# Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input [Extended Abstract]

Jung-Su Ha Danny Driess Marc Toussaint
Learning & Intelligent Systems Lab, TU Berlin, Germany

***Abstract*—We propose to represent objects as image-conditioned neural implicit functions and to model manipulation constraints on top of such implicit functions. The proposed framework, which we call Deep Visual Constraint, acts as as a perception component in the whole manipulation pipeline, enabling long-horizon planning *only from visual input*. Project page: https://sites.google.com/view/deep-visual-constraints**

## I. Introduction

Manipulation planning is a type of motion planning problem that computes not only the robot's own movement but also objects' motions *subject to* interaction constraints. At the core of robot's dexterity and generalization capability is designing interaction constraints which describe how the robot should interact with objects in a plausible way, such as grasping and placing an object, or more general tool-use.

Traditional constraint models, or constraint features, for manipulation planning were built upon geometric object representations such as meshes or combinations of shape primitives representing its shape in conjunction with its pose in $SE(3)$. However, this traditional approach places a number of limitations on its perception and generalization: (i) The representations have to be inferred from raw sensory inputs like images or point clouds – raising the fundamental problem of perception and shape estimation. (ii) With increasing generality of object shapes and interaction, the complexity of representations grows and hand-engineering of the interaction features becomes inefficient. However, if the aim is manipulation skills, the hard problem of precise shape estimation might be unnecessary to predict accurate interaction features.

Inspired by recent advances in 3D modeling, e.g. NeRF [7], we propose a data-driven approach to learning interaction features that are conditioned on object images. The whole constraint model is trained end-to-end directly with the task supervisions so as to make the representation and perception *task-specific* and thus to simplify the interaction prediction. The object representation, which we propose to be a $d$-dimensional neural implicit function over the 3D space, acts as a bottleneck and is shared across multiple features so that the *task-agnostic* aspects can emerge. In particular, this implicit function over 3D is associated with the 2D images from multiple cameras (e.g. stereo) via the known camera geometry. We demonstrate that integrating the learned constraint models into Logic-Geometric Programming (LGP) [12] enables computing dexterous manipulation plans involving various interactions with complex-shaped objects. Since the representations generalize well, the learned constraint models are directly applicable to manipulation of unseen objects.

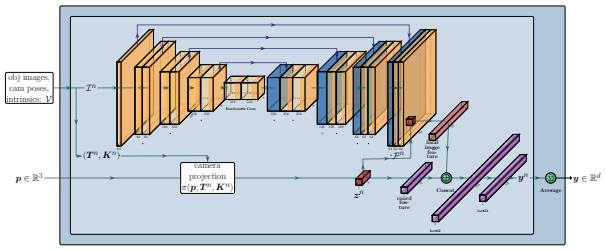



**Fig. 1:** PIFO

## II. Deep Visual Constraints

### A. Pixel-Aligned Implicit Functional Object (PIFO)

Given $N_{\text{view}}$ images with their camera poses/intrinsics, $\mathcal{V} = \{(\mathcal{I}^1, \boldsymbol{T}^1, \boldsymbol{K}^1), ..., (\mathcal{I}^{N_{\text{view}}}, \boldsymbol{T}^{N_{\text{view}}}, \boldsymbol{K}^{N_{\text{view}}})\}$, the proposed implicit object representation is a mapping:

$$\psi(\boldsymbol{p}; \mathcal{V}) = \boldsymbol{y}, \tag{1}$$

where $\boldsymbol{p} \in \mathbb{R}^3$ and $\boldsymbol{y} \in \mathbb{R}^d$ are a queried 3D position and a representation vector at that point, respectively. This implicit function, implemented as a neural network as depicted in Fig. 1, consists of three parts: (i) **Image Encoder** transforms a color image to a feature image. We adopted the hourglass network architecture architecture [10] so as to capture both local and global information in the image, i.e.,

$$\mathcal{F}^n = UNet(\mathcal{I}^n), \ \forall n \in \{1, ..., N_{\text{view}}\}. \tag{2}$$

