IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

# MIND - Multi-Feature Implicit Neural Descriptors for Robotic Surface Processing of 3D Objects With Variations in Geometry

Anish Pratheepkumar, Christian Hartl-Nesic, *Member, IEEE*, Markus Ikeda, Fabian Widmoser, Andreas Pichler, and Markus Vincze

***Abstract*—The recent shift from mass production to mass personalization leads to a production environment in which workpieces have a high degree of geometric variations. The robotic process automation in such high-mix low-volume environments poses significant challenges since predetermined robot programs are not viable anymore. In this letter, we consider the automation of surface processing for category-level objects with significant variations in geometry by operating on point clouds without relying on CAD models. To achieve this, we present a novel *multi-feature implicit neural descriptor* (MIND) representation which leverages dense correspondence to generalize across diverse objects, enabling a one-shot transfer of process trajectories and associated process knowledge. The quantitative and qualitative evaluation shows that MIND outperforms other state-of-the-art dense correspondence approaches. A real-world application case study of robotic surface processing on geometry-varying basin molds validates the efficacy of the proposed approach.**

***Index Terms*—Computer vision for automation, industrial robots, representation learning.**

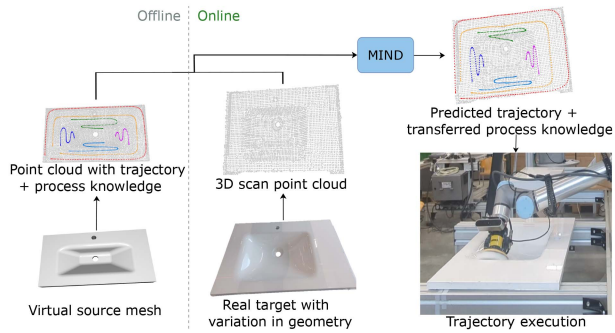


Fig. 1. Illustration of the proposed approach to address generalization in robotic surface processing by using the MIND dense correspondences for transfer of trajectory and associated process knowledge such as end-effector orientation, speed, and force across basins with variation in geometry.

## I. Introduction

ROBOTIC surface processing is of rising demand in industrial manufacturing [1], [2], driven by the unhealthy work conditions for humans, particularly in the sectors of, e.g., casted and molded products, vehicles, and wooden products. Example applications are polishing, painting, and grinding. Moreover, sectors such as household, and health care would benefit from automated cleaning, and disinfection [3]. However, the process complexity introduced by the 3D surface variations poses a significant challenge to robotic automation, because of which these processes are still largely performed by humans [1]. Practically, variations in shape, size, proportions, and spatial structure have become more prominent in recent years due to high-mix low-volume scenarios. For example, the source and target basins shown in Fig. 1 have significant variations in geometry, despite being category-level objects. Such scenarios demand frequent and tedious reprogramming of the robot. In general, surface processing with robots has been extensively studied [1], [4], [5]. However, existing solutions do not generalize well when there is high degree of variation in geometry [2].

To address this gap, we propose to use dense correspondence as a key method such that the surface processing trajectories and associated process knowledge can be transferred across diverse category-level objects. Additionally, we consider that the object meshes are unavailable and robotic processes are instead performed using real-time sensor data, a situation commonly observed in small and medium enterprises [6]. From this perspective, point clouds are the best-suited representation as real-time sensors do not directly generate meshes or voxels, which both need computationally expensive processing [7]. Even after processing, the presence of sensor noise significantly degrades the model performance. Also, directly using point cloud representation makes it difficult to generalize since scanned point clouds are noisy and have discontinuities. Hence, we propose to use an implicit descriptor representation which is continuous in 3D space and works at arbitrary resolutions [7].

Recently, approaches based on neural implicit descriptors have shown potential for generalizable robotic processes [8], [9],

Received 7 June 2025; accepted 5 October 2025. Date of publication 24 October 2025; date of current version 3 November 2025. This article was recommended for publication by Associate Editor Zhongkui Wang and Editor Hyungpil Moon upon evaluation of the reviewers' comments. This work was supported by the Austrian Institute of Technology (AIT), through the Lighthouse Project. *(Corresponding author: Anish Pratheepkumar.)*

Anish Pratheepkumar is with the PROFACTOR GmbH, 4407 Steyr-Gleink, Austria, and also with the Automation and Control Institute, 1040 TU Wien, Austria (e-mail: aprath@profactor.at).

Christian Hartl-Nesic and Markus Vincze are with the Automation and Control Institute, 1040 TU Wien, Austria (e-mail: hartl@acin.tuwien.ac.at; vincze@acin.tuwien.ac.at).

Markus Ikeda, Fabian Widmoser, and Andreas Pichler are with the PROFACTOR GmbH, 4407 Steyr-Gleink, Austria (e-mail: mikeda@profactor.at; fwidmo@profactor.at; apichl@profactor.at).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3625495, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3625495

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

[10], [11]. The descriptors are trained and optimized to provide dense 3D correspondences for facilitating the transfer of process knowledge across diverse objects. While this gives good results for point-based operations like object manipulation [8], [10], it does not directly transfer to surface processing. Towards this goal, a neural region descriptor field (NRDF) [2] representation was introduced recently. However, the approach is limited to minor variations in the geometry, and needs a CAD/CAM module for trajectory generation [2].

To overcome these problems, we propose to embed the local and global geometric features of a given 3D object category into one description. These *multi-feature implicit neural descriptors* (MIND) provide high-quality dense correspondences that enable direct prediction of processing trajectories and facilitate associated knowledge transfer, also for objects with a high degree of variation in geometry. Additionally, MIND is capable of operating on point clouds without relying on CAD models. An illustration of the proposed approach is shown in Fig. 1. The underlying dense correspondence formulation ensures that any trajectory and its associated knowledge are transferred across category-level objects, i.e., no retraining is needed when the source trajectory or process knowledge are changed.

The main contributions of our work are:

1) We propose a novel neural implicit representation called MIND, generated by systematically infusing fully convolutional features, point normals, signed distance functions (SDF), and geometric feature-rich sparse keypoint correspondences into the network.
2) We address the challenge of generalization of robotic 3D surface processing in a high-mix low-volume setting, by utilizing the MIND dense correspondences for direct transfer of trajectories and process knowledge between diverse category-level objects.
3) Extensive evaluations and real-world robotic surface processing experiments on geometry-varying basin molds demonstrate the effectiveness of our approach.

## II. Related Work

In this work, we utilize neural implicit representation learning to obtain generalizable robotic surface processing of 3D objects with different geometry. The relevant works on generalizable robotic processes, and generalizable robotic 3D surface processing are discussed in the following.

