	21.4 ± 1.7	0.00	4.84	0.3 ± 0.5	0.90	2.03
008 pudding box	11.0 ± 10.3	0.10	4.92	10.9 ± 10.2	0.10	7.88	6.2 ± 10.1	0.70	0.91
009 gelatin box	14.1 ± 8.7	0.10	3.89	14.3 ± 9.2	0.10	8.38	1.7 ± 5.9	0.90	1.44
010 potted meat can*	20.4 ± 2.9	0.00	3.96	20.7 ± 2.8	0.00	6.89	7.1 ± 11.0	0.65	0.46
011 banana*	4.4 ± 6.1	0.30	3.52	6.1 ± 8.0	0.40	2.33	7.0 ± 9.4	0.45	3.18
019 pitcher base*	6.4 ± 8.4	0.50	8.11	7.1 ± 8.0	0.30	13.64	6.4 ± 10.1	0.65	1.56
021 bleach cleanser*	0.8 ± 1.9	0.90	5.76	0.5 ± 0.6	0.80	6.82	2.1 ± 6.4	0.90	4.89
024 bowl	5.5 ± 8.0	0.70	4.93	5.7 ± 8.3	0.70	3.24	5.2 ± 7.7	0.40	1.78
025 mug*	7.9 ± 9.4	0.50	4.32	8.2 ± 9.5	0.40	2.19	1.3 ± 5.2	0.95	4.78
035 power drill*	12.3 ± 12.0	0.20	5.84	12.1 ± 12.1	0.20	8.40	11.4 ± 11.5	0.20	1.27
036 wood block	22.0 ± 4.6	0.00	7.28	22.2 ± 4.2	0.00	5.06	2.6 ± 7.5	0.90	2.99
037 scissors*	5.0 ± 7.5	0.30	2.37	8.2 ± 8.5	0.30	1.50	0.7 ± 1.4	0.85	2.18
040 large marker	10.3 ± 9.6	0.40	1.86	10.1 ± 9.7	0.40	3.41	7.2 ± 9.6	0.55	0.92
052 extra large clamp	3.6 ± 5.9	0.40	4.97	3.6 ± 6.0	0.40	8.21	3.4 ± 7.2	0.65	3.37
061 foam brick	20.9 ± 1.8	0.00	3.46	21.3 ± 2.0	0.00	6.56	1.3 ± 5.1	0.95	1.66
Average	12.4 ± 6.4	0.25	4.92	12.7 ± 6.5	0.24	7.31	3.9 ± 7.2	0.75	2.84

Table 9. Detailed results for the DexYCB and HO3D train set with grasp references from a static grasp synthesis method [8]. HO3D objects are marked by *.

