2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

FuncGrasp: Learning Object-Centric Neural Grasp Functions from Single Annotated Example Object

Hanzhi Chen[1,3] Binbin Xu[2] Stefan Leutenegger[1,3]

***Abstract*— We present *FuncGrasp*, a framework that can infer dense yet reliable grasp configurations for unseen objects using one annotated object and single-view RGB-D observation via categorical priors. Unlike previous works that only transfer a set of grasp poses, *FuncGrasp* aims to transfer *infinite* configurations parameterized by an object-centric continuous grasp function across varying instances. To ease the transfer process, we propose *Neural Surface Grasping Fields* (NSGF), an effective neural representation defined on the surface to densely encode grasp configurations. Further, we exploit function-to-function transfer using sphere primitives to establish semantically meaningful categorical correspondences, which are learned in an unsupervised fashion without any expert knowledge. We showcase the effectiveness through extensive experiments in both simulators and the real world. Remarkably, our framework significantly outperforms several strong baseline methods in terms of density and reliability for generated grasps.**

I. Introduction

When a robot is interacting with the physical world, inferring suitable grasping strategies for objects of interest has been a long-standing problem in the robotics community. In recent years, learning-based methods have significantly boosted the performance of stable grasp detection relying on supervision from a massive amount of training data [1]–[6]. Nevertheless, generating dense yet reliable grasp configurations from these methods usually presents a challenge since rigorous filtering criteria are employed to select the most confident grasp proposals from (partial) observations. This tends to ignore the variability in workspace setups, leading to practically unreachable grasps due to kinematic constraints. Several attempts have been made to address this seemingly "mutually exclusive" process. One line of research explores using categorical priors to generate rich grasp configurations for novel objects. At its core, it first (densely) annotates one source object, then establishes categorical correspondences among instances, which can be realized by learning feature-metric descriptors [7], [8], deformation fields [9], etc. They have demonstrated a substantial potential for few-shot grasp learning. However, as the unseen target objects' inferred grasp density is completely dependent on the size of the set of discrete grasp poses to transfer, it could be impractical to transfer a huge amount of configurations since they demand gradient-based optimization per grasp (e.g., high-dimensional descriptor matching loss in [7]). Another trend of research leverages neural networks to create smooth grasp representations not restricted by resolution. Though existing methods like [6] have demonstrated impressive results in terms of density and reliability, massive datasets are required for better generalization. Moreover, function-level transfer for grasps is often achieved through shallow vector embeddings [6], [10], which can potentially compromise expressiveness. This limitation is evident in several 3D shape modeling works with even more constrained dimensionality [11], [12].

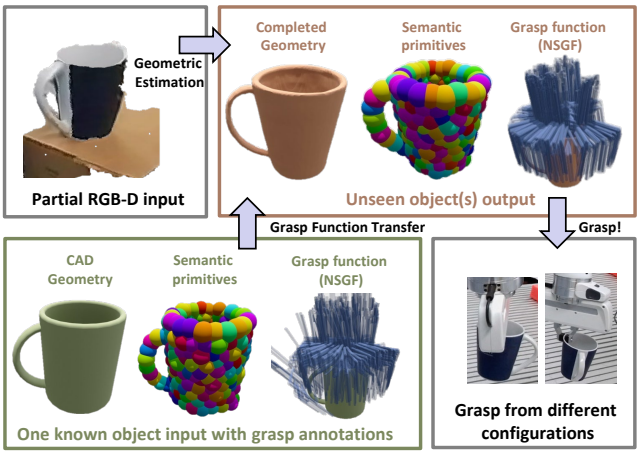

Fig. 1: Given a partial RGB-D input, our framework transfers a known object's continuous grasp function fitted from discrete annotations to the unseen object. We represent such a function using our proposed *Neural Surface Grasping Fields* (NSGF) formulation. This process is achieved by completing the object's geometry and estimating its semantic primitives learned in an unsupervised fashion. Using the transferred NSGF, the robot can query the dense dependable grasp knowledge embedded in a smooth function to conduct grasping from different configurations.

Given all these emerging challenges, our goal is to develop a data-efficient framework that enables robots to infer a wide range of dependable grasp configurations for previously unseen objects by transferring dense grasp knowledge from one single annotated object within the same class. We seek to address two key questions: (1) how to effectively represent dense grasp configurations; and (2) how to accurately transfer such grasp representations, tailored for unseen objects. To this end, we parameterize grasp configurations as object-centric functions, leveraging the power of neural representations to smoothly interpolate between discrete samples. Inspired by [13], we first propose *Neural Surface Grasping Fields* (NSGF) to continuously define grasp configurations on the entire surface. The intuition for NSGF lies in that the grasp pose defined on $\mathrm{SE}(3)$ has much higher dimensions compared to implicit shape representations, and only a very small portion of the object's volume has meaningful results. Hence, we decouple shape and grasp modeling in contrast to previous methods, e.g., [10], [14], by first obtaining the completed

This work was funded by TUM Georg Nemetschek Institute under the project SPAICR.