### A. Generalizable Robotic Processes Using Neural Implicit Representations

The recent advances in neural implicit representation learning [7], [12] have made noticeable progress in generalizing robotic processes [2], [13]. These representations are generated by optimizing a neural network to encode the distribution of the boundary of a geometry in 3D space to a unified and continuous descriptor space. The specialty of such descriptors is that they are unique, and enable descriptor-level dense 3D correspondences across category-level objects, facilitating the transfer of robotic processes between these objects.

In [8] the descriptor space is optimized by training on the object boundary occupancy, which specifies whether a given 3D point is occupied or not on the surface and hence encoding the boundary distribution of the object. Chu et al. [9] also learn the occupancy but emphasizes the local geometrical features. Huang et al. show improved performance by learning the space coverage function (SCF) [10]. Recently, Cai et al. [11] proposed a combined feature-learning method with occupancy, extended SCF, and SDF for one-shot manipulation tasks. These prior works focus on corresponding points for manipulation knowledge transfer, in contrast, a recent work [2] explores the applicability of neural descriptors in one-shot robotic surface process transfer by learning the occupancy, and explicitly optimizing descriptor fields to be similar for corresponding regions across objects.

### B. Generalizable Robotic 3D Surface Processing

Several prior works have studied robotic 3D surface processing, addressing various aspects such as coverage planning [1], specific materials [4], and applications [5]. Still, research targeting generalization in robotic surface processing remains limited. Early works consider model-based methods, e.g., [4] targets polishing of wooden workpieces, and [5] addresses uniform coverage for painting. However, the models need to be adapted when the geometry and structure of the object varies. In [6] trajectory is estimated on filtered point clouds, but it is a simple sequence of all points in a manually selected region lacking adaptability. A probabilistic approach utilizing local surface features and human demonstrations for robotic edge cleaning application was proposed in [14]. The work [15] adopt impedance control for learning from expert demonstrations. However, a change in geometry and structure would require new demonstrations. A coverage-planning approach for polishing of freeform surfaces was introduced recently in [1]. Nevertheless, presence of discontinuities or holes result in inaccuracies [1], limiting generalization.

The recent work [2] introduces generalizable robotic surface processing utilizing neural descriptors. However, the approach requires additional CAD/CAM software for trajectory transfer, and is limited to minor variation in geometry. Möhl et al. [16] proposes an approach based on point cloud morphing for surface-finishing applications. This approach is assisted with keypoint correspondences, but needs online computation and is limited to objects with minor size variations. Inspired from point cloud morphing [16], we adapt the idea of utilizing keypoints by injecting this knowledge within the trained model, and thus prevent computationally expensive and time-consuming online optimization. Compared to the existing approaches [1], [2], [8], our work targets generalization for high degree of variations in geometry, and we propose correspondence-assisted trajectory transfer towards eliminating the need of CAD/CAM software.

## III. Multi-Feature Implicit Neural Descriptors

Given an object category, we learn a unified descriptor space such that the descriptors of corresponding locations on various category-level objects are as similar as possible, i.e., they lie in the same location in the descriptor space. Towards this goal, we propose the novel MIND. In the following, we first introduce the MIND network architecture and then explain the proposed descriptor optimization methodology. Finally, we explain the proposed approach of utilizing descriptor-assisted dense correspondence for trajectory and process knowledge transfer in surface processing.

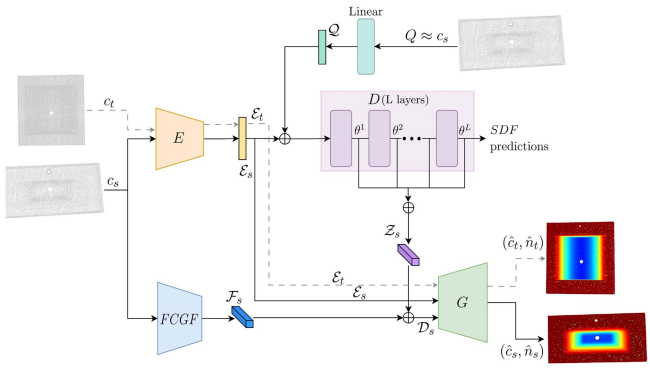


Fig. 2. MIND network architecture: Given a source object point cloud $c_s$, the encoder $E$ extracts a global object embedding $\mathcal{E}_s$. The decoder $D$ is trained to learn the SDF feature for any 3D query point $Q$ in space. Then the concatenated intermediate layer activations of $D$ give the latent space embedding (LSE) feature $\mathcal{Z}_s$. At the same time, an FCGF network extracts geometric features $\mathcal{F}_s$ from $c_s$. Then the MIND descriptors $\mathcal{D}_s$ are given by the concatenation of $\mathcal{Z}_s$ and $\mathcal{F}_s$. Given the object embedding $\mathcal{E}_s$, the descriptor decoder function $G$ is trained to map the descriptors $\mathcal{D}_s$ to the corresponding point-normal pairs $(\hat{c}_s, \hat{n}_s)$ for an object-level descriptor optimization. By swapping the object embedding with that of a target object $\mathcal{E}_t$, the descriptor decoder $G$ is trained to predict the point-normal pairs of the target object $(\hat{c}_t, \hat{n}_t)$ for a category-level descriptor optimization.

TABLE I
PARAMETERS FOR MIND

| Description | Value |
|---|---|
| Encoder $E$ input feature $I \times 3$ | $8192 \times 3$ |
| Encoder $E$ output feature $\lvert\mathcal{E}\rvert$ | 384 |
| Query point embedding $\nu$ | 32 |
| Decoder $D$ input feature | 416 |
| Decoder $D$ output feature $\lvert SDF\rvert$ | 1 |
| FCGF feature $\lvert\mathcal{F}\rvert$ | 32 |
| LSE feature $\lvert\mathcal{Z}\rvert$ | 1953 |
| MIND feature $d$ | 1985 |
| Descriptor decoder $G$ input feature | 2369 |
| Descriptor decoder $G$ output feature | 6 |
| Random sampled query points $Q$ per iteration $J$ | 8192 |
| Exponential weighting constant $\alpha$ | 1 |

### A. Network Architecture

The overall MIND network architecture is depicted in Fig. 2 and the utilized parameters are summarized in Table I. We use a PointNet-based encoder $E$ [17], while the SDF decoder $D$ and the descriptor decoder $G$ are designed as multilayer perceptrons. Additionally, a fully convolutional geometric features (FCGF) network [18] is utilized to extract the geometric features from the input point cloud.

### B. Approach and Descriptor Optimization

Below, we explain the three stages of the network optimization. The descriptor optimization builds on the concept of self-object and cross-object reconstructions introduced in [2]. However, our approach differs significantly as we introduce a multi-feature approach to carefully optimize the network by focusing on implicitly embedding the local and global geometric features. Furthermore, we use SDF rather than occupancy, as it offers a richer geometric abstraction with SDF gradients directly indicating local surface orientations.