(ii) **3D Reprojector** first transforms a queried point $\boldsymbol{p}$ into the image coordinate including depth, $\pi(\boldsymbol{p}; \boldsymbol{T}, \boldsymbol{K}) = \boldsymbol{z} \in \mathbb{R}^3$ and then extracts the local image feature at the projected pixel point via bilinear interpolation, which is then fed into a couple of fully connected layers to compute a representation vector for a single image, i.e., $\forall n \in \{1, ..., N_{\text{view}}\}$,

$$\boldsymbol{y}^n = MLP(\mathcal{F}^n(\boldsymbol{z}^n), \boldsymbol{z}^n), \ \boldsymbol{z}^n = \pi(\boldsymbol{p}; \boldsymbol{T}^n, \boldsymbol{K}^n). \tag{3}$$

(iii) **Feature Aggregator** combines representation vectors from multiple images simply by taking average, i.e., $\boldsymbol{y} = \frac{1}{N_{\text{view}}} \sum_{n=1}^{N_{\text{view}}} \boldsymbol{y}^n$.

### B. Interaction Feature Prediction

An interaction feature is also a neural implicit function:

$$h = \phi_{\text{task}}(\boldsymbol{q}; \mathcal{V}), \tag{4}$$

where $\boldsymbol{q} \in SE(3)$ is the pose of the robot frame interacting with the object and $h \in \mathbb{R}$ is the interaction value which, analogous to energy potentials, is zero when feasible and non-zero otherwise. As shown in Fig. 2, the interaction feature

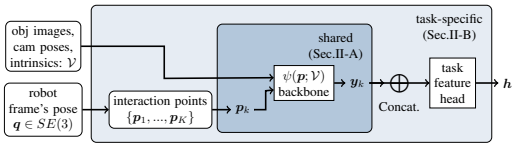



**Fig. 2:** The interaction feature prediction scheme of DVC

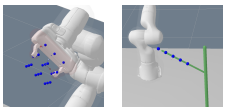


**Fig. 3:** Key interaction points on the gripper and hook

predictions are made through the feature heads based on a set of representation vectors obtained by querying the backbone at a set of key interaction points. The keypoints are rigidly attached to the robot frame to represent its pose (See Fig. 3) and are used to query the backbone, i.e., $\forall k \in \{1, ..., K\}$,

$$\boldsymbol{y}_k = \psi(\boldsymbol{p}_k; \mathcal{V}),\ \boldsymbol{p}_k = \boldsymbol{R}(\boldsymbol{q})\hat{\boldsymbol{p}}_k + \boldsymbol{t}(\boldsymbol{q}), \tag{5}$$

where $\hat{\boldsymbol{p}}_k$ is $k^{\text{th}}$ keypoint's local coordinate, and $\boldsymbol{R}(\boldsymbol{q})$ and $\boldsymbol{t}(\boldsymbol{q})$ denote the rotation matrix and the translation vector of the frame's pose $\boldsymbol{q}$, respectively. The feature head then takes as input the resulting representation vectors and predicts a feature value through a couple of fully connected layers, i.e.,

$$h = MLP(\boldsymbol{y}_1, ..., \boldsymbol{y}_K). \tag{6}$$

### C. Training Data and Loss Function

We considered three types of interaction features: an SDF feature for collision avoidance and grasping/hanging features, and generated the corresponding dataset of *posed images, SDFs, Grasping and Hanging poses*. Specifically, we took 131 mesh models of mugs from ShapeNet [2] and convex-decomposed/randomly scaled those meshes. For each mug, we generated 100 images ($128 \times 128$) with the corresponding camera poses and intrinsic matrices, 12,500 3D positions and their signed distance values following the approach of DeepSDF [8], and 1,000 feasible grasping and hanging poses of the gripper and the hook, respectively, in Bullet [3] or using kinematics checking. In the end, we have

$$\left\{\left(\mathcal{I}^{1:100}, \boldsymbol{T}^{1:100}, \boldsymbol{K}^{1:100}, \boldsymbol{p}^{1:12500}, SDF^{1:12500}, \boldsymbol{q}_{\text{grasp}}^{1:1000}, \boldsymbol{q}_{\text{hang}}^{1:1000}\right)^{(i)}\right\}_{i=1}^{131},$$

which we divided into 78 train, 25 validation and 28 test sets.