[1] Smart Robotics Lab, School of CIT, Technical University of Munich. {hanzhi.chen, stefan.leutenegger}@tum.de.
[2] University of Toronto Robotics Institute, University of Toronto. binbin.xu@utoronto.ca
[3] Munich Institute of Robotics and Machine Intelligence (MIRMI).

geometry of the object from [11], then extracting the object surface so that grasp configurations can be effectively embedded in a more bounded space. This also eases the function transfer process thanks to the explicit parameterization of geometry. Moreover, we leverage semantically consistent sphere primitives learned in an unsupervised fashion based on [15] to achieve the transferability of NSGF, respecting the fact that grasp configurations shall not vary much in local surface regions. An overview of our proposed framework is shown in Fig. 1. To summarize, our key contributions are:

- An effective neural representation to encode grasp configurations for surface points, *Neural Surface Grasping Fields* (NSGF), capable of harnessing geometric cues to provide accurate, reliable, and dense grasp poses.
- A novel approach to perform function-level transfer for our proposed NSGF, leveraging semantic primitives learned in an unsupervised manner. To the best of our knowledge, we are the first to design a feasible paradigm to transfer continuous grasp functions instead of using shallow vector embeddings.
- Extensive experiments in simulators and the real world to validate the effectiveness of our framework, *FuncGrasp*.

II. Related works

Model-free grasp detection. Without the requirement for CAD models, model-free grasp detection methods aim to produce dense point(pixel)-wise predictions on grasping quality and poses from sensor observations. GPD [4] and PointNetGPD [2] leverage deep neural networks to predict the scores of grasp pose candidates obtained from the pointcloud. VGN [16] is fed with a TSDF representation of the scene and in turn, outputs structured voxelgrids with grasp quality, orientation, and width. 6-DoF GraspNet [3] models grasp detection as a generative process and uses a variational auto-encoder (VAE) to produce grasp proposals. ContactGraspNet [13] eases the grasp learning process by proposing a more structured grasp representation. Notably, those methods usually require huge amounts of data with grasp annotations for supervision, which can be time-consuming even when employing advanced physics simulators. Our method only requires annotations for one object from each category, eliminating the need for large-scale simulations to acquire grasp labels.

Category-level grasp learning. Another line of research conducts model-free grasp learning from an object-centric perspective. Those works usually assume consistent categorical priors among objects to establish meaningful correspondence. CaTGrasp [5] proposes non-uniform normalized object coordinates for better correspondence learning so as to build grasp codebooks in canonical space. DON [8] and NDF [7] leverage neural networks to learn deep descriptors to discover categorical correspondence in a self-supervised manner. kPAM [17] establishes categorical correspondence through learning manually annotated 3D key points. TransGrasp [9] proposes to use implicit deformations fields to infer grasps for novel objects from a pre-labeled instance. Instead of transferring a pre-labeled set of discrete grasp poses as in previous works, our approach aims to infer a continuous grasp function for novel objects, which provides dense yet reliable configurations and is not limited by resolution.

Neural representations in grasping. Recently neural representations have shown a strong capacity for novel view synthesis [18]–[20], 3D reconstruction [21]–[23], and generative modeling [24], [25]. At their core, multi-layer perceptrons (MLP) are used to encode the scene content of every 3D point. In the context of robotic grasping, such scene contents can be represented using geometric information (density, occupancy, (un-)signed distance), deep descriptors, and grasp configurations. Most attempts [26]–[28] use neural representations to learn the accurate geometry of photometrically challenging objects with depth sensing failures. [7], [29] try to use neural fields to discover distinctive descriptors. NGDF [6] models the grasp distribution as a distance field. [10] uses neural radiance fields (NeRF) to unify the learning process of shape, appearance, and grasp. As the grasp poses are distributed in unconstrained SE(3) space and tend to lie near the surface, in our NSGF formulation, we propose to define them explicitly on the surface instead of the volume so as to increase the function's expressiveness and ease the transfer process.

Neural fields transfer. With the prevalence of neural representations, a few works started to explore approaches to conduct function-level transfer. NeSF [30] introduces an approach to transfer neural density fields to semantic fields through voxel grids-based field approximation. NFGP [31] conducts geometry processing tasks, e.g., shape deformation, directly on neural signed distance fields through invertible neural networks. Inspired by NeSF [30], we leverage sphere primitives possessing semantic consistency to approximate our proposed NSGF to achieve flexible field transfer.

III. Methods

Assuming one source object with one grasp function pre-fitted to annotated grasp labels that can be obtained manually or from simulations, our objective is to conduct grasp function transfer to several unseen target objects of the same type respecting their geometric features. For each target object, our pipeline only requires a partial pointcloud extracted from a single-view RGB-D frame. We introduce geometric inference to predict completed geometry, shape-aware grasp confidence, and semantic primitives in Sec. III-A. In Sec. III-C, we explain the proposed approach that uses primitive-based shape abstraction to transfer our formulated object-centric grasp function presented in Sec. III-B. The implementation details are provided in Sec. III-D. Detailed illustration of our framework is provided in Fig. 2.