*a) Stage 1: Encoding Object Geometry:* As input representation, we utilize a voxel-downsampled point cloud $c_o \in \mathbb{R}^{I\times 3}$ of the object $o$ with a set of $I$ points in 3D space $c_o = \{\mathbf{c}_{o,i}\}_{i=1}^{I}$. To implicitly encode the geometric boundary, we propose to utilize the SDF feature [12]: for any 3D point inside the boundary $SDF < 0$, along the boundary $SDF = 0$, and outside the boundary $SDF > 0$. Given a point cloud $c_o$ and random set of $J$ query points $Q = \{\mathbf{q}_j\}_{j=1}^{J}$ in 3D space, an autoencoder is trained to predict associated SDF values. Mathematically, the encoder $E$, and decoder $D$ are defined as

$$E\left(c_o\right) = \mathcal{E}_o, \tag{1}$$

$$D\left(\mathcal{E}_o, \mathbf{q}_j\right) \approx SDF\left(\mathbf{q}_j\right),\ j = 1, \ldots, J \tag{2}$$

where the encoder $E$ extracts a global geometric embedding $\mathcal{E}_o$. By conditioning the decoder $D$ on the embedding $\mathcal{E}_o$, it learns to predict the SDF. Therefore, a zero level set of the autoencoder gives the boundary points of the geometry. Here a simple linear layer maps the query points $\mathbf{q}_j \in \mathbb{R}^3$ to a higher dimension $\mathcal{Q}_j \in \mathbb{R}^{\nu}$ before passing it to the decoder $D$. To train the decoder $D$, we adopt the loss function $\mathcal{L}_{\text{SDF}}$

$$\mathcal{L}_{\text{SDF}} = \frac{1}{J}\sum_{j=1}^{J}(SDF\left(\mathbf{q}_j\right) - D\left(\mathcal{E}_o, \mathbf{q}_j\right))^2. \tag{3}$$

*b) Stage 2: Descriptor Optimization on Object Level:* After capturing the geometric boundary as zero level sets of an SDF, we extract a latent space embedding (LSE) $\mathcal{Z}_{o,j}$ for any given query point $\mathbf{q}_j,\ j = 1, \ldots, J$, with respect to the object $o$ by concatenating ($\bigoplus$) the activations $\theta^l$ of each intermediate layer $l$ of the decoder $D$ with $L$ layers. Typically, this LSE feature is taken as descriptor, see the recent works [2], [10]. In this work, we instead propose a novel enhanced descriptor space by concatenating the geometric features $\mathcal{F}_o$ extracted using FCGF with the LSE features $\mathcal{Z}_o$. Additionally, for estimating the descriptor $\mathcal{D}_o$ we limit the query points $Q$ to the geometric boundary; the query points then essentially reduce to the object point cloud $c_o$. Then, for each point $\mathbf{c}_{o,i},\ i = 1, \ldots, I$, in the point cloud, we compute the LSE $\mathcal{Z}_{o,i}$ and geometric features $\mathcal{F}_{o,i}$. These features are then concatenated to give its $d$-dimensional descriptor $\mathcal{D}_{o,i} \in \mathbb{R}^d$, reading as

$$\mathcal{Z}_{o,i} = \bigoplus_{l=1}^{L} \theta^l(\mathcal{E}_o, \mathbf{c}_{o,i}), \tag{4}$$

$$\mathcal{F}_{o,i} = FCGF(\mathbf{c}_{o,i}|c_o), \tag{5}$$

$$\mathcal{D}_{o,i} = f(\mathbf{c}_{o,i}, \mathcal{E}_o|c_o) = \mathcal{Z}_{o,i} \bigoplus \mathcal{F}_{o,i}\ . \tag{6}$$

Hence given the point cloud $c_o$, each descriptor $\mathcal{D}_{o,i}$ is a function $f$ of a point $\mathbf{c}_{o,i}$ and the encoder embedding $\mathcal{E}_o$. The encoder embedding $\mathcal{E}_o$ represents the global feature of the geometry as in (1). Furthermore, the LSE $\mathcal{Z}_{o,i}$ and the geometric features $\mathcal{F}_{o,i}$ are computed locally with LSE focusing on the SDF feature, and geometric features focusing on the local neighborhood. Hence, within the descriptor we implicitly embed the local geometric features with respect to the global geometric feature of the object category.

To optimize the descriptor space with respect to the geometry, we propose a descriptor decoder $G$ which can reconstruct the object point cloud $c_o$ along with its point normal vectors $n_o = \{\mathbf{n}_{o,i}\}_{i=1}^{I}$, where each normal is a unit vector with $\|\mathbf{n}_{o,i}\| = 1$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

defining the local surface orientation at $\mathbf{c}_{o,i}$. The descriptor decoder function $G$ is designed to map the descriptors $\mathcal{D}_{o,i}$ along with the encoder embedding $\mathcal{E}_o$ to the corresponding point-normal pairs $(\mathbf{c}_{o,i}, \mathbf{n}_{o,i})$ as

$$G(\mathcal{E}_o, \mathcal{D}_{o,i}) = (\hat{\mathbf{c}}_{o,i}, \hat{\mathbf{n}}_{o,i}) \approx (\mathbf{c}_{o,i}, \mathbf{n}_{o,i}). \tag{7}$$

Therefore, we propose to optimize the descriptor space to correspond to the 3D space with the 3D point prediction $\hat{\mathbf{c}}_{o,i}$, and simultaneously inject the knowledge of local surface orientation with the normal vector prediction $\hat{\mathbf{n}}_{o,i}$ explicitly into the descriptor. This optimization is governed by the mean squared error (MSE) loss function $\mathcal{L}_{\text{MSE}}$ ensuring accurate 3D point reconstruction $\hat{\mathbf{c}}_{o,i} \approx \mathbf{c}_{o,i}$

$$\mathcal{L}_{\text{MSE}}\left(c_o, \hat{c}_o\right) = \frac{1}{I}\sum_{i=1}^{I}(\mathbf{c}_{o,i} - \hat{\mathbf{c}}_{o,i})^2, \tag{8}$$

and the cosine similarity loss function $\mathcal{L}_{\text{CS}}$ ensuring accurate point normal prediction $\hat{\mathbf{n}}_{o,i} \approx \mathbf{n}_{o,i}$

$$\mathcal{L}_{\text{CS}}\left(n_o, \hat{n}_o\right) = \frac{1}{I}\sum_{i=1}^{I}\left(1 - \frac{\mathbf{n}_{o,i}\cdot\hat{\mathbf{n}}_{o,i}}{\|\mathbf{n}_{o,i}\|_2\,\|\hat{\mathbf{n}}_{o,i}\|_2}\right), \tag{9}$$

where $\|\cdot\|_2$ denotes the L2 norm.