	Jiang *et. al* [8]+PD			Jiang *et. al* [8]+IK			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
002 master chef can	24.0 ± 1.6	0.00	7.54	23.3 ± 1.3	0.00	7.03	5.3 ± 10.5	0.80	1.88
003 cracker box*	22.8 ± 2.5	0.00	6.73	20.3 ± 5.4	0.00	7.74	5.4 ± 10.6	0.80	7.06
004 sugar box*	15.2 ± 8.7	0.10	5.18	15.0 ± 8.4	0.00	13.85	0.2 ± 0.2	1.00	3.48
005 tomato soup can	14.6 ± 9.0	0.10	5.29	15.0 ± 9.0	0.00	10.75	0.1 ± 0.1	1.00	5.86
006 mustard bottle*	5.2 ± 7.3	0.50	5.19	7.6 ± 7.6	0.10	14.29	0.4 ± 0.9	0.90	6.23
007 tuna fish can	21.4 ± 1.7	0.00	2.61	19.8 ± 6.7	0.10	5.25	2.5 ± 7.0	0.90	1.51
008 pudding box	10.9 ± 10.2	0.10	5.51	7.0 ± 9.1	0.30	9.38	7.3 ± 10.7	0.60	0.46
009 gelatin box	14.3 ± 9.2	0.10	3.49	13.0 ± 10.6	0.40	6.91	0.8 ± 1.6	0.80	1.83
010 potted meat can*	20.7 ± 2.8	0.00	4.74	21.0 ± 2.6	0.00	8.70	7.6 ± 10.4	0.40	0.60
011 banana*	6.1 ± 8.0	0.40	3.38	8.1 ± 8.9	0.50	3.16	4.8 ± 9.1	0.80	2.09
019 pitcher base*	7.1 ± 8.0	0.30	8.50	6.3 ± 8.1	0.40	0.00	12.3 ± 11.9	0.40	0.98
021 bleach cleanser*	0.5 ± 0.6	0.80	6.50	9.9 ± 9.1	0.30	7.75	0.2 ± 0.2	1.00	5.69
024 bowl	5.7 ± 8.3	0.70	4.51	2.5 ± 5.1	0.80	2.83	3.4 ± 7.0	0.80	2.18
025 mug*	8.2 ± 9.5	0.40	5.94	3.7 ± 6.3	0.60	2.41	0.1 ± 0.0	1.00	4.64
035 power drill*	12.1 ± 12.1	0.20	4.91	14.7 ± 12.0	0.30	5.91	10.2 ± 11.4	0.20	1.49
036 wood block	22.2 ± 4.2	0.00	6.65	24.0 ± 2.0	0.00	1.94	10.7 ± 12.4	0.50	1.84
037 scissors*	8.2 ± 8.5	0.30	2.94	5.1 ± 7.1	0.50	1.26	7.8 ± 11.3	0.60	1.74
040 large marker	10.1 ± 9.7	0.40	1.65	6.8 ± 9.9	0.60	4.01	7.3 ± 10.4	0.50	1.94
052 extra large clamp	3.6 ± 6.0	0.40	4.09	3.1 ± 6.1	0.60	7.46	5.1 ± 8.4	0.60	3.22
061 foam brick	21.3 ± 2.0	0.00	3.51	20.9 ± 1.3	0.00	7.81	0.1 ± 0.0	1.00	1.46
Average	12.7 ± 6.5	0.24	4.94	12.4 ± 6.8	0.28	6.42	4.6 ± 6.7	0.73	2.81

Table 10. Detailed results for the DexYCB and HO3D test set with grasp references from a static grasp synthesis method [8]. HO3D objects are marked by *.

	Grady *et. al* [5]+PD			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
003 cracker box	2.5 ± 6.7	0.85	14.33	0.3 ± 0.1	1.00	3.18
004 sugar box	16.3 ± 9.3	0.05	17.04	2.9 ± 6.6	0.70	2.40
006 mustard bottle	9.1 ± 9.7	0.40	26.46	0.3 ± 0.4	0.95	2.89
010 potted meat can	3.9 ± 8.5	0.70	15.42	2.2 ± 5.9	0.90	0.78
011 banana	10.0 ± 9.9	0.35	13.80	1.7 ± 4.9	0.80	1.98
021 bleach cleanser	0.9 ± 3.3	0.95	18.84	0.3 ± 0.1	1.00	2.86
025 mug	2.7 ± 6.7	0.80	5.49	2.0 ± 5.4	0.85	4.74
035 power drill	0.2 ± 0.4	0.95	16.09	0.3 ± 0.2	1.00	2.56
037 scissors	0.1 ± 0.1	1.00	6.96	3.0 ± 7.3	0.75	2.60
Average	5.1 ± 6.1	0.67	14.94	1.4 ± 3.4	0.88	2.67

Table 11. Detailed results for the train set with ContactOpt [5] on HO3D images.

	Grady *et. al* [5]+PD			Ours		
Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
003 cracker box	6.7 ± 9.7	0.60	13.41	0.26 ± 0.1	1.00	2.14
004 sugar box	23.6 ± 1.5	0.00	17.71	0.95 ± 2.2	0.90	2.53
006 mustard bottle	4.6 ± 7.6	0.60	25.16	0.64 ± 0.8	0.80	2.46
010 potted meat can	1.9 ± 5.5	0.90	14.08	0.63 ± 1.3	0.90	0.38
011 banana	12.0 ± 10.4	0.20	13.38	0.61 ± 0.5	0.80	1.91
021 bleach cleanser	0.9 ± 2.4	0.90	18.23	2.86 ± 7.6	0.90	2.25
025 mug	6.4 ± 8.4	0.50	4.80	5.05 ± 9.7	0.80	4.16
035 power drill	0.1 ± 0.1	1.00	14.84	0.71 ± 0.6	0.60	1.68
037 scissors	2.7 ± 6.9	0.70	4.34	5.30 ± 9.3	0.60	1.21
Average	6.5 ± 5.8	0.60	13.99	1.9 ± 3.57	0.81	2.08

Table 12. Detailed results for the test set with ContactOpt [5] on HO3D images.