A. Single-view Geometric Inference

Geometric estimation. Given a single-view RGB-D image input together with a foreground mask for the object of interest provided by an off-the-shelf detector [32], we acquire a segmented pointcloud $\mathcal{X} = \{\mathbf{x}^1, ..., \mathbf{x}^{N_\mathcal{X}}\}$ and its voxelized TSDF volume $\mathbf{V}$ for each object in the scene. We further feed $\mathcal{X}$ to a pre-trained pose estimator [33] to acquire its 7-DoF pose $\mathbf{T} \in \mathrm{SIM}(3)$ from its canonical system to the camera

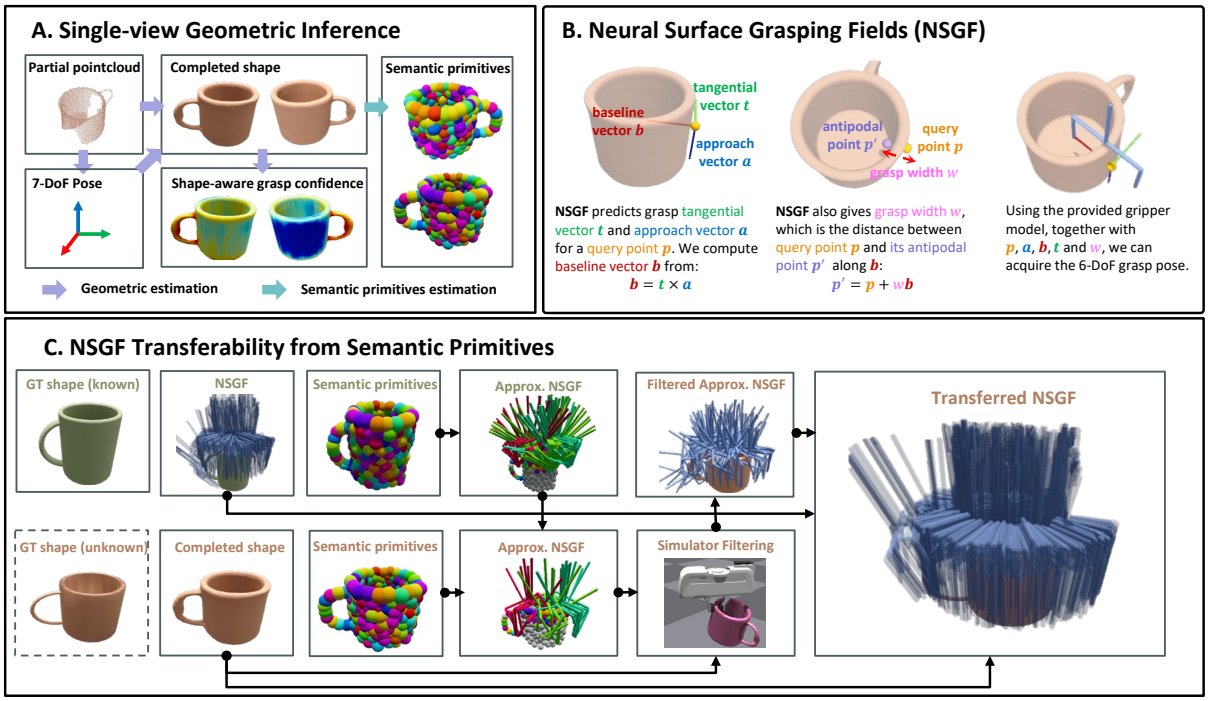

Fig. 2: Illustration of our proposed framework, *FuncGrasp*. (A) Our geometric estimation module infers the target object's 7-DoF pose, completed shape, semantic primitives, and shape-aware confidence (red indicates low; blue indicates high). (B) Our Neural Surface Grasping Field (NSGF) formulation defines point-wise grasp configurations on the surface. (C) We approximate NSGF using semantic primitives. We sample a small number of grasp configurations for each primitive and transfer them to the target object using the corresponding primitive (grippers colored magenta, green, and cyan indicate three different primitives). After adjusting the transferred grasps based on the object's shape and filtering invalid samples in the simulator, we fit a new NSGF using the rest of the samples to achieve grasp function transfer.

system. $\mathbf{V}$ is then canonicalized with $\mathbf{T}$ and passed to the shape completion module from [11] to acquire the complete geometry represented by a voxel grid filled with occupancy probabilities $\mathbf{O}$. The object mesh $\mathcal{M}$ is extracted from $\mathbf{O}$ using multi-resolution iso-surface extraction strategy from [22]. We can also compute the shape confidence for mesh surface point $\mathbf{p}$ represented by the norm of their gradients w.r.t the occupancy probabilities, i.e., $||\partial \mathbf{O}[\mathbf{p}]/\partial \mathbf{p}||_2$. Shape confidence will later be used to rank grasp poses decoded from our formulated object-centric grasp function.

Unsupervised semantic primitives learning. Inspired by recent advances in shape manipulation [15], we establish the categorical correspondence by exploring the shape abstraction in a semantically consistent fashion. Specifically, an object's geometry can be approximated by a fixed number of spherical primitives and this abstraction leads to part-to-part correspondence (c.f. color-coded semantic primitives in Fig. 2-C). More importantly, such correspondence labels can be learned in an unsupervised manner without any expert knowledge in contrast to [17]. After fitting the primitive-based representation for each object used for training, we assign the closest primitive label to every surface point. During inference, we use a part segmentation network to estimate point-wise semantic primitive labels for uniformly sub-sampled surface points. For points with the same label, we pass them to Mean Shift to compute the clustering center, which is the final predicted primitive center location. All primitive centers are denoted as $\mathcal{S} = \{\mathbf{s}^1, ...\mathbf{s}^{N_\mathcal{S}}\}$, where $N_\mathcal{S}$ is the number of pre-defined primitives. Here we obtain a geometric representation $\mathcal{G}$ for each unseen object with a tuple $\mathcal{G} = (\mathbf{T}, \mathcal{M}, \mathcal{S})$. Note $\mathcal{M}, \mathcal{S}$ are represented in the canonical system.