*c) Stage 3: Descriptor Optimization on Category Level:* With the object boundary encoding, followed by the object-level descriptor optimization we have controlled the descriptor space to correspond well with the 3D space. However, our goal is to optimize the descriptor space to correspond uniquely across many geometry-varying objects belonging to the same category. We propose a category-level cross reconstruction of the object point-normal pairs from descriptors by conditioning on the object embedding $\mathcal{E}_o$. Specifically, we design the optimization such that given a source object descriptor $\mathcal{D}_{s,i}$, the function $G$ should give the corresponding source point-normal pairs $(\mathbf{c}_{s,i}, \mathbf{n}_{s,i})$, if the input embedding is from the same source object $\mathcal{E}_s$ as seen in (7). At the same time, if the input embedding is from a new target object $\mathcal{E}_t$, then the function $G$ should give the corresponding target point-normal pairs $(\mathbf{c}_{t,i}, \mathbf{n}_{t,i})$, reading as

$$G(\mathcal{E}, \mathcal{D}_{s,i}) = \begin{cases} (\mathbf{c}_{s,i}, \mathbf{n}_{s,i}), & \text{if } \mathcal{E} = \mathcal{E}_s\ , \\ (\mathbf{c}_{t,i}, \mathbf{n}_{t,i}), & \text{if } \mathcal{E} = \mathcal{E}_t\ . \end{cases} \tag{10}$$

In the following, we carefully design a set of loss functions to enforce accurate cross reconstruction $G(\mathcal{E}_t, \mathcal{D}_s) \approx (c_t, n_t)$. To ensure $\hat{c}_t \approx c_t$, we could ideally compare the similarity between the two point clouds $\hat{c}_t$ and $c_t$. However, since the descriptors are from the source object, the initial prediction $\hat{c}_t$ would have an affinity to resemble $c_s$ resulting from the descriptor optimization in Stage 2. Subsequently, due to variations in geometry of the objects being compared, there is a noticeable difference in the local density distribution of the point clouds $c_s$ and $c_t$. Let us consider the simple example of a plane with short wings as source object, and a target object with long wings. Now, if both object point clouds have the same number of points, then the short wings will be denser in contrast to the long wings, which will appear sparse. Additionally, for thinner parts such as the wings of a plane, the nearest-neighbor estimation in the point set similarity computations tend to be erroneous due to the close proximity of lower and upper surface of the wings. Hence, geometry awareness and density awareness need to be simultaneously considered to obtain well-posed descriptors. To this end, we propose a new loss function called geometry-aware density-aware chamfer distance $\mathcal{L}_{\text{GDCD}}$, which is an adapted version of [19] defined as

$$\mathcal{L}_{\text{GDCD}}\left(h_t, \hat{h}_t\right) = \frac{1}{2}\left(\frac{1}{I_t}\sum_{i=1}^{I_t}\left(1 - \frac{1}{\hat{\bar{I}}_t}\mathrm{e}^{-\alpha\|h_{t,i} - \hat{\bar{h}}_{t,i}\|_2^2}\right) + \frac{1}{\hat{I}_t}\sum_{i=1}^{\hat{I}_t}\left(1 - \frac{1}{\bar{I}_t}\mathrm{e}^{-\alpha\|\hat{h}_{t,i} - \bar{h}_{t,i}\|_2^2}\right)\right), \tag{11}$$

where $h_t$ are the concatenated point-normal pairs $c_t \bigoplus n_t$, with their corresponding predictions denoted as $\hat{h}_t$, $\hat{\bar{h}}_{t,i}$ indicates the nearest neighbor of $h_{t,i}$ in $\hat{h}_t$, $\alpha$ is a constant in the exponential weighting to control sensitivity of the distances between the neighbors. Furthermore, $I_t = |h_t|$, $\hat{I}_t = |\hat{h}_t|$, $\hat{\bar{I}}_t$ indicate the number of points in $h_t$ that consider $\hat{\bar{h}}_{t,i}$ as its neighbor, and $\bar{I}_t$ is the number of points in $\hat{h}_t$ that consider $\bar{h}_{t,i}$ as its neighbor. Note that a single point $\hat{\bar{h}}_{t,i}$ may be shared by $\hat{\bar{I}}_t$ points in the point set being compared. Normalizing the effect with $1/\hat{\bar{I}}_t$ in (11) takes into account the differences in local density distributions [19]. The distances from $h_t$ to $\hat{h}_t$ are computed in the first part of (11), and its reverse in the second part. Moreover, by incorporating the local surface orientation, i.e., normal vectors, we integrate the geometry awareness into $\mathcal{L}_{\text{GDCD}}$.

The consistency of the predicted normals is ensured by a normal consistency distance loss function $\mathcal{L}_{\text{NCD}}$ comparing the bidirectional cosine similarity between the neighboring normals of target $n_t$ and prediction $\hat{n}_t$, given by

$$\mathcal{L}_{\text{NCD}}\left(n_t, \hat{n}_t\right) = \frac{1}{I_t}\sum_{i=1}^{I_t}\left(1 - \frac{\mathbf{n}_{t,i}\cdot\hat{\bar{\mathbf{n}}}_{t,i}}{\|\mathbf{n}_{t,i}\|_2\,\|\hat{\bar{\mathbf{n}}}_{t,i}\|_2}\right) + \frac{1}{\hat{I}_t}\sum_{i=1}^{\hat{I}_t}\left(1 - \frac{\hat{\mathbf{n}}_{t,i}\cdot\bar{\mathbf{n}}_{t,i}}{\|\hat{\mathbf{n}}_{t,i}\|_2\,\|\bar{\mathbf{n}}_{t,i}\|_2}\right), \tag{12}$$

where $\hat{\bar{\mathbf{n}}}_{t,i}$ indicates nearest-neighbor normal of $\mathbf{n}_{t,i}$ in $\hat{n}_t$.