In each iteration of the network training, we first choose a minibatch of mugs from which a subset of augmented images with their camera poses and intrinsics, $\hat{\mathcal{V}} = \{(\hat{\mathcal{I}}^1, \hat{\boldsymbol{T}}^1, \hat{\boldsymbol{K}}^1), ..., (\hat{\mathcal{I}}^4, \hat{\boldsymbol{T}}^4, \hat{\boldsymbol{K}}^4)\}$, a subset of SDF data, $\left(\boldsymbol{p}^{1:300}, SDF^{1:300}\right)$, and the grasping/hanging data, $\left(\hat{\boldsymbol{q}}_{\text{task}}^{1:100}, d_{\text{task}}^{1:100}\right)$, where $\hat{\boldsymbol{q}}$ is a random pose and $d = \min_{i=1,...N_{\text{task}}} ||\hat{\boldsymbol{q}} - \boldsymbol{q}_{\text{task}}^i||_2$ is its unsinged distance in $SE(3)$ to the set of the feasible poses, are sampled. The images are encoded only once per iteration and then the SDF, grasping, hanging features are queried at the sampled points and poses. The overall loss is given as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sdf}} + \mathcal{L}_{\text{grasp}} + \mathcal{L}_{\text{hang}}$, where we used a typical $L1$ loss for SDFs, i.e. $\mathcal{L}_{\text{sdf}} = \frac{1}{N_{\text{SDF}}} \sum_{i=1}^{N_{\text{SDF}}} |\phi_{\text{sdf}}(\boldsymbol{p}^i) - SDF^i|$, and the sign-agnostic $L1$ loss in [1] for grasping and hanging, i.e., $\forall \text{task} \in \{\text{grasp, hang}\}$ $\mathcal{L}_{\text{task}} = \frac{1}{N_{\text{task}}} \sum_{i=1}^{N_{\text{task}}} \left| \left|\phi_{\text{task}}(\hat{\boldsymbol{q}}_{\text{task}}^i; \hat{\mathcal{V}})\right| - d_{\text{task}}^i \right|$. The feature heads and the backbone are trained end-to-end.

## III. Sequential Manipulation Planning with DVC

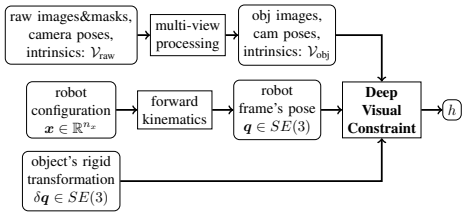



**Fig. 4:** Deep visual constraints for manipulation planning

The learned features are integrated into LGP [12] as differentiable interaction constraints as shown in Fig. 4. First, scene images are warped into the object-centric images by the multi-view processing (see the full-text for details) and the robot frame's poses are computed from a robot joint configuration via a forward kinematics engine. One core concept of manipulation planning is the rigid transformations of objects. For an object transformed by $\delta\boldsymbol{q} \in SE(3)$, we define rigid transformations of the representation function as $T(\delta\boldsymbol{q})[\psi](\cdot) = \psi\left(\boldsymbol{R}(\delta\boldsymbol{q})^T\left(\cdot - \boldsymbol{t}(\delta\boldsymbol{q})\right)\right)$ or, equivalently, of the interaction feature as: $T(\delta\boldsymbol{q})[\phi_{\text{task}}](\cdot) := \phi_{\text{task}}\left(\delta\boldsymbol{q}^{-1}\cdot\right)$. By composing the forward kinematics with the feature as

