Learning Robust Real-World Dexterous Grasping Policies via Implicit Shape Augmentation

Anonymous Author(s)
Affiliation
Address
email

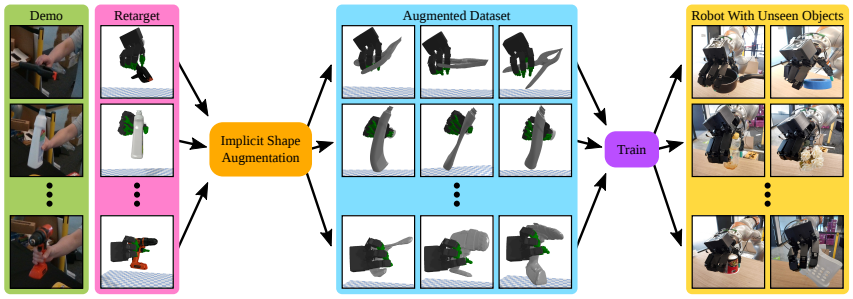


Figure 1: An illustration of the training scheme in ISAGrasp. A few human demonstrations are provided from motion capture. These demonstrations are retargeted to a four-fingered allegro robot in simulation. We use a correspondence-aware generative model to extrapolate the retargeted demonstrations to a large dataset of novel objects. We use this dataset to train a single grasping policy that is able to generalize to a variety of unseen objects in simulation and real world.

Abstract: Dexterous robotic hands have the capability to interact with a wide variety of household objects to perform tasks like grasping. However, learning robust real world grasping policies for arbitrary objects has proven challenging due to the difficulty of generating high quality training data. In this work, we propose a learning system (*ISAGrasp*) for leveraging a small number of human demonstrations to bootstrap the generation of a much larger dataset containing successful grasps on a variety of novel objects. Our key insight is to use a correspondence-aware implicit generative model to deform object meshes and demonstrated human grasps in order to generate a diverse dataset of novel objects and successful grasps for supervised learning, while maintaining semantic realism. We use this dataset to train a robust grasping policy in simulation which can be deployed in the real world. We demonstrate grasping performance with a four-fingered Allegro hand in both simulation and the real world, and show this method can handle entirely new semantic classes and achieve a 79% success rate on grasping unseen objects in the real world.

Keywords: Dexterous Manipulation, Learning from Demonstration, Data Augmentation, Grasping

1 Introduction

Human hands are powerful tools for manipulating a wide range of objects. Our goal is to build dexterous robotic manipulators that can robustly and adeptly interact with human-centric environments. However, robustly controlling a dexterous hand remains challenging due to its high dimensional state and action space and its multi-modal contact dynamics.

Recent work has used deep reinforcement learning algorithms to learn complex tasks with dexterous manipulators [1, 2, 3, 4, 5, 2, 6]. These methods often require careful reward engineering and environment design, and often lack generalization, struggling with robust real world deployment. An alternative paradigm involves collecting a large labelled dataset and using supervised learning

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

(imitation learning). This has shown significant success with parallel jaw grippers [7, 8, 9] and suction cups [10, 9]. However, imitation learning methods have proven difficult to scale to multi-fingered robots due to the burden of human data collection. To scale robust policy learning to dexterous grasping, we require techniques that can leverage a small amount of human effort to learn robust, general grasping policies. This can be done by extrapolating a small number of human demonstrations to generate an abundance of data that interacts with *diverse* objects and is *successful* at grasping objects under real world dynamics. The key question is —how do we generate this type of diverse training data?

In this work, we propose *Implicit Shape Augmentated Grasping* (ISAGrasp), shown in Figure 1, a learning system that leverages a correspondence-aware generative model [11] to extrapolate a small number of human demonstrations to a large dataset of realistic objects and their corresponding grasps. In particular, we directly perform deformations on implicit 3-D shape representations of various objects in a learned latent space. The representation of point-wise shape deformations allows us to generate transformed grasps for novel objects that can be made successful with a small amount of active interaction in simulation. In doing so, implicit shape augmentation allows us to grow from a small dataset of human demonstration data for dexterous grasping to a significantly larger and more diverse dataset with novel object shapes and dynamically successful grasps. Given the dataset constructed by implicit shape augmentation, we can now perform large scale supervised learning. In this work, we represent a policy as predicting a pre-grasp and a final pose from pointclouds of the target object, with intermediate motion being performed with standard motion planning libraries. We show empirically that a policy learned via supervised learning on the dataset constructed by ISAGrasp is able to generalize widely across different objects in simulation and the real world. We demonstrate the efficacy of this pipeline on the rescaled YCB instances, ShapeNet and GoogleScans objects in simulation and achieve a 79% success rate on 22 unseen objects in the real world.