The efficiency of $\mathcal{L}_{\text{GDCD}}$ and $\mathcal{L}_{\text{NCD}}$ degrades with respect to variations in geometry, especially with respect to size, since the neighborhood computation gets deteriorated as the actual corresponding points of the objects being compared are not spatially close anymore. To tackle this challenge, we introduce predefined sparse corresponding keypoints across category-level objects as a constraint during training to steer the cross reconstruction. We propose a corresponding-keypoints loss function $\mathcal{L}_{\text{CK}}$ such that the descriptors $\mathcal{D}_{s,u}$ of the predefined set of $U$ keypoints $k_s = \{\mathbf{k}_{s,u}\}_{u=1}^{U} \subseteq c_s$ in the source $c_s$ are responsible for reconstructing the corresponding keypoints $k_t$ on the target $c_t$, introduced as

$$\mathcal{D}_{s,u} = f(\mathbf{k}_{s,u}, \mathcal{E}_s | c_s), \tag{13}$$

$$\hat{\mathbf{k}}_{t,u} = G(\mathcal{E}_t, \mathcal{D}_{s,u}), \tag{14}$$

$$\mathcal{L}_{\text{CK}}\left(k_t, \hat{k}_t\right) = \frac{1}{U}\sum_{u=1}^{U}\left(\mathbf{k}_{t,u} - \hat{\mathbf{k}}_{t,u}\right)^2. \tag{15}$$

The keypoints are sparse, i.e., $U \ll |C|$, and chosen to be at local geometric feature-rich regions such as corners, edges, and intersections of constituent parts. This careful consideration

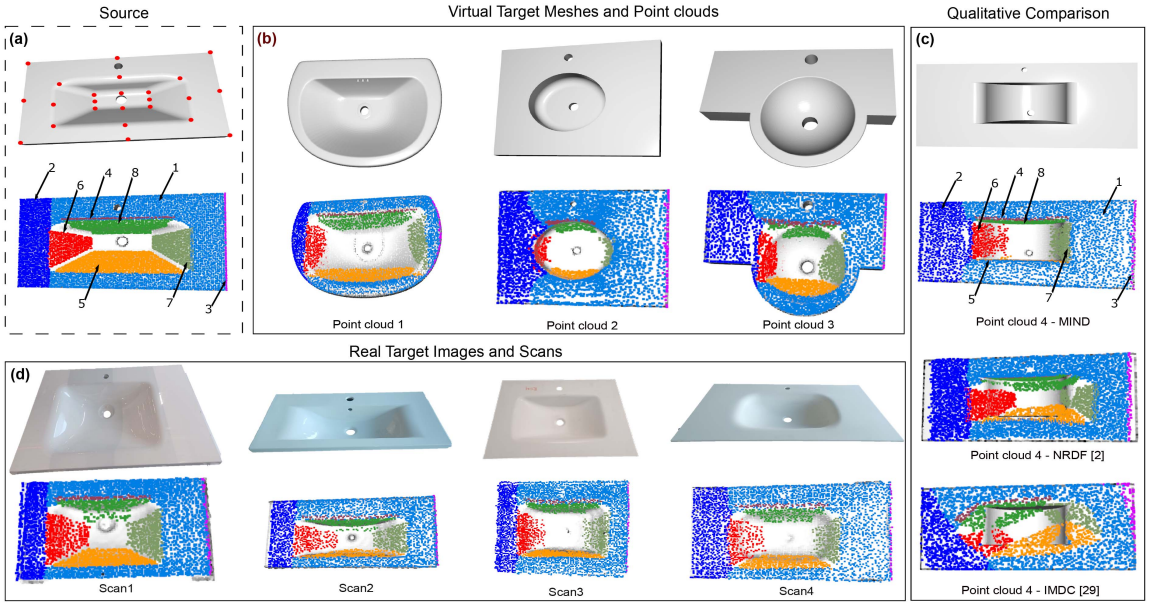


Fig. 3. Category-level dense correspondence capacity. (a) The virtual source basin mesh along with the keypoints, and the eight source ROIs on the point cloud. (b) Diverse virtual target basin meshes and the MIND-predicted corresponding ROIs on the point cloud. (c) Qualitative comparison between MIND, and the NRDF [2] and IMDC [24] baselines for dense ROI correspondence from source to a virtual target. (d) Diverse real target basin images and the MIND-estimated corresponding ROIs on the 3D scan point clouds.

ensures that even with sparsity, the keypoints capture a skeletal topology of the geometry. Hence, minimizing (15), which in turn enforces (14), leads to a constrained optimization of the descriptor space steered by the corresponding keypoints $k_s$ and $k_t$ across diverse geometry-varying objects in the same category.

### C. Trajectory and Knowledge Transfer

This section details how the MIND dense correspondences facilitate process trajectory and associated process knowledge transfer. We consider a high-mix low-volume production scenario where a set of processing trajectories $T_s$ and its associated process knowledge such as end-effector orientation, speed, force, or tool diameter, has to be transferred from a source object to a target object with variation in geometry within the same object category. Let the list of processing trajectories $T_s$ for the source object be defined as

$$T_s = \{\tau_{s,r}\}_{r=1}^{R},\ \tau_{s,r} = \{\boldsymbol{\tau}_{s,r,\lambda} \mid 1 \le r \le R, 1 \le \lambda \le \Lambda\}, \tag{16}$$

where $\boldsymbol{\tau}_{s,r,\lambda}$ is a way point in a specific trajectory $\tau_{s,r}$. Note that the number of waypoints $\Lambda$ may vary for each trajectory $\tau_{s,r}$ depending on the size of the region on which the process is executed. Then we utilize the MIND descriptors $\mathcal{D}_{s,r,\lambda},\ \lambda = 1, \ldots, \Lambda$, to transform the trajectory waypoint $\boldsymbol{\tau}_{s,r,\lambda}$ from source to target via dense correspondence by using (10), mathematically

$$\mathcal{D}_{s,r,\lambda} = f(\boldsymbol{\tau}_{s,r,\lambda}, \mathcal{E}_s | c_s)\,, \tag{17}$$

$$G(\mathcal{E}_t, \mathcal{D}_{s,r,\lambda}) = (\hat{\boldsymbol{\tau}}_{t,r,\lambda}, \hat{\mathbf{n}}_{t,r,\lambda})\,. \tag{18}$$

By applying (17) and (18) on all waypoints of a specific source trajectory $\tau_{s,r}$, we estimate the corresponding target trajectory $\hat{\tau}_{t,r} \approx \tau_{t,r}$. The process knowledge defined once for each source trajectory in $T_s$ is then automatically transferred to the target by using the MIND correspondences as a medium. Finally, the estimated trajectory is subjected to Gaussian smoothing and the smoothed trajectory-knowledge pairs are then executed by the robot on the target.

## IV. Evaluation Experiments

In this section, we first present our implementation details. Then, the following experiments evaluate the effectiveness of MIND in comparison with baselines for the dense correspondence accuracy and the execution time. Additionally, we perform a feature interaction and ablation analysis. Finally, we present a proof-of-concept application case study of real-world robotic surface polishing on geometry-varying basins.