B. Neural Surface Grasping Fields (NSGF)

NSGF formulation. We propose to use a continuous function to represent grasp configurations. This function is parameterized by an MLP and defined on the object surface to map the point to its grasp validity and grasp pose. We refer to such representation as *Neural Surface Grasping Fields* (NSGF). As shown in Fig. 2-B, inspired by [13], for each point $\mathbf{p}$, NSGF outputs its grasp validity q, grasp approach vector $\mathbf{a}$, grasp tangential vector $\mathbf{t}$ and grasp width w. Grasp baseline vector $\mathbf{b} = \mathbf{t} \times \mathbf{a}$. For each point, the grasp pose can be solved using its coordinate $\mathbf{p}$, the predicted width w, rotation vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{t}$, and the gripper model. We notice that such representation often fails when dealing with thick objects like bottles as the predicted width has no geometric awareness. Different from [13], we harness the completed geometry to acquire a more precise grasp width from raw prediction w_{coarse}. As shown in Fig.3-A, the antipodal point $\mathbf{p}'$ is initialized along the baseline vector $\mathbf{b}$, i.e., $\mathbf{p}' = \mathbf{p} + w_{\text{coarse}}\mathbf{b}$. Then we search for width offset Δw so that $\mathbf{p}'$ can be moved to the nearby surface along $\mathbf{b}$, i.e., $\mathbf{p}' = \mathbf{p} + (w_{\text{coarse}} + \Delta w)\mathbf{b}$, and the final grasp width is $w = w_{\text{coarse}} + \Delta w$ (*c.f.* Fig. 3-B,C). In practice, we query the occupancy value to determine if a point lies on the surface. With a slight abuse of notation, we denote NSGF as:

$$F(\mathbf{p}) \colon S^2 \rightarrow \mathbb{R} \times \mathrm{SE}(3). \tag{1}$$

Notably, NSGF is defined directly on the 2D surface space instead of bounding volume as [10], [14] and thus avoids extensive forward-passing and post-processing to locate stable grasps inherently near the surface.

NSGF fitting. To fit an NSGF F for one object, validity loss (l_v) and rotation vectors loss (l_r) are adopted from [13]. Note

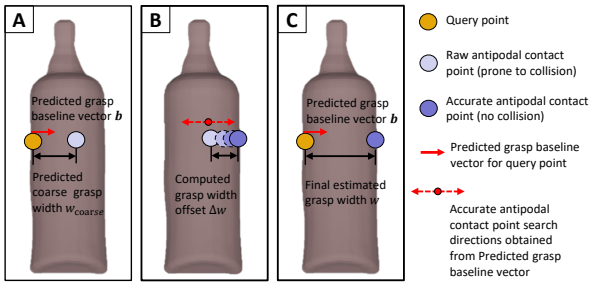

Fig. 3: (A) Raw antipodal contact point using predicted coarse width w_{coarse} without geometric awareness, leading to grasp failure due to collision. (B) Use of completed geometry to search for nearby on-surface points. (C) Accurate antipodal contact point thanks to precise shape completion, yielding a collision-free grasp pose.

in l_{r}, we regress the tangential vector instead of the baseline vector as we empirically find it helps encode more stable grasp poses. With the ground-truth antipodal contact point $\mathbf{p}'_{\text{gt}}$, coarse width regression loss for each valid point is:

$$l_{\text{w}} = \sum_{w_{\text{coarse}}} ||\mathbf{p} + w_{\text{coarse}}\mathbf{b} - \mathbf{p}'_{\text{gt}}||_2^2. \tag{2}$$

Besides, we regularize the predicted baseline vector $\mathbf{b}$ with l_{reg} to align it with the point normal $\mathbf{n}$ per valid point:

$$l_{\text{reg}} = \sum_{\mathbf{b}} \min_{s\in\{-1,1\}} ||s\mathbf{n} - \mathbf{b}||_1 + ||1 - s\mathbf{n}^T\mathbf{b}||_1. \tag{3}$$

The final fitting loss is given as $l = l_{\text{v}} + l_{\text{r}} + l_{\text{w}} + \lambda l_{\text{reg}}$.

A full object-centric representation $\mathcal{I}$ for each instance is given as $\mathcal{I} = (\mathcal{G}, F)$.

C. Transferability of NSGF from Semantic Primitives

Given NSGF F_{src} of one labeled source object $\mathcal{I}_{\text{src}}$ pre-fitted to its grasp labels, we aim to derive a tailored grasp function F_{tgt} for an unseen target object $\mathcal{I}_{\text{tgt}}$.

NSGF transfer via semantic primitives. We probe the source NSGF F_{src} using semantic primitives $\mathcal{S}_{\text{src}}$ because they greatly help to sample a small yet informative subset of grasps that can express the valid domains of the NSGF. Specifically, each source primitive center $\mathbf{s}^j_{\text{src}}$ is assigned with N_j valid grasps decoded from F_{src} ($N_j \leq 5$ for the trade-off between accuracy and speed). The criteria to assign primitive labels for grasps are based on the distance between the left (gripper finger) contact points and the primitive centers. F_{src} is hence approximated as $\overline{\mathcal{F}}_{\text{src}}$ with:

$$\begin{aligned} \overline{\mathcal{F}}_{\text{src}} &= \{\overline{\mathcal{F}}^j_{\text{src}} \mid j \in \{1, ..., N_{\mathcal{S}}\}\}, \\ \overline{\mathcal{F}}^j_{\text{src}} &= \{\mathbf{g}^{j,k}_{\text{src}} \mid k \in \{1, \ldots, N_j\}\}. \end{aligned} \tag{4}$$

where $\mathbf{g}^{j,k}_{\text{src}}$ is the k-th grasp pose from the j-th primitive.