In summary, our contributions are (1) we propose a novel system, *ISAGrasp*, that is able to generate both a wide variety of objects *and* the corresponding dexterous grasps from a few human demonstrations. (2) We show that the augmented dataset can be used for learning point-cloud based control policies that are able to robustly grasp a large variety of *novel* objects in simulation (3) We demonstrate the efficacy of the proposed pipeline by deploying it in the real world. (4) We analyze the proposed pipeline through several ablation studies in simulation.

2 ISAGrasp: Dexterous Grasping Policies via Implicit Shape Augmentation

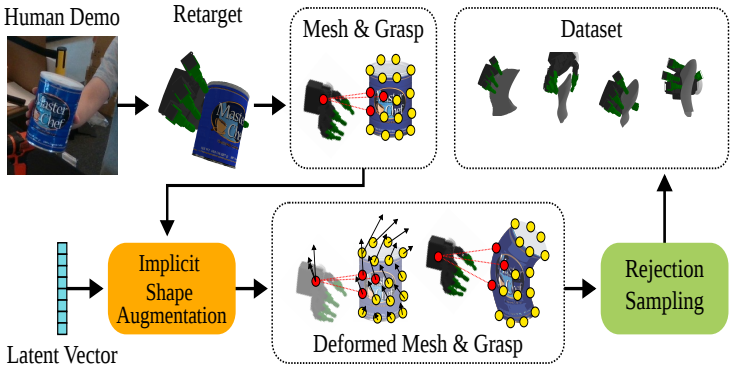


Figure 2: Implicit Shape Augmentation for generating augmented dataset from demonstrations. First a human demonstration is retargeted onto the Allegro hand to generate meshes and grasp labels. This data can then be used to generate a variety of new objects via shape augmentation with DIF-Net [11]. Grasps for these deformed objects can then be further refined with rejection sampling to generate dynamically consistent grasps.

To learn robust grasping policies that can operate in the real world for grasping novel objects of various shapes, we propose ISAGrasp —a framework for supervised learning of robust and generalizable grasping policies from successful grasps on a large variety of objects generated from a small set of human provided expert demonstrations.

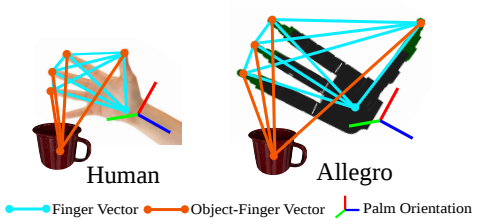


Figure 3: Illustration of retargeting a human hand demonstration to an Allegro hand

$$d_g = \sum_{i=0}^{N} \left\| \vec{\mathbf{a}_{r_i}} - s_r \vec{\mathbf{a}_{h_i}} \right\|^2, \quad (1)$$

$$d_c = \sum_{i=0}^{N} \left\| \vec{\mathbf{c}_{r_i}} - \vec{\mathbf{c}_{h_i}} \right\|^2, \quad (2)$$

$$d_r = G(\mathbf{M}_r, \mathbf{M}_h), \quad (3)$$

$$\underset{(\mathbf{q}_r, \mathbf{f}_r, \mathbf{M}_r)}{\arg\min} \; (w_g d_g + w_c d_c + w_r d_r), \quad (4)$$

Figure 4: Optimization setup describing the retargeting problem from human to Allegro hand

Given a pointcloud of an object, we model a grasping policy as predicting (1) a pre-grasp pose $\mathbf{T}$ that controls robot hand's translation $\mathbf{T_t}$ and rotation $\mathbf{T_q}$, and (2) a final pose $\mathbf{G_f}$ that controls the 16-DoF finger pose that can firmly grab the object.