### A. Implementation Details

*1) Training and Evaluation Dataset:* To train MIND we use the ShapeNet [20] dataset. The SDF and point normal ground truths are generated using the Open3D library [21], while the keypoint ground truths are obtained from the KeypointNet dataset [22]. We consider chair, car, and plane objects, with 10, 22, and 14 keypoints, respectively. For each category we use about 500 objects. Note that some ShapeNet objects have internal details, e.g., the engine or the seats of a car. We either exclude or manually remove these details since our focus is to learn the surface boundary. Additionally, we include a custom-generated basin category with 200 objects and manually annotated 48 feature-rich sparse keypoint correspondences. The locations of 24 keypoints are shown on the basin mesh in Fig. 3(a), and the remaining are located in the symmetrically opposite side behind the basin. Then for the dense correspondence evaluation we utilize an extended version of the benchmark for surface region correspondence (BSRC) dataset [2]. We extend the BSRC data significantly to 190 pairwise correspondence evaluations in each object category with 20 objects per category.

TABLE II
COMPARATIVE EVALUATION RESULTS FOR DENSE-CORRESPONDENCE CAPACITY MEASURED WITH CHAMFER DISTANCE, THE AVERAGE EXECUTION TIME IN SECONDS, AND THE ABLATION ANALYSIS OF MIND

| | | Chamfer Distance | | | | | Execution Time |
|---|---|---|---|---|---|---|---|
| Model | Modality | Basin | Car | Plane | Chair | Average | Average |
| DM [23] | mesh | 0.288 | 0.653 | 0.249 | 0.521 | 0.428 | 171.102 |
| SFM [25] | mesh | 0.448 | 0.493 | 0.341 | 0.546 | 0.457 | 173.360 |
| NIFT [10] | point | 0.390 | 0.279 | 0.159 | 0.492 | 0.330 | 46.178 |
| NDF [8] | point | 0.451 | 0.326 | 0.214 | 0.446 | 0.359 | 46.622 |
| L-NDF [9] | point | 0.180 | 0.137 | 0.094 | 0.248 | 0.165 | 0.295 |
| IMDC [24] | point | 0.083 | 0.033 | 0.037 | 0.063 | 0.054 | **0.150** |
| NRDF [2] | point | 0.059 | 0.025 | 0.026 | 0.046 | 0.039 | 0.215 |
| MIND | point | **0.013** | **0.022** | **0.023** | **0.043** | **0.025** | 0.225 |
| $MIND_{ablate\_\mathcal{F}}$ | point | 0.014 | 0.023 | 0.024 | 0.045 | 0.027 | 0.214 |
| $MIND_{ablate\_k}$ | point | 0.029 | 0.073 | 0.024 | 0.054 | 0.045 | 0.225 |
| $MIND_{ablate\_den}$ | point | 0.028 | 0.039 | 0.035 | 0.096 | 0.050 | 0.225 |
| $MIND_{ablate\_geom}$ | point | 0.015 | 0.027 | 0.025 | 0.052 | 0.030 | 0.225 |

*2) Training Details:* MIND is trained in three stages as mentioned in Section III-B, such that Stage 1 optimizes the object encoding with $\mathcal{L}_{\text{SDF}}$ loss function, Stage 2 optimizes the descriptor on object level with $\mathcal{L}_{\text{SDF}}$, $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{CS}}$ losses with the weights 10, 10 and 0.1, respectively, and Stage 3 optimizes the descriptor space across multiple objects with a combination of $\mathcal{L}_{\text{SDF}}$, $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{CS}}$, $\mathcal{L}_{\text{GDCD}}$, $\mathcal{L}_{\text{NCD}}$, and $\mathcal{L}_{\text{KS}}$ losses with the weights 1, 10, 0.1, 0.5, 0.1, and 10, respectively. Each stage is trained for 500 iterations progressively, and the model parameters are optimized by minimizing the loss functions corresponding to each stage using Adam optimizer with a learning rate of $10^{-4}$.

*3) Baselines:* We compare our approach with the following state-of-the-art baselines that use dense correspondence for robotic processes: NDF [8], L-NDF [9], NIFT [10], NRDF [2] and DM [23]. Additionally, two generic dense correspondence baselines IMDC [24] and SFM [25] are also considered. Among these, DM and SFM use mesh representation and the remaining ones use point cloud representation. All point-based baselines are trained on the same ShapeNet object categories. For DM we utilize the pretrained foundation model designed for zero-shot inference [23], while SFM directly provides dense correspondence estimates for arbitrary input mesh pairs by an optimization-based analytical framework [25]. Note that, for evaluating the mesh-based baselines, our point cloud inputs are converted to meshes using the reconstruction approach in the robot deployment pipeline of DM [23].

### B. Dense-Correspondence Capacity

We evaluate the dense correspondence capacity of MIND by testing on the extended version of BSRC benchmark dataset to recover eight random regions of interest (ROI) across different objects in each category. The different regions considered in the basin category are illustrated on the source basin given in Fig. 3(a), and the regions for the remaining objects are the same as defined in the benchmark [2]. For each object category all possible pairs of objects are selected from the dataset and we estimate corresponding ROIs from one to the other, in both directions. Then the chamfer distance (CD) error metric [19] between the predicted and ground truth ROIs are computed to evaluate the accuracy of trained models. The performance of MIND in comparison to the baselines is presented in Table II. MIND achieves the lowest average error value of 0.025, substantially outperforming all seven baselines. In particular, it yields approximately 36% and 54% lower error values compared to the next best-performing baselines NRDF (0.039) and IMDC (0.054), respectively. Furthermore, MIND demonstrates an improvement of nearly 94%, when compared against the best mesh-based baseline DM (0.428), highlighting the effectiveness of the proposed approach.

### C. Execution Time Analysis

Next, we investigate the execution time for estimating the dense correspondence between input object pairs. The tests are performed on an NVIDIA RTX 4090 GPU with 24 GB memory using the extended BSRC data, where each point cloud contains approximately 8200 points and each reconstructed mesh comprises approximately 8200 vertices. The average values are reported in Table II. Evidently, the computation times of point-based approaches are significantly lower than that of the mesh-based approaches. In comparison to the baselines, MIND achieves a competitive execution time of $0.225$ s, demonstrating its operational efficiency for real-world robotic applications.

### D. Feature Interaction and Ablation Analysis

The multiple features to design MIND are carefully chosen. Here we provide details on how these features complement and reinforce one another with principled explanations and quantitative ablation studies. The SDF and normal features directly complement each other with the SDF gradients giving the local surface orientations. This additional supervision of surface reconstruction with normals constrains the network to accurately learn surface-adjacent relations. Since surface processing does not depend on far-field SDF, any possible impact of gradient decay distant from the surface with SDF learning is negligible. Ablation of the geometric awareness ($ablate\_geom$), i.e., removal of the normal features from $\mathcal{L}_{\text{GDCD}}$, results in an average 20% rise in error as seen in Table II, highlighting their contribution. The FCGF feature uses convolutional layers [18] and therefore encodes a geometric neighborhood. With SDF being a point-level feature, the combination with FCGF improves the system by encoding the geometric features with a wider receptive field. This impact is shown in Table II by the 8% increase in error value with the ablation of the FCGF features ($ablate\_\mathcal{F}$).