Without loss of generality, we exemplify the approximate NSGF transfer for grasp poses from one primitive, i.e., obtaining $\overline{\mathcal{F}}^j_{\text{tgt}}$ from $\overline{\mathcal{F}}^j_{\text{src}}$. For each source grasp $\mathbf{g}^{j,k}_{\text{src}}$ assigned to j-th primitive based on the left contact point, we compute the primitive label for its right contact point, denoted as j'. The primitive centers closest to left and right contact points are hence $\mathbf{s}^j_{\text{src}}$ and $\mathbf{s}^{j'}_{\text{src}}$, and the corresponding centers from the target object are $\mathbf{s}^j_{\text{tgt}}$ and $\mathbf{s}^{j'}_{\text{tgt}}$. We first compensate the grasp translation with the averaged primitives offset $\Delta\mathbf{s}$: $\mathbf{g}^{j,k}_{\text{tgt}}[\mathbf{t}] = \mathbf{g}^{j,k}_{\text{src}}[\mathbf{t}] + \Delta\mathbf{s}$, $\Delta\mathbf{s} = (\mathbf{s}^j_{\text{tgt}} - \mathbf{s}^j_{\text{src}} + \mathbf{s}^{j'}_{\text{tgt}} - \mathbf{s}^{j'}_{\text{src}})/2$. Then we compute the two gripper finger contact points' normals $\mathbf{n}^{j,k}_{\text{tgt}}, \mathbf{n}^{j,k'}_{\text{tgt}}$ on the target object using the grasp $\mathbf{g}^{j,k}_{\text{tgt}}$ translated with $\Delta\mathbf{s}$, and align the grasp baseline vector computed from the rotation of $\mathbf{g}^{j,k}_{\text{tgt}}$ to $\mathbf{n}^{j,k}_{\text{tgt}} - \mathbf{n}^{j,k'}_{\text{tgt}}$ to respect the antipodal principle. We repeat this process for other primitives with valid grasps and acquire an approximated NSGF $\overline{\mathcal{F}}_{\text{tgt}}$ (*c.f.* Approx. NSGF in Fig. 2-C). As the geometric estimation module provides accurate geometry, we also feed the object mesh $\mathcal{M}_{\text{tgt}}$ and $\overline{\mathcal{F}}_{\text{tgt}}$ to a GPU-accelerated parallel simulator [34] with free-floating grippers to filter out unstable grasps and acquire a better approximation, $\hat{\mathcal{F}}_{\text{tgt}}$, for the target object's NSGF. Finally, we load the pre-fitted weight of the source NSGF F_{src} and fit new NSGF for the target object using samples from $\hat{\mathcal{F}}_{\text{tgt}}$ with much fewer iterations (5 times less).

Inference. We uniformly sample 5k points on the object surface and feed them in parallel to the transferred NSGF F_{tgt} to obtain all grasp configurations. We further select the valid grasp poses with the following criteria: (1) predicted as valid ($q > 0$); and (2) kinematically reachable without collision to the scene. We rank them using the shape-aware grasp confidence based on the gradient response introduced in Sec. III-A (*c.f.* "Shape-aware grasp confidence" in Fig. 2-A), and choose the highest-ranked one.

D. Implementation Details

Data preparation. The 3D models are provided by ShapeNet repository [35] with shape augmentation from [9]. This resulted in 1,548 mugs, 1,296 bowls, and 1,275 bottles for training. We use 3D models with their rendered depth data to train the shape completion network, and 3D models to train the part segmentation network for semantic primitives. The source objects with grasp annotations, one per category, are picked randomly from the ACRONYM dataset [36].

Network architecture. We use an off-the-shelf object detector [37] and pose estimator [33]. For the shape completion network, we adopt the same architecture as [11]. The semantic primitives' segmentation network is modified from a 3D-GCN [38] with 2,048 points as input, and outputs one-hot logit vectors for 256 primitive labels. The NSGF is designed using SIREN backbone [39], where an 8-layer SIREN network takes a 3-dimensional point coordinate and a 16-dimensional geometric feature trilinearly interpolated from the shape completion network as input and outputs a 128-dimensional feature. Then the feature is fed to two individual 4-layer SIREN networks to output 6-dimensional rotation vectors [40] and the coarse grasp width. An additional 4-layer SIREN network is used to predict the grasp validity.

Training protocol. Training of the shape completion network follows [11]. The semantic primitives' segmentation network is trained for 10k iterations with a batch size of 12 using Adam [41] with a learning rate of 0.0005. The source object's NSGF fitting takes 200 iterations and uses Adam [41] with a learning rate of 0.0001. $\lambda = 0.1$ is set in the fitting loss l.

Transfer protocol. NSGF transfer to the target object loads the pre-fitted weights of the source object's NSGF and is only trained for 40 iterations. Both NSGF fitting and transfer are fed with 2k points per iteration.