ISAGrasp assumes access to a set of labelled human grasping demonstrations, can be used to learn grasping policies via supervised learning. However, to learn truly robust and general grasping policies, this dataset must be grown to a much more diverse set of objects and successful grasps. Generating a multi-fingered grasping dataset of diverse yet realistic objects is a non-trivial problem [12, 13, 14]. We approach this problem by leveraging a correspondence-aware, deformation-based generative model to widely augment the set of human provided demonstrations, which can then be used to learn robust and general policies via supervised learning. An overview of our ISAGrasp system is shown in Figure 2, and we detail each component below.

2.1 Human-Robot Retargeting

ISAGrasp first collects N human hand-arm demonstrations $\mathcal{D}=\{\tau_0^h, \tau_1^h, \ldots, \tau_N^h\}$ using motion capture, where each demonstration $\tau_i = \{(h_0^i, o_0^i), (h_1^i, o_1^i), \ldots, (h_T^i, o_T^i)\}$ is a trajectory of hand pose h_t and object pose o_t. These demonstrations are then "retargeted" to a 22-DoF floating Allegro hand [15] in simulation. Note that this is a nontrivial problem since these demonstrations are in the morphology of the human hand and are typically not functional when directly mapped to a robot hand using standard Inverse Kinematics [16].

We formulate the retargeting objective as a non-linear optimization problem. First, to allow a robot to grasp in a similar pose to a human demonstration, we use the same cost function proposed in DexPilot [17], shown in Eq. 1, where s_r is the ratio between the sizes of the robot and human hands, and each $\vec{\mathbf{a}_{r_i}}$ and $\vec{\mathbf{a}_{h_i}}$ is a displacement vector between finger tips and the palm of the robot and human hands, shown as blue in Figure 3. While Eq. 1 encourages grasp shape similarity, we additionally minimize the differences between the relative poses to the object, as captured by the vectors between fingers and the object center for the human grasp, $\vec{\mathbf{c}_{r_i}}$, and the retargeted grasp, $\vec{\mathbf{c}_{h_i}}$ (see Eq.2 and orange lines in Fig. 3). Finally, to orient the robot palm similar to the human hand, we add Eq. 3, which optimizes the minimum geodesic distance G in SO(3) between the rotation matrix of the human palm $\mathbf{M}_h$ and the robot palm $\mathbf{M}_r$. The final retargeting goal is to find the 22 DoF configuration that minimizes the weighted sum of these three terms, which results in a candidate grasp $\mathbf{G_r}$ for each object in a human provided dataset.

As we describe in Section 2.3, the retargeted grasps are then refined via a rejection sampling step and used to generate the dataset with successful, dynamically consistent retargeted grasps. However, since this dataset is typically quite limited, we next outline how to augment the dataset with novel objects and grasps via implicit shape augmentation.

2.2 Implicit Shape Augmentation

Given a small set of object point clouds and successful retargeted grasps, $\mathbf{G_r}$, we build a large scale augmented dataset of novel objects and their corresponding successful grasps $\mathbf{G_d}$, using a correspondence-aware implicit generative model, DIF-Net [11].

To recap at a high level, DIF-Net aims to learn an implicit representation of 3D shapes as a scalar field. More specifically, DIF-Net represents a 3D shape via a instance agnostic template implicit field (which is common for all shapes of a particular category and represent the common features of a category), together with a latent conditional 3D deformation field (that perturbs this template field) which allows the shapes to be adapted to every particular object instance while maintaining semantic 3-D structure. Different novel but realistic shapes can be generated by sampling different latent vectors α and generating object deformations on known object classes using the learned deformation and correction fields, while maintaining semantics. This model naturally provides dense correspondences across object instances since different object instances are pointwise deformations on the same template.