The point cloud density awareness incorporated with the $\mathcal{L}_{\text{GDCD}}$ loss, enhances the model's capacity to discriminate between similar geometric parts across diverse category-level objects for accurate cross reconstruction. The impact of density awareness ($ablate\_den$) is evaluated by using simple CD [19] instead of $\mathcal{L}_{\text{GDCD}}$ by which a 100% rise in error value is observed. Then the keypoint feature reinforces the whole system by guiding the optimization such that the cross reconstruction respects key geometric feature correspondences. The error increases by 80% with the ablation of keypoints ($ablate\_k$) as reported in Table II. Hence the $\mathcal{L}_{\text{GDCD}}$ loss clearly exposes the cross reconstruction error, and the keypoint feature supports to minimize this error, facilitating high-quality dense correspondence predictions. Note that the keypoints do not increase network complexity, since our system does not learn the keypoint prediction, but instead uses keypoint ground truths to guide the optimization.

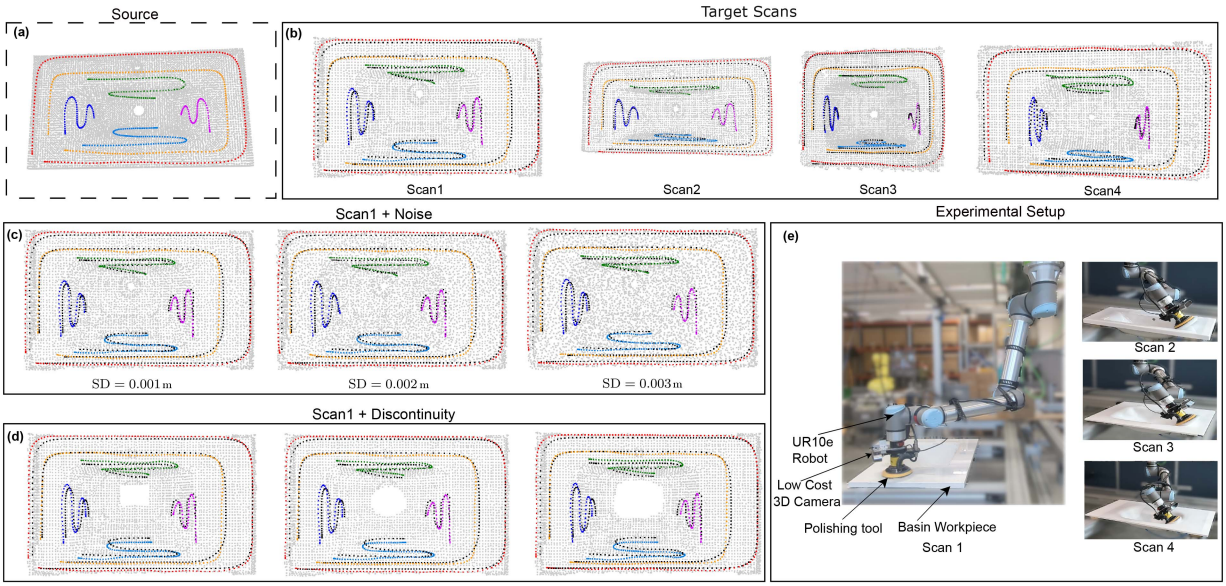


Fig. 4. (a) Virtual point cloud with source trajectory. (b) Illustration of trajectory transfer from (a) to target real-world 3D scans, predictions shown in color and expert annotated ground truths in black. (c) Performance with noise perturbations. (d) Performance with added discontinuities. (e) Experimental setup.

### *E. Application Case Study: Robotic Basin Surface Polishing*

In this case study, we qualitatively evaluate the dense correspondence performance on diverse basins including real-world scans captured using a low-cost 3D camera. Next, the trajectory transfer capacity across basins is first examined qualitatively, and then its accuracy, maximum deviation and repeatability is evaluated quantitatively. Finally, a proof-of-concept real-world demonstration of process trajectory and knowledge transfer using MIND for robotic surface polishing on geometry-varying basins is presented.

*1) Region Retrieval on Unseen Basins:* We qualitatively compare the performance of MIND with the NRDF [2] and IMDC [24] baselines, which achieve the next-best dense correspondence accuracy as reported in Table II. To this end, we define eight ROIs on the point cloud of a virtual source basin including flat surfaces, curves, and edges, as shown in Fig. 3(a). Then we estimate the corresponding ROIs on a target basin with significant variations in geometry, and the results are given in Fig. 3(c). We observe that MIND recovers the target ROIs with higher accuracy than the baselines. For instance, NRDF erroneously predicts the ROI-3 on the right edge of the source basin to be on a flat surface on the target. Similarly, the ROI-7 which ideally corresponds to the inner curvature, is not only incomplete but also erroneously extends onto flat areas. IMDC, on the other hand, appears to learn an incorrect deformation from source to target.

We further examine the ability of MIND to retrieve corresponding ROIs on diverse virtual point clouds, and real scans, as depicted in Fig. 3(b) and (d), respectively. We observe qualitatively that MIND is able to recover reasonable corresponding ROIs on both target virtual point clouds and real-world 3D scans of basins with significant variations in geometry.

*2) Trajectory Transfer to Unseen Basin Scans:* We examine how general trajectories are transferred via the dense correspondence of MIND across different real-world 3D basin scans. To this end, we define six reference trajectories on the virtual source basin covering flat and curved ROIs as indicated in Fig. 4(a). We then utilize MIND to estimate corresponding trajectories on all four 3D scans. Fig. 4(b) shows the predicted trajectories in similar color of the reference for qualitative comparison. Furthermore, to facilitate a quantitative analysis, we use ground truth trajectories annotated by an expert operator as shown in black in Fig. 4(b). To provide a comprehensive view of MIND's reliability and practical applicability, we evaluate the prediction accuracy using root mean squared error (RMSE) between prediction and ground truth, the maximum deviation (MD) between trajectory points which corresponds to the worst-case safety margin, and the repeatability is assessed with the standard deviations (SD) across 10 trials for each test basin. The mean values are reported in Table III, which also shows the comparison to the NRDF method. MIND achieves an accuracy of 0.018m, which is a 72% improvement over NRDF, while also improving MD and SD by 72% and 57%, respectively, demonstrating higher accuracy and repeatability. In practice, a tolerance of 0.039m (15% of the tool diameter, 0.26 m) is appropriate for large-area surface processing, where our MD of 0.024 m remains within bounds.