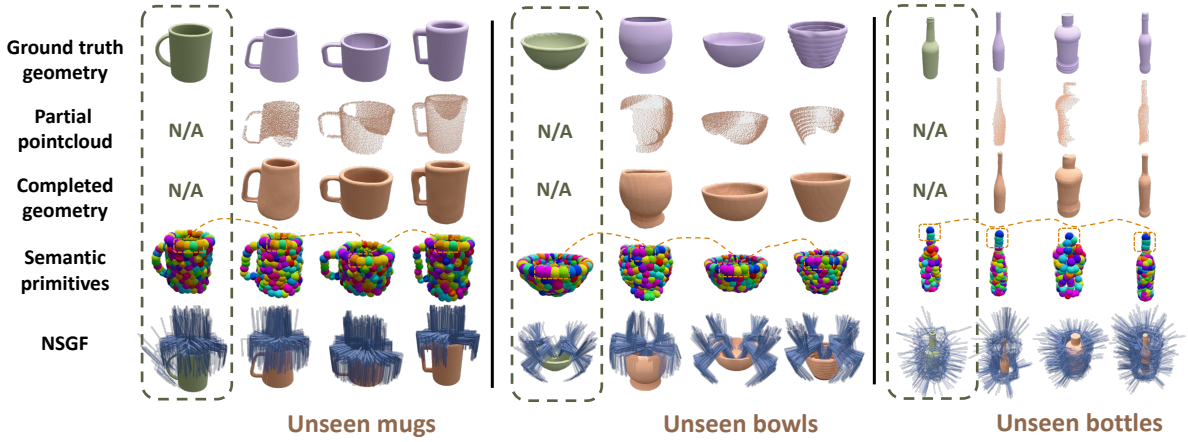

Fig. 4: Qualitative results for each category tested in simulations. Source objects with grasp annotations are marked within the green dotted boxes. Their NSGFs are fitted with valid labels. For every unseen object, we visualize its ground-truth geometry (inaccessible for unseen objects), completed geometry inferred from partial pointcloud, semantic primitives, and the transferred NSGF. We highlight the semantic primitives-based correspondence among objects with orange dotted boxes and lines (zoom in for details). Note here we intentionally down-sample the grasps inferred by NSGF by a factor of 10 compared to the inference time so that the actual ground-truth geometry is visible.

IV. Experiments

In this section, we aim to understand the effectiveness of our developed framework. Three different household object categories are tested: Mugs, Bowls, and Bottles. We use the same splits and annotated objects as [9].

A. Experimental Setups

Testing environment. The simulator is built on top of IsaacGym [34] to evaluate our method and several baselines. The real robotic system consists of a 7-DoF Franka Panda robot and an Azure Kinect DK RGB-D camera (*c.f.* Fig. 5-A).

Baselines. For a fair comparison, the baseline methods shall be able to generate sufficiently dense grasp configurations for each object as ours. To this end, our framework is compared against the following representative works: 6-DoF GraspNet (6DGN) [3], Contact GraspNet (CGN) [13] and TransGrasp [9]. 6DGN [3] and CGN [13] are trained with grasp annotations and agnostic to categories. TransGrasp [9] uses one labeled instance and leverages categorical priors as ours. As our generated grasps are defined as a continuous function, we uniformly select 5k points on each object and query their grasp validity and poses from their own NSGFs. We encounter difficulties in comparing our work with other notable works, such as NGDF [6], primarily because it requires dense grasp annotations for all objects during training to establish grasp manifolds. Additionally, it lacks a robust strategy for accurately transferring grasps across instances in a label-efficient manner.

Evaluation protocols. Our grasping evaluation procedures are categorized into two different setups.

- **Omni-grasp**: In this setup, we report the success rate and the size of all generated grasps for each object so as to investigate both the reliability and the density of the generated results. We compute the success rate of one object as the ratio of the successful grasps among all valid grasps as [9]. For each category, the success rate for this setup is defined as $s_{\text{omni}} = {}^{1}/_{N_{\text{cat}}} \sum_{i}^{N_{\text{cat}}} {}^{N^i_{\text{succ}}}/_{N^i_{\text{all}}}$ where N_{cat} is the number of instances, N^i_{succ} and N^i_{all} are the successful grasps and all valid grasps for each object, respectively.
- **Best-grasp**: In this setup, we follow the commonly used evaluation protocol in previous works by selecting the highest-scored grasp configuration among all candidates and grasping each object once. The success rate is hence defined as $s_{\text{best}} = {}^{N^{\text{succ}}_{\text{cat}}}/_{N_{\text{cat}}}$, where $N^{\text{succ}}_{\text{cat}}$ is the number of successfully grasped objects using the selected pose.

A grasp is considered successful if the object is lifted for more than 15 seconds.

B. Results and Discussions in Simulators

Rows 1-4 of Table I show the evaluation results in the simulator for the two setups, i.e., omni-grasp and best-grasp.

Omni-grasp evaluation. We see that methods trained with large-scale datasets (6DGN [3] and CGN [13]) perform poorly in omni-grasp evaluation with fewer grasp candidates (859.57 and 119.60) and lower success rate (55.97% and 72.60%). As they aim to identify the optimal grasp, they are expected to rigorously filter out the majority of generated results.