We use this generative model in two ways: (1) leverage the ability to sample a variety of object instances by sampling latent vectors α to generate various novel objects via deformations and (2) the resulting dense correspondences allow us to estimate dynamically consistent grasps from human demonstrations to novel generated objects. Specifically, we use the latent conditional deformation field from a pretrained DIF-Net to generate novel instances and dynamically consistent grasps. To generate novel objects, we sample latent vectors α from a Gaussian that conditions a latent conditional deformation field. This deformation field generates point-wise deformations of particular object meshes chosen from the set of human demonstrations to generate novel objects. Second, to estimate the new grasps $\mathbf{G_d}$ for the deformed objects, we find N reference points on the original mesh that are close to the root position t^r of the robot hand and compute the position t_o of the robot relative to a local coordinate system O_i, centered at each reference point i: $t_i^o = O_i^{-1} \cdot t^r$. Once the object is deformed, we compute the corresponding coordinate system O_i^d on the deformed meshes using the same deformation field, to estimate the new grasp location t_d using the average of the local offsets: $t^d = \frac{1}{N}\Sigma_{i=1}^{N}(O_i^d \cdot t_i^o)$. In this way, the dense correspondences obtained via the DIF-Net are directly useful in generating grasps for novel, deformed object instances. Figure 5 visualizes how

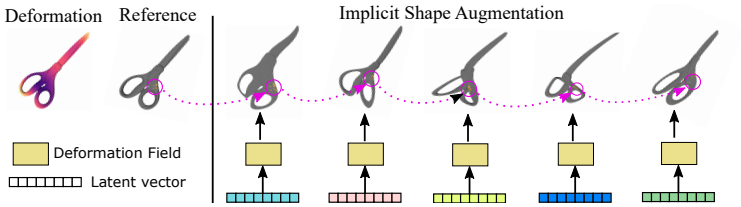


Figure 5: Deformation map and grasping correspondences for objects generated in ISAGrasp. Grasping correspondences on the original object (reference) and the deformed objects are highlighted inside the circle. As can be seen, object semantics are maintained. Different object instances are generated by sampling different latents

objects and correspondences deform with different sampled latent vectors. We highlight the reference points on the original mesh, and their correspondences on deformed meshes (purple circles). The objects are deformed into a variety of realistic shapes while maintaining the original semantic structures and grasping correspondences.

Using the new grasp location t^d, together with retargeted orientation q^r and finger pose f^r, we can construct new grasps $\mathbf{G_d}(t^d, q^r, f^r)$. While $\mathbf{G_d}$ are not guaranteed to be successful grasps, they provide good starting points for the local search procedure described next.

2.3 Grasp Refinement for Dynamics Consistency

Given transformed grasps $\mathbf{G_d}$, we use pose perturbation with rejection sampling to generate successful and dynamically consistent grasping poses. Since our shape augmentation model transforms successful hand grasps via the 3D deformation field, we found that only a small amount of perturbation is typically needed to find successful grasps for the deformed models using rejection sampling. In particular, we sample local perturbations δ_t, δ_r and δ_f from a uniform distribution, and add these perturbations to the translation t^d, rotation q^r and finger joints f^r of the transformed poses $\mathbf{G_d}$. We evaluate success of each perturbed grasp $\mathbf{G_p}$ using a physics simulator (Pybullet [18]) and add domain randomization to save robust successful grasps and objects to the training dataset.

Using above methods, we can generate a large and successful dexterous grasping dataset with a variety of novel objects, and then perform supervised policy learning, as described in the next section.

2.4 Policy Learning via Supervised Learning

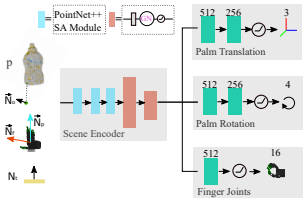


Figure 6: Policy network architecture. The point-net++ architecture inputs an object point cloud p, object surface normal $\vec{N_o}$, the table normal $\vec{N_t}$, robot facing direction $\vec{N_f}$ and the pointing direction $\vec{N_p}$ to generate palm translation, rotation and finger joints for the dexterous grasp.