$$H_{\text{task}}(\boldsymbol{x}, \delta\boldsymbol{q}) := \left(T(\delta\boldsymbol{q})[\phi_{\text{task}}] \circ FK\right)(\boldsymbol{x}), \tag{7}$$

we have an interaction feature as a function of a robot joint configuration $\boldsymbol{x}$ and object's rigid transformation. Following the procedure of LGP, given a discrete action sequence $\text{a}_{1:K}$ and the corresponding symbolic modes $\text{s}_{1:K}$ with $\text{s}_K \in \mathcal{S}_{\text{goal}}$, we solve the geometric path problem over sequences of the robot joint configurations $\boldsymbol{x}_{1:KT}$, $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and the object's rigid transformations $\delta\boldsymbol{q}_{1:KT}$, $\delta\boldsymbol{q} \in SE(3)^m$ where their interactions are constrained by the learned DVCs.

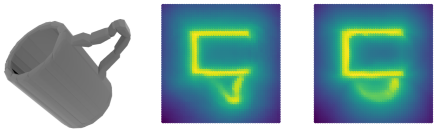
## IV. Summary of Experiments

We refer the readers to the project page for a collection of videos, the full text, as well as the available source code.

**Ablation Study:** We compared the proposed representation with three baselines: (i) Global image feature that uses a CNN to output global image features instead of using the hourglass network and extracting pixel-aligned local features, (ii) vector object representation that represents an object as a finite-dimensional vector instead of an implicit function, and (iii) SDF representation where the learned SDF feature serves as object representation. Table I shows that, while the SDF representation performs best in shape reconstruction, the task performances of PIFO are significantly better than the others. The SDF representation is especially worse in the hanging task, which implies that SDFs along the line are not sufficient

| | IoU | Grasp+c (%) | Hang+c (%) |
|---|---|---|---|
| PIFO | 0.816 / 0.656 | **88.1 / 82.5** | **94.0 / 78.9** |
| Glo. Fea. | 0.697 / 0.581 | 82.7 / 75.7 | 91.2 / 78.2 |
| Vec. Rep. | 0.036 / 0.014 | 0.5 / 0.4 | 0.0 / 0.0 |
| SDF Rep. | **0.845 / 0.667** | 67.9 / 64.3 | 3.7 / 4.3 |

**TABLE I:** Individual Feature Evaluation (Training / Test)



(a) (b) (c)

**Fig. 5:** SDFs from (b) PIFO and (c) the global image feature model

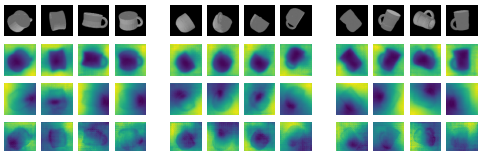
for the feature prediction and our task-guided representation simplifies the feature prediction. Notably, Fig. 5 depicts SDF values of an unseen mug (with a complex shape handle) predicted by PIFO and the global image feature model; the pixel-aligned method allowed to capture more fine-grained details whereas the global image feature model reconstructed the handle shape as being more "typical".

**Sequential Manipulation:** Once constraint models are learned, various types of long-horizon manipulation plan can be computed by combining them differently. Fig. 6 illustrates some of such scenarios: (a) Single mug hanging that simply combines two discrete actions (with the corresponding learned constraints) [(GRASP, gripper, mug), (HANG, hook, mug)], (b) the three-mug scenario having 6 discrete phases with [(GRASP, gripper, mug1), (HANG, M_hook, mug1), (GRASP, gripper, mug2), (HANG, U_hook, mug2), (GRASP, gripper, mug3), (HANG, L_hook, mug3)], and (c) the handover scenario where two arms at different heights and the target hook is placed very high, requiring two arms to coordinate a handover motion; the corresponding discrete actions are [(GRASP, R_gripper, mug), (GRASP, L_gripper, mug), (HANG, U_hook, mug)].