Moreover, they do not retrieve the geometric information of the objects through 3D completion as ours, so occasionally, the gripper could collide with invisible parts of the objects. Such limitation becomes even more evident when CGN [13] deals with bottles as it has an assumption on the thickness of the objects and thus generates fewer poses than the others. TransGrasp [9] greatly improves the performance through topology-aware optimizations for grasp poses with a success rate of 80.82%. However, the size of their grasp proposals for each object is fixed with the number of labeled grasp poses from the source annotated object (992 on average). In contrast, we only use a much smaller subset of the grasps (364.30 on average) for transfer, thanks to the geometric abstraction from primitive-based shape representation. By further leveraging continuous neural representations, the transferred NSGF can smoothly interpolate between discrete samples and produce denser grasp poses than other baseline methods (1431.38 on average). Even with a considerably larger number of grasp poses for evaluation, our success rate still significantly outperforms the strong baseline (TransGrasp [9]) by 10.30%. Besides the effectiveness of our proposed paradigm for grasp transfer with NSGF representation, we also attribute it to

	Omni-grasp				Best-grasp			
	Mug	Bowl	Bottle	Avg.	Mug	Bowl	Bottle	Avg.
6-DoF GraspNet (6DGN) [3]	37.77% (463.30)	53.84% (968.25)	76.30% (1147.17)	55.97% (859.57)	42.85%	68.89%	78.57%	63.44%
Contact GraspNet (CGN) [13]	68.32% (58.16)	76.65% (177.73)	72.84% (122.91)	72.60% (119.60)	69.04%	88.89%	76.42%	78.12%
TransGrasp [9]	88.06% (612.00)	68.31% (**1000.00**)	86.10% (1364.00)	80.82% (992.00)	92.07%	78.88%	88.57%	86.51%
Ours	**94.48% (801.85)**	**92.13%** (782.55)	**86.75% (2709.75)**	**91.12% (1431.38)**	**95.23%**	**95.56%**	**91.42%**	**94.07%**
Ours w/o width from completion (A1)	89.56% (794.31)	86.62% (764.33)	69.53% (2627.49)	81.90% (1395.37)	-	-	-	-
Ours w/o pre-fitting (A2)	91.04% (806.18)	89.42% (668.55)	83.05% (2643.02)	87.83% (1372.58)	-	-	-	-
Ours w/o simulator filtering (A3)	91.44% (906.09)	87.48% (1013.52)	85.09% (2854.03)	88.00% (1591.21)	-	-	-	-

TABLE I: Grasp success rate results of the two evaluation metrics (omni-grasp and best-grasp) tested in simulators. Rows 1-4: Comparison with other methods in simulation. Rows 4-7: Ablation study in simulation. For omni-grasp evaluation results, the numbers of generated grasp poses are given in brackets.

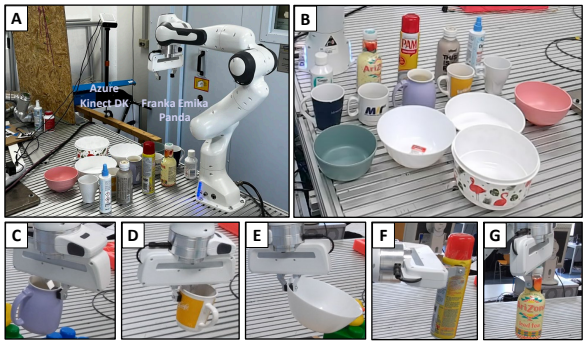

Fig. 5: (A) Physical setup for real robot experiments. (B) Tested objects, five instances per category. (C)-(G) Examples of successful grasps.

the simulator filtering for unstable grasp poses thanks to the accurate geometry from the geometric estimation module. All these results further showcase the impressive ability of our framework to infer reliable yet dense grasp configurations for a wide range of novel objects. Fig. 4 further exemplifies the inferred geometry, semantic primitives, and NSGFs for several unseen objects.

Best-grasp evaluation. Our approach reports the best performance for all categories compared to the others. One major reason is that our generated grasp poses are highly reliable, as shown in the omni-grasp evaluation. Besides, our shape-aware grasp selection strategy based on gradient response helps avoid grasping object regions with noisy observations or unsmooth regions, e.g., the handles of the mugs, which are usually more prone to grasping failures as we noticed in simulators (*c.f.* "Shape-aware grasp confidence" in Fig. 2-A).

C. Real Robot Experiments

As it would be practically infeasible to perform omni-grasp evaluation in the real world, we fix the object pose in the workspace and test the five furthest valid grasps based on queried points' distance sequentially for approximate verification. For the best-grasp evaluation, we randomly place each object in five arbitrary positions and execute grasping using the inference strategy we suggested. We evaluate five different instances per category (*c.f.* Fig. 5-B). Both omni-grasp and best-grasp had five trials per object and 25 trials for each category. For the omni-grasp evaluation, we report an overall success rate of 90.66%, with Mug, Bowl, and Bottle achieving 23/25, 24/25, and 21/25, respectively. The best-grasp evaluation gives a success rate of 93.33% with each category being 23/25 (Mug), 25/25 (Bowl), and 22/25 (Bottle). These results reflect a similar pattern as the simulations and further suggest though only a limited amount of data is provided (several 3D object meshes and one with grasp labels), our proposed framework is capable of producing dependable grasp poses for real-world objects regardless of appearance, geometry, and positions (*c.f.* Fig. 5-(C-G)). Currently, our approach takes *ca.* 2.71s to fit NSGF for one target object, which includes parallel simulator filtering. Subsequently, the inference time for the fitted NSGF is 0.0043s. These metrics are evaluated on a single NVIDIA GeForce RTX 3080 GPU.