As described in the beginning of Section 2, we model the grasping problem as predicting a pre-grasp pose $\mathbf{T}$ and a final pose $\mathbf{G_f}$ given an object pointcloud observation. We perform a standard empirical risk minimization procedure on the above-mentioned dataset as an architecture and use a network consisting of PointNet++ SA modules [19] as a feature extractor. Instead of only feeding the raw point cloud to the network, we found significant improvements by appending object points with additional information regarding the alignment between the robot hand and the local object surface. In particular, we use the object point cloud p, the surface normal at each point $\vec{N_o}$, the normal of the table surface $\vec{N_t}$, and the hand facing direction $\vec{N_f}$ and pointing direction $\vec{N_p}$ to define the following feature vectors: $f(p)$=$(p_x, p_y, p_z, (\vec{N_o} \cdot \vec{N_t}), (\vec{N_o} \cdot \vec{N_f}), (\vec{N_o} \cdot \vec{N_p}), (\vec{N_f} \cdot \vec{N_t})$. We append these vectors to each point in the point cloud. We found these features provide a compact description of alignment between the robot hand and the object, and using the relative vector alignments via the pairwise dot products allows us to improve the final grasping performance.

The network outputs a 3-dim translation and 4-dim quaternion, which are used to define the pregrasp pose $\mathbf{T}$. Additionally, the network predicts a 16-dim finger poses, which is used to define final pose $\mathbf{G}_f$. Figure 6 shows our network architecture details. We provide training details in Appendix A.

3 System Details

Robotic System. In simulation, we control a 22-DOF Allegro robot at 12Hz using a PD controller, with the torque force that is close to a real robot: $0.6\ N.m$. In real world experiments, We deploy our policy to a robotic platform that has 23 actuators across a KUKA LBR iiwa 7 R800 robot arm and a Wonik Robotics Allegro robotic hand, and use two cameras to provide necessary point cloud information (Appendix C). To control the robot in simulation, we directly reset the robot at the pregrasp pose $\mathbf{T}$ with open fingers, and linearly interpolate 10 times and move fingers to the grasping finger poses $\mathbf{G_f}$. In real world, we extend the pregrasp from the pointcloud center by 5cm, and interpolate both translation and rotation from the initial robot pose to this pose. These pose targets are passed to an underlying, manually-derived geometric fabrics policy that generates high-frequency joint position, velocity, and acceleration targets across all joints of the arm. The design and tuning of this policy is exactly the same as the one reported in [20]. Finally, the target joint positions generated by fabrics are directly fed to an underlying gravity-compensated joint PD controller at 30 Hz, which are upsampled to 1000 Hz via polynomial interpolation.

Initial Dataset Construction. We extracted human demonstrations from the DexYCB dataset [21], which contains mocap sequences $\mathcal{D}_{\text{human}}$ of humans hand poses q_T^i picking up 20 YCB objects [22] with poses o_T^i . We picked 10 demonstrations per object, resulting in 200 demonstration in total.

ISAGrasp Implementation Details. We use a pretrained DIF-Net [11] to augment objects into a variety of novel shapes. First, we sample a 128-dim latent vector from a Gaussian distribution $\sim \mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 0.002$. Second, to estimate corresponding new grasps $\mathbf{G_d}$, we choose $N = 20$ closest points on the object surface as reference points and compute corresponding grasp location t_d (Section2.2). To obtain successful deformed grasps, we apply rejection sampling by uniformly sampling $\delta_t \in [-0.02\ m, 0.02\ m]$ for translation t_d, and $\delta_r \in [-0.5, 0.5]$ radians for rotation q_r, and apply the same perturbation $\delta_f \in [-0.1, 0.1]$ radians for finger joints (Appendix C).

4 Experiments

Through our experimental evaluation, we aim to answer the following questions:

Table 1: Baselines evaluated in simulation

	RescaledYCB	ShapeNet	GoogleScans
Random	0.03	0.05	0.02
Heuristic	0.27	0.40	0.16
GraspIt!	0.14	0.48	0.24
PPO without demonstration	0.00	0.00	0.00
PPO with Demonstration	0.01	0.01	0.00
DexYCB without augmentation	0.74	0.56	0.51
DexYCB+Implicit Shape Augmentation (ours)	**0.74**	**0.74**	**0.70**

1. Does ISAGrasp generate realistic novel objects based on a small set of scanned objects?
2. Does ISAGrasp allow for easy generation of dynamically consistent grasps on novel objects through grasp transformation and refinement?
3. Do policies learned on the dataset produced by ISAGrasp show improved robustness and generalization on unseen objects?
4. How does the training perform with different input features.