		GT+PD			GT+IK			Ours		
	Object	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]	SimDist [mm/s] ↓	Success ↑	Interp. [cm^3]
Test set 1	004 sugar box	15.2 ± 11.4	0.35	5.86	7.4 ± 9.4	0.60	12.87	2.3 ± 7.2	0.92	3.35
	005 tomato soup can	11.4 ± 10.5	0.32	4.17	11.5 ± 9.8	0.32	10.21	5.7 ± 10.3	0.76	1.56
	006 mustard bottle	6.0 ± 8.5	0.60	8.64	2.9 ± 4.2	0.72	15.44	2.1 ± 5.7	0.80	1.87
	Average	10.9 ± 10.1	0.42	6.22	7.3 ± 7.8	0.55	12.84	3.3 ± 7.7	0.83	2.26
Test set 2	061 foam brick	18.1 ± 8.3	0.16	2.06	15.3 ± 9.9	0.23	5.13	9.6 ± 12.0	0.62	0.55
	010 potted meat can	12.8 ± 10.2	0.33	3.80	11.3 ± 10.2	0.39	8.42	5.1 ± 9.3	0.61	1.36
	052 extra large clamp	15.4 ± 9.6	0.24	2.11	10.3 ± 10.1	0.44	3.98	14.8 ± 11.5	0.16	1.53
	Average	15.4 ± 9.4	0.24	2.66	12.3 ± 10.1	0.35	5.84	9.8 ± 10.9	0.46	1.15
Test set 3	003 cracker box	17.4 ± 10.3	0.20	4.00	9.1 ± 10.9	0.52	10.11	12.9 ± 13.3	0.52	2.86
	007 tuna fish can	17.1 ± 9.4	0.14	2.28	16.5 ± 9.3	0.18	4.83	5.4 ± 10.2	0.79	0.88
	011 banana	15.8 ± 9.0	0.11	3.11	14.9 ± 10.2	0.23	4.78	7.2 ± 10.3	0.50	1.24
	Average	16.8 ± 9.6	0.15	3.13	13.5 ± 10.2	0.31	6.57	8.5 ± 11.3	0.60	1.66
Test set 4	002 master chef can	18.5 ± 9.7	0.19	5.33	17.6 ± 9.7	0.19	12.45	6.6 ± 11.1	0.65	1.21
	036 wood block	15.1 ± 11.1	0.36	6.75	11.7 ± 11.7	0.50	12.98	3.4 ± 8.8	0.85	3.31
	052 extra large clamp	15.4 ± 9.6	0.24	2.11	10.3 ± 10.1	0.44	3.98	15.1 ± 11.3	0.28	2.26
	Average	16.3 ± 10.1	0.26	4.73	13.2 ± 10.5	0.38	9.80	8.4 ± 10.4	0.59	2.26
Test set 5	008 pudding box	16.0 ± 9.4	0.20	3.45	13.0 ± 11.0	0.38	6.44	3.5 ± 8.4	0.83	0.90
	019 pitcher base	11.4 ± 11.4	0.42	7.56	11.6 ± 11.5	0.40	19.06	11.3 ± 11.5	0.40	3.52
	035 power drill	0.6 ± 1.3	0.87	10.04	4.0 ± 6.8	0.70	15.57	11.0 ± 12.8	0.43	1.63
	Average	9.3 ± 7.4	0.50	7.02	9.6 ± 9.8	0.49	13.69	8.6 ± 10.9	0.56	2.02
Test set 6	005 tomato soup can	11.4 ± 10.5	0.32	4.17	11.5 ± 9.8	0.32	10.21	9.1 ± 12.0	0.60	2.10
	037 scissors	9.9 ± 9.9	0.39	2.55	11.3 ± 9.2	0.31	4.11	18.3 ± 10.4	0.19	0.71
	040 large marker	19.6 ± 5.3	0.03	1.38	20.0 ± 6.4	0.07	2.95	18.2 ± 10.2	0.19	0.51
	Average	13.6 ± 8.6	0.25	2.70	14.3 ± 8.4	0.23	5.76	15.2 ± 10.9	0.33	1.11