D. Ablation Study

We conduct ablation experiments to validate crucial components' contributions in rows 4-7 of Table I. For *Ours w/o width from completion* variant (A1), we remove the precise width calculation via the completed shape; for *w/o pre-fitting* variant (A2), we fit the NSGF for unseen objects from scratch instead of loading the pre-fitted NSGF weights of the source annotated object; for *w/o simulator filtering* variant (A3), we skip the simulator filtering before target NSGF fitting. We highlight the effectiveness of shape completion to obtain reliable grasps. As shown in A1, grasping thick objects like bottles easily fails (decrease from 86.75% to 69.53%) because the gripper tends to collide due to non-geometry-aware width prediction. Besides, loading the pre-fitted weight of the NSGF from the source object helps faster convergence with 87.83% compared to 91.12% of the full model (*c.f.* A2). Simulator filtering (A3) shows its effectiveness with an increase in the success rate of 3.12% because the inferred shape is sufficiently accurate so it aids in eliminating false positive samples.

V. Conclusion

In this work, we propose *FuncGrasp*, a framework to infer dense yet reliable grasp configurations for unseen objects and requires only one annotated object and a single-view pointcloud. With our *Neural Surface Grasping Fields* formulation, grasp configurations can be effectively embedded on the object surface. This formulation further eases the transfer effort for continuous function-based grasp representation. Using the semantic primitives learned in an unsupervised fashion, we successfully translate smoothly distributed configurations encoded in the grasp function from one single annotated object to several novel instances. The effectiveness is validated through extensive grasping experiments in both simulations and the real world.

For limitations, the transferred NSGFs could fail to produce grasp configurations for certain regions of some instances, e.g., handles of the mugs. We observe this is occasionally caused by the simulator filtering step. Furthermore, our framework relies on the fidelity of the physics simulator for seamless sim-to-real transfer. Hence, utilizing a less precise simulator could lead to performance degradation. In future work, we aim to enhance the speed using techniques like vectorized training [42], thereby aligning with low-latency demands.

REFERENCES

[1] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4461–4468.

[2] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: Detecting grasp configurations from point sets," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[3] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[4] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[5] B. Wen, W. Lian, K. Bekris, and S. Schaal, "Catgrasp: Learning category-level task-relevant grasping in clutter from simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6401–6408.

[6] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1814–1821.

[7] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

[8] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *CoRL*, 2018.

[9] H. Wen, J. Yan, W. Peng, and Y. Sun, "Transgrasp: Grasp pose estimation of a category of objects by transferring grasps from only one labeled instance," in *European Conference on Computer Vision*. Springer, 2022, pp. 445–461.

[10] V. Blukis, T. Lee, J. Tremblay, B. Wen, I. S. Kweon, K.-J. Yoon, D. Fox, and S. Birchfield, "Neural fields for robotic object manipulation from a single image," *arXiv preprint arXiv:2210.12126*, 2022.

[11] B. Xu, A. J. Davison, and S. Leutenegger, "Learning to complete object shapes for object-level mapping in dynamic scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2257–2264.

[12] J. Wang, M. Rünz, and L. Agapito, "Dsp-slam: Object oriented slam with deep shape priors," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1362–1371.

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

[14] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *Robotics: science and systems*, 2021.

[15] Z. Hao, H. Averbuch-Elor, N. Snavely, and S. Belongie, "Dualsdf: Semantic shape manipulation using a two-level representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7631–7641.

[16] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[17] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "KPAM: Keypoint affordances for category-level robotic manipulation," *ISRR*, 2019.

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[19] H. Chen, F. Manhardt, N. Navab, and B. Busam, "Texpose: Neural texture learning for self-supervised 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4841–4852.

[20] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *CVPR*, 2021.

[21] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[22] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[23] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020.

[24] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] K. Jo, G. Shim, S. Jung, S. Yang, and J. Choo, "Cg-nerf: Conditional generative neural radiance fields," *arXiv preprint arXiv:2112.03517*, 2021.

[26] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," in *5th Annual Conference on Robot Learning*, 2021.

[27] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*, 2022.

[28] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "Nerf-supervision: Learning dense object descriptors from neural radiance fields," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6496–6503.

[29] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," *arXiv preprint arXiv:2302.03573*, 2023.

[30] S. Vora, N. Radwan, K. Greff, H. Meyer, K. Genova, M. S. M. Sajjadi, E. Pot, A. Tagliasacchi, and D. Duckworth, "Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes," 2021.

[31] G. Yang, S. Belongie, B. Hariharan, and V. Koltun, "Geometry processing with neural fields," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 483–22 497, 2021.

[32] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9799–9808.

[33] G. Li, Y. Li, Z. Ye, Q. Zhang, T. Kong, Z. Cui, and G. Zhang, "Generative category-level shape and pose estimation with semantic primitives," in *Conference on Robot Learning*. PMLR, 2023, pp. 1390–1400.

[34] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[35] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[36] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A large-scale grasp dataset based on simulation," in *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[37] R. B. Rusu and S. Cousins, "Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*, 2011, pp. 1–4.

[38] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1800–1809.

[39] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. NeurIPS*, 2020.

[40] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] R. Z. Horace He, "functorch: Jax-like composable function transforms for pytorch," https://github.com/pytorch/functorch, 2021.