We address these questions through a study in a PyBullet [18] simulation, followed by a real world experimental evaluation using the robotic system described in Section 3. In subsequent sections we describe baseline methods, evaluation protocols and present experimental results followed by an ablation study on object shape generation.Additional analysis on different elements of the ISAGrasp system are in Appendix D, and more videos in the appendix and https://sites.google.com/view/implicitaugmentation/home

Evaluation Metrics. We choose three unseen datasets with an increasing complexity: RescaledYCB, ShapeNet [23], and GoogleScans [24] (see appendix B). In particular, RescaledYCB contains 65 rescaled YCB objects, ShapeNet contains 200 unseen objects from "Can", "Bottle", "Mug" and "Bowl" categories, and GoogleScans objects contains 200 everyday objects. We place objects randomly on the table, and evaluate performance with 5 sets of object mass and friction. The success is defined as when the object is above the table by 10cm.

Baselines. Table 1 additionally compares the performance of our method to other baselines —**(1)** Randomly generate grasps around the object (Random) **(2)** perform a predefined grasp where robot always grasps from the top (Heuristic). **(3)** Graspit!, an optimization-based grasp planner from prior work[25] **(4)** In addition, we train two RL baselines using PPO[26], both without demonstrations and use the demonstration as a reward function. **(5)** Training only on the human demonstrations provided on the initial 20 YCB objects using supervised learning.

Figure 7: Qualitative results on unseen objects in simulation and in the real world. The top 3 rows shows the successful examples of our method on GoogleScans objects and the bottom 2 rows show successful examples using our policy deployed on unseen objects in real world.

4.1 Simulation Results.

We first evaluate the efficacy of the ISAGrasp system on learning dexterous grasping policies in simulation. Table 1 shows the overall success rate of our method ISAGrasp on three unseen datasets

Table 2: Real world test of ISAGrasp on 22 unseen objects. ISAGrasp is able to achieve 79% success rate

obj	Tray	Crab	Chips	Box1	Box2	Bread	Flower	Pot	Roll	Hat	Honey
#	3/5	4/5	4/5	2/5	4/5	5/5	3/5	4/5	4/5	5/5	5/5
obj	Box5	Scarf	Purse	Tape	Box3	Bottle	Cream	Cap	Drill	Bag	Pringles
#	4/5	5/5	3/5	4/5	5/5	4/5	5/5	5/5	2/5	4/5	3/5

in simulation. ISAGrasp (bottom row) is able to achieve 74% success rate on rescaled unseen YCB objects even the policy is only trained on augmented shapes. In addition, our method achieves 74% and 70% success rate on ShapeNet and GoogleScans objects.

We find that the Random and Heuristic baseelines described above do not perform well on RescaledYCB, ShapeNet and GoogleScans objects in simulation. The random grasp baseline achieves less than 5% success rate, while heuristic grasp achieves 40% success rate (only on ShapeNet objects) and performs poorly on the other two datasets. This shows the importance of actually learning the grasping behavior rather than hard coding it or sampling randomly.

The GraspIt baseline achieves the 48% on ShapeNet, and perform poorly on other two datasets, showing the benefit of learning from demonstrations as well as dataset augmentation that is leveraged by ISAGrasp. We also find that the reinforcement learning baselines using PPO perform very poorly, indicating the importance of using supervised learning to avoid the challenge of exploration.

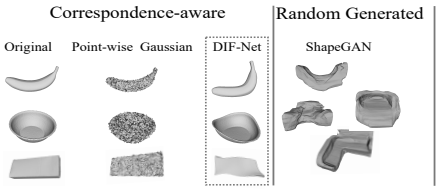
Lastly, we find that ISAGrasp significantly outperforms training without shape augmentation on unseen ShapeNet and GoogleScans objects, and achieves the same performance on RescaledYCB objects (Baseline details are in appendix C). As seen from Table 1, the generalization performance is good, even on completely unseen object instances.

4.2 Real World Experiments.

Next, we evaluated the grasping policy learned in simulation by ISAGrasp, directly in the real world. We perform directly simulation to reality transfer of the learned policy as described in Section 2.4. We evaluate our policy on 22 unseen real world daily objects (see Appendix D) and evaluate each object 5 times with random poses. We report number of success on each object in Table 2. On average, the policy is able to achieve 79% success rate on real world evaluation on novel objects. We observe several failure cases: (1) if the object is more transparent (Box1: container box), the pointclouds are incomplete and the network is less robust. (2) objects which require more careful grasping (Power drill) often have lower success rate. These show that we can leverage policies trained in simulation directly for real world grasping using ISAGrasp and that the robustness and generalization properties transfer from simulation to the real world.