Table 13. **Generalization to Unseen Objects**. We evaluate generalization to unseen objects and compare our model with the baselines. We create six different test sets of three objects, which we leave out during training. We report the detailed results per test set in this table.

Notation	Meaning
$\mathbf{s}$	state
$\mathbf{a}$	action
$\boldsymbol{\pi}_g$	grasping policy
$\boldsymbol{\pi}_m$	motion synthesis policy
$\mathbf{D}$	static grasp label
$\mathbf{x}$	3D joint position
$\mathbf{q}$	joint angles
$\mathbf{T}$	6D pose
$\mathbf{T}_h$	6D global hand pose
$\dot{\mathbf{T}}_h$	6D global hand velocities
$\mathbf{T}_o$	6D object pose
$\dot{\mathbf{T}}_o$	6D object velocities
$\mathbf{T}_g$	6D goal object pose
$\mathbf{q}_h$	hand joint angles
$\dot{\mathbf{q}}_h$	hand joint angular velocities
$\overline{\mathbf{T}}_h$	6D global hand pose in grasp label
$\overline{\mathbf{T}}_o$	6D global object pose in grasp label
$\overline{\mathbf{q}}_h$	3D hand pose in grasp label
$\overline{\mathbf{g}}_c$	target contacts
$\overline{\mathbf{x}}$	3D target joint position
$\mathbf{f}$	contact forces
$\boldsymbol{\tau}$	joint torques
k_p	PD-controller parameter
k_d	PD-controller parameter
$\mathbf{q}_{\text{ref}}$	reference joint angles
$\mathbf{q}_b$	bias joint angle term
$\phi(\cdot)$	feature extractor
$\widetilde{\cdot}$	transformation to wrist reference frame
$\widetilde{\mathbf{T}}_o$	6D object pose in wrist reference frame
$\dot{\widetilde{\mathbf{T}}}_o$	6D object velocities in wrist reference frame
$\dot{\widetilde{\mathbf{T}}}_h$	6D global hand velocities in wrist reference frame
$\widetilde{\mathbf{x}}_z$	vertical distance to surface where object rests

Table 14. Glossary (part 1) for the notation used in this paper.

Notation	Meaning
$\mathbf{G}$	goals
$\widetilde{\mathbf{g}}_x$	3D distance between current and target joint positions
$\widetilde{\mathbf{g}}_q$	angular distance between current and target joint/wrist rotations
$\mathbf{g}_c$	contact vector
$\mathbf{g}_{o,x}$	3D distance between current and target object position
$\mathbf{g}_{o,q}$	angular distance between current and target object rotation
r	total reward for grasping
r_x	position reward
w_x	position reward weight
r_q	pose reward
w_q	pose reward weight
r_c	contact reward
w_c	contact reward weight
λ	contact reward coefficient
m_{o}	object's mass
r_{reg}	regularizing reward term
$w_{\text{reg},h}$	regularizing reward term hand weight
$w_{\text{reg},o}$	regularizing reward term object weight
r_m	total reward motion synthesis
$r_{m,x}$	position reward motion synthesis
$r_{m,q}$	pose reward motion synthesis
α_x	position reward weight motion synthesis
α_q	pose reward weight motion synthesis
$\psi(\cdot)$	feature extractor motion synthesis
$\widehat{\mathbf{T}}_h$	estimated 6D target hand pose
$\mathbf{T}_{\text{pd}}$	6D pose input to the PD-controller for motion synthesis
$\overline{\mathbf{v}}_h$	hand mesh
$\overline{\mathbf{v}}_o$	object mesh

Table 15. Glossary (part 2) for the notation used in this paper.