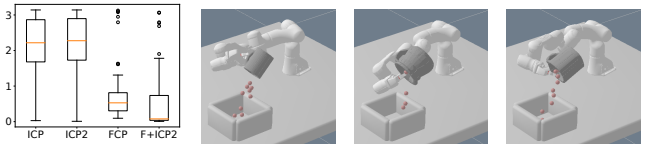
**Exploiting Learned Representation:** Figs. 7 (a)-(c) visualize three principal components of the image feature vectors. It can be observed that each component represents a certain property of the objects, such as inside vs. outside, handle vs. other parts, or above vs. below. This enables the image-based pose estimation which we call feature-based closest point (FCP) matching, i.e., the problem of finding the relative pose of a target mesh w.r.t. a model mesh, without defining any canonical coordinate of the objects. We compared this to the conventional iterative closest point algorithms on point clouds from a depth camera or from a reconstructed mesh (ICP/ICP2). As shown in Fig. 7(d), FCP outperforms ICP (due to the local optima issue) and a further improvement was observed when we used the FCP results as starting points of ICP. The PCA result also implies that the semantics of the representation are consistent across different objects, e.g. the handle parts of different mugs have similar representations. We therefore considered an image-based zero-shot imitation

(a) Single mug (b) Three-mug (c) Handover

**Fig. 6:** Sequential manipulation scenarios



(a) (b) (c)




(d) Orientation errors (e) Reference (f) Imitation 1 (g) Imitation 2

**Fig. 7:** Exploiting Learned Representation

scenario, where we manually designed the pouring motion for one mug and stored the images of pre- and post-pouring postures of the mug, $\mathcal{V}_{\text{pre}} = (\mathcal{I}_{\text{pre}}, \boldsymbol{T}_{\text{pre}}, \boldsymbol{K}_{\text{pre}})$ and $\mathcal{V}_{\text{post}} = (\mathcal{I}_{\text{post}}, \boldsymbol{T}_{\text{post}}, \boldsymbol{K}_{\text{post}})$, respectively. For a new mug, we solved LGP with [(GRASP, gripper, mug), (POSEFCP, $\mathcal{V}_{\text{pre}}$, mug), (POSEFCP, $\mathcal{V}_{\text{post}}$, mug)], where (POSEFCP, $\cdot$, $\cdot$) imposes the FCP constraint at each motion phase. As depicted in Figs. 7 (e)-(g), the learned representation enables transferring motions across different objects *only* using the posed images.

**Real Robot Demo:** To successfully apply the learned DVCs to the real robot by closing the sim-to-real gap, we had to extend training to a larger dataset. Specifically, we randomized the material of mugs to get more diverse appearances by adjusting metalness and roughness in PyRender [6]. More extensive data augmentations, such as ColorJitter or Blur, were also applied during training. At test time, we attached RealSense D435 on one of the grippers and took 8 color images from some predefined poses.

## V. DISCUSSIONS

The idea of DVCs is not limited to color images as input. Depending on the setting, e.g. whether reliable depth sensing is available, point clouds also can be considered (as in [11]) using a PointNet [9] encoder. Incorporating non-visual, like tactile, input would be another exciting direction to explore. Fig. 7 implies considering more diverse tasks and objects in our multi-task learning would lead to more generalized representations as well as synergies between individual feature learning; all those task features don't necessarily model physical interaction feasibility for planning; e.g., they can also serve as a value or energy function of a direct control policy and be trained via imitation or reinforcement learning [5, 4].

## ACKNOWLEDGMENTS

This research has been supported by the German Research Foundation (DFG) under Germany's Excellence Strategy – EXC 2002/1–390523135 "Science of Intelligence".

## REFERENCES

[1] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[3] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016–2021.

[4] Danny Driess*, Jung-Su Ha*, Russ Tedrake, and Marc Toussaint. Learning geometric reasoning and control for long-horizon tasks from visual input. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021.

[5] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*. PMLR, 2021.

[6] Matthew Matl. Pyrender. https://github.com/mmatl/pyrender, 2019.

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[8] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[9] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[11] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. *arXiv preprint arXiv:2112.05124*, 2021.

[12] Marc Toussaint, Kelsey Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems*, 2018.