4.3 Ablations and Analysis

Impact of using correspondence-aware generative models. To understand how important using



	Refinement Rate
ShapeGAN(R)	0.09
ShapeGAN(D)	0.29
DIF-Net(R)	0.13
DIF-Net(D)	0.51
DIF-Net(Corr)	**0.76**

Figure 8: Analysis on shape generation. DIF-Net generates realistic and semantically meaningful objects (Left Panel) while ShapeGAN generates novel objects but often unrealistic and without any correspondences Middle Panel). We show the refinement rate for obtaining successful grasps on these objects (**R**: random grasps, **D**: retargeted demonstration, **Corr**: correspondence-guided grasps (right panel). The lack of correspondences makes refinement challenging, yielding only 30% success rate as compared to DIF-Net at 76%.

the dense correspondences provided by DIF-Net are for generating successful grasps on novel objects, we compare DIF-Net [11] with a correspondence-agnostic model, shapeGAN [14] by showing their generated meshes and the refinement rate for obtaining successful grasps. Shown in Figure8 (Left), DIF-Net can smoothly deform the original shapes and generate more realistic shapes. We

conduct 50 times of perturbation with rejection sampling on 100 objects generated by both methods. Figure8 (Right) shows refinement rate (**R**: random grasps without human demonstrations, **D**: using original human demonstrations without new grasps $\mathbf{G_d}$ estimation, **Corr**: using new grasps $\mathbf{G_d}$). Initialized with correspondence-guided grasps, DIF-Net (Corr) achieves 76% refinement rate while ShapeGAN(D) achieves below 30% with the same number of refinements. This indicates the importance of transferring grasps through a deformation based transformation for novel objects.

Table 3: Ablations on input features

	p	p+$\vec{N_o}$	p+$\vec{N_o}$+$\vec{N_f}$ +$\vec{N_p}$
Success	0.47	0.57	0.72

Input Representation. To understand the choice of input representation for supervised learning, we compare feature choices used as input for training policies, evaluated on objects from the ShapeNet and GoogleScans datasets. We observe that using pointcloud p, surface normal vector $\vec{N_o}$, Robot vectors $\vec{N_f}$ and $\vec{N_p}$ performs best.

5 Related Work

Manipulation with Dexterous Hands. To control a dexterous hand, prior work has investigated including planning with analytical models [27] and online trajectory optimization [28]. However, these methods assume accurate dynamics models and robust state estimates, which are difficult to obtain in complex real-world manipulation. Learning-based approaches particularly with deep reinforcement learning (RL) [29, 30, 5, 31, 32, 6, 2] have been investigated. Despite the progress, training deep RL models remains challenging due to high sample complexity and reward engineering. Although this issue has been mitigated by incorporating human demonstrations [33, 34, 3, 35, 1], these methods are still faced with a major challenge in scalability[3, 2]. Prior work has collected demonstrations kinesthetically [34], through VR interfaces [33], or using motion capture (mocap) solutions [36, 37, 38, 3], which are often limited in size, unlike our work which performs augmentations to generate a large augmented dataset.

Data Augmentation and Robustness. Data augmentation has been used traditionally in vision tasks where images are cropped, rotated, normalized, etc [39, 40, 41, 42, 41, 42] to improve generalization and model robustness. In contrast, our work aims to generate novel objects with varying physical dynamics and does not simply learn invariant behavior but learns different grasping behavior for different objects. In a similar vein, domain randomization is used in robotics, where predefined parameters such as lighting, camera are randomized during training[43, 44] in order to learn a policy that is invariant to these parameters. However, this randomization does not aid with generalization to novel object shapes. More recently, methods have aimed to generate novel objects shapes programmatically [13], or use a generative model to create new objects in an adversarial setting [45]. This is. motivated similarly, but differs in being for parallel jaw grasping problems and not generating dynamically consistent grasping behavior via correspondences.