Our scans are captured from a top-down viewpoint which contains about 50% occlusion as the bottom part of basin is missing, and measurement noise due to the low-cost 3D camera. Nevertheless, to explicitly quantify robustness, we perform controlled perturbation studies by adding Gaussian noise with SD of 0.001 m, 0.002 m and 0.003 m, and manually introducing

TABLE III
QUANTITATIVE EVALUATION OF TRAJECTORY TRANSFER CAPACITY. VALUES ARE GIVEN IN M

| | Scan | | | Scan + Noise | | | Scan + Discontinuity | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | RMSE | MD | SD | RMSE | MD | SD | RMSE | MD | SD |
| MIND | **0.018** | **0.024** | **0.006** | **0.020** | **0.026** | **0.008** | **0.022** | **0.030** | **0.009** |
| NRDF [2] | 0.064 | 0.087 | 0.014 | 0.068 | 0.090 | 0.017 | 0.066 | 0.089 | 0.016 |

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

discontinuities in the scan. A sample qualitative result with added noise and discontinuities are shown in Fig. 4(c) and (d), respectively. The quantitative results in Table III shows that MIND is robust against perturbations and missing data in comparison with NRDF.

*3) Robotic Demonstration:* Finally, we utilize the dense correspondence and trajectory transfer capacity of MIND to perform a real-world experiment of robotic polishing on basins. The experimental task is to polish target basins with significant variations in geometry by transferring the process trajectory and associated knowledge defined on the virtual source basin. The process knowledge specific to each trajectory includes the orientation and velocity of the end-effector, and the diameter and rotation speed of the tool. The experimental setup is illustrated in Fig. 4(e), and the example robotic demonstration is given as supplementary video. As a proof-of-concept, MIND achieves reasonable trajectory transfer on real-world geometry-varying basins. For enhanced industrial safety and reliability, particularly in scenarios involving complex geometrical features, MIND can be integrated with existing flexible path-planning frameworks such as [26], which account for collisions, process constraints, and tolerances.

## V. Conclusion

In this letter, we propose *multi-feature implicit neural descriptor* (MIND) for solving robotic surface processing in a challenging high-mix low-volume production scenario. MIND are carefully designed descriptors with systematic incorporation of multiple features such as fully convolutional geometric features, point normals, signed distance function, and geometric feature-rich sparse keypoint correspondences. We demonstrate with a proof-of-concept robotic polishing experiment that MIND is able to transfer process trajectories and associated knowledge across diverse geometry-varying basins by operating on real-world 3D scans. This specifically circumvents the need of CAD models, robot reprogramming, and computationally expensive CAD/CAM software modules for trajectory estimation on new workpieces. Limitations of the current approach include challenges in dynamically varying the tool contact angles, and the trajectory coverage density with respect to the variation in the surface region being processed, which will be investigated in future research.

## References


[1] S. Schneyer, A. Sachtler, T. Eiband, and K. Nottensteiner, "Segmentation and coverage planning of freeform geometries for robotic surface finishing," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 5267–5274, Aug. 2023.

[2] A. Pratheepkumar, M. Ikeda, M. Hofmann, F. Widmoser, A. Pichler, and M. Vincze, "NRDF-neural region descriptor fields as implicit ROI representation for robotic 3D surface processing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2024, pp. 12955–12962.

[3] H. Kim, H. Lee, S. Chung, and C. Kim, "User-centered approach to path planning of cleaning robots: Analyzing user's cleaning behavior," in *Proc. ACM/IEEE Int. Conf. Hum.-Robot Interaction*, 2007, pp. 373–380.

[4] M. Xiao, Y. Ding, and G. Yang, "A model-based trajectory planning method for robotic polishing of complex surfaces," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2890–2903, Oct. 2022.

[5] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *Int. J. Robot. Res.*, vol. 24, no. 11, pp. 883–898, 2005.

[6] A. M. A. Zaki, M. Carnevale, C. Schlette, and H. Giberti, "On the use of low-cost 3D stereo depth camera to drive robot trajectories in contact-based applications," *Int. J. Adv. Manuf. Technol.*, vol. 128, no. 9, pp. 3745–3759, 2023.

[7] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4460–4470.

[8] A. Simeonov et al., "Neural descriptor fields: SE(3)-equivariant object representations for manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 6394–6400.

[9] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1830–1836.

[10] Z. Huang et al., "NIFT: Neural interaction field and template for object manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1875–1881.

[11] Y. Cai, J. Gao, C. Pohl, and T. Asfour, "Visual imitation learning of task-oriented object grasping and rearrangement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2024, pp. 364–371.

[12] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[13] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-DoF grasp detection via implicit representations," in *Proc. Robot.: Sci. Syst.*, 2021. [Online]. Available: https://www.roboticsproceedings.org/rss17/p024.html

[14] C. Unger, C. Hartl-Nesic, M. N. Vu, and A. Kugi, "ProSIP: Probabilistic surface interaction primitives for learning of robotic cleaning of edges," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2024, pp. 5956–5963.

[15] S. Kana, S. Lakshminarayanan, D. M. Mohan, and D. Campolo, "Impedance controlled human–robot collaborative tooling for edge chamfering and polishing applications," *Robot. Comput.-Integr. Manuf.*, vol. 72, 2021, Art. no. 102199.

[16] P. Möhl, A. Pratheepkumar, M. Ikeda, and A. Pichler, "Morphing based transfer of demonstrated surface finishing trajectories to point clouds of similar objects," *Procedia Comput. Sci.*, vol. 253, pp. 1002–1011, 2025.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[18] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[19] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware Chamfer distance as a comprehensive metric for point cloud completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 29088–29100.

[20] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[21] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.

[22] Y. You et al., "KeypointNet: A large-scale 3D keypoint dataset aggregated from numerous human annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13647–13656.

[23] J. Zhu et al., "DenseMatcher: Learning 3D semantic correspondence for category-level manipulation from one demo," in *Proc. 13th Int. Conf. Learn. Representations*, 2025. [Online]. Avaialble: https://openreview.net/forum?id=8oFvUBvF1u

[24] F. Liu and X. Liu, "Learning implicit functions for topology-varying dense 3D shape correspondence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 4823–4834.

[25] R. Magnet, J. Ren, O. Sorkine-Hornung, and M. Ovsjanikov, "Smooth non-rigid shape matching via effective dirichlet energy optimization," in *Proc. Int. Conf. 3D Vis.*, 2022, pp. 495–504.

[26] T. Weingartshofer, B. Bischof, M. Meiringer, C. Hartl-Nesic, and A. Kugi, "Optimization-based path planning framework for industrial manufacturing processes with complex continuous paths," *Robot. Comput.-Integr. Manuf.*, vol. 82, 2023, Art. no. 102516.