6 Limitations

Our setup assumes an open-loop policy with one single object in the scene, which might not work in a dynamic scene or a cluttered scene. The network takes pointclouds as input, which can require a multi-camera setting for real world testing. The policy is trained for a 22-DOF floating Allegro hand in simulation, and does not consider robot arm constrains. Because of this, the predicted poses might be kinematically infeasible during deployment. Additionally, the grasps considered right now are not functional grasps, which is often the point of grasping with a dexterous hand.

7 Conclusion

We present ISAGrasp, a novel system for learning dexterous grasping policies in the real world. ISAGrasp leverages a data augmentation approach that bootstraps a small number of human demonstrations with a large dataset with diverse and novel objects and grasps. By using a correspondence-aware generative model, we can deform original object shapes and generate dynamically consistent new grasps. We create a large and diverse grasping dataset and train a policy via supervised learning that can then be deployed in simulation and the real world on grasping novel objects, achieving over 75% success rate on grasping novel object instances in the real world.

References

[1] P. Mandikal and K. Grauman. DexVIP: Learning dexterous grasping with human hand pose priors from video. In *CoRL*, 2021.

[2] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In *CoRL*, 2021.

[3] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. DexMV: Imitation learning for dexterous manipulation from human videos. *arXiv preprint arXiv:2108.05877*, 2021.

[4] P. Mandikal and K. Grauman. Learning dexterous grasping with object-centric visual affordances. In *ICRA*, 2021.

[5] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *CoRL*, 2019.

[6] R. Jeong, J. T. Springenberg, J. Kay, D. Zheng, A. Galashov, N. Heess, and F. Nori. Learning dexterous manipulation from suboptimal experts. In *CoRL*, 2020.

[7] A. Mousavian, C. Eppner, and D. Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *ICCV*, 2019.

[8] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[9] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[10] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. *arXiv preprint arXiv:2010.14406*, 2020.

[11] Y. Deng, J. Yang, and X. Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021.

[12] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox. ContactGrasp: Functional multi-finger grasp synthesis from contact. In *IROS*, 2019.

[13] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, et al. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489. IEEE, 2018.

[14] M. Kleineberg, M. Fey, and F. Weichert. Adversarial generation of continuous implicit shape representations. *arXiv preprint arXiv:2002.00349*, 2020.

[15] W. Robotics. Allegro Hand. https://www.wonikrobotics.com/research-robot-hand.

[16] M. Gleicher. Retargetting motion to new characters. In *SIGGRAPH*, 1998.

[17] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020.

[18] E. Coumans and Y. Bai. PyBullet: a Python module for physics simulation for games, robotics and machine learning. https://pybullet.org, 2016–2021.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[20] K. Van Wyk, M. Xie, A. Li, M. A. Rana, B. Babich, B. Peele, Q. Wan, I. Akinola, B. Sundaralingam, D. Fox, B. Boots, and N. Ratliff. Geometric fabrics: Generalizing classical mechanics to capture the physics of behavior. *RA-L*, 2022.

[21] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021.

[22] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, 2015.

[23] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[24] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022.

[25] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[27] Y. Bai and C. K. Liu. Dexterous manipulation using both palm and fingers. In *ICRA*, 2014.

[28] V. Kumar, Y. Tassa, T. Erez, and E. Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *ICRA*, 2014.

[29] V. Kumar, E. Todorov, and S. Levine. Optimal control with learned local models: Application to dexterous manipulation. In *ICRA*, 2016.

[30] A. Gupta, C. Eppner, S. Levine, and P. Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *IROS*, 2016.

[31] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

[32] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[33] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *RSS*, 2018.

[34] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *ICRA*, 2019.

[35] I. Radosavovic, X. Wang, L. Pinto, and J. Malik. State-only imitation learning for dexterous manipulation. In *IROS*, 2021.

[36] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018.

[37] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[38] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020.

[39] G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson. Learning invariances in neural networks from training data. *Advances in Neural Information Processing Systems*, 33:17605–17616, 2020.

[40] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[41] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[42] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[43] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[44] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.

[45] A. Z. Ren and A. Majumdar. Distributionally robust policy learning via adversarial environment generation. *IEEE Robotics and Automation Letters*, 2022.