High-Degrees-of-Freedom Dynamic Neural Fields for Robot Self-Modeling and Motion Planning

Lennart Schulze
Columbia University
New York, NY 10027, USA
lennart.schulze@columbia.edu

Hod Lipson
Columbia University
New York, NY 10027, USA
hod.lipson@columbia.edu

Abstract

A robot self-model is a task-agnostic representation of the robot's physical morphology that can be used for motion planning tasks in absence of classical geometric kinematic models. In particular, when the latter are hard to engineer or the robot's kinematics change unexpectedly, human-free self-modeling is a necessary feature of truly autonomous agents. In this work, we leverage neural fields to allow a robot to self-model its kinematics as a neural-implicit query model learned only from 2D images annotated with camera poses and configurations. This enables significantly greater applicability than existing approaches which have been dependent on depth images or geometry knowledge. To this end, alongside a curricular data sampling strategy, we propose a new encoder-based neural density field architecture for dynamic object-centric scenes conditioned on high numbers of degrees of freedom (DOFs). In a 7-DOF robot test setup, the learned self-model achieves a Chamfer-L2 distance of 2% of the robot's workspace dimension. We demonstrate the capabilities of this model on a motion planning task as an exemplary downstream application.

1. Introduction

Neural fields paired with differentiable rendering allow to learn accurate 3D scene information from annotated 2D images. This is achieved by overfitting a neural network to the scene observed from multiple camera views using a photometric reconstruction loss [23]. At inference time, this allows to render realistic images of the scene from novel camera views. Due to the importance of scene representations in robotics, neural field extensions have evolved focusing on use cases in this area. While most of these approaches [19, 1, 24, 21] use neural fields to capture and harness information about the robot's environment such as for reconstruction, navigation, or localization tasks, here we propose to learn neural fields to represent - and control - the robot.

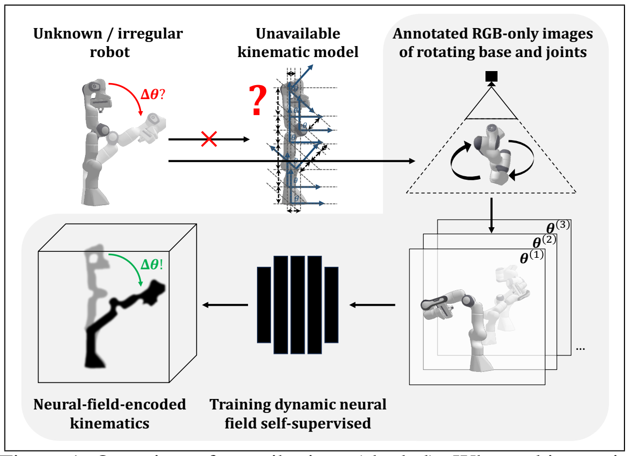

Unknown / irregular robot
$\Delta\theta$?
Unavailable kinematic model
?
Annotated RGB-only images of rotating base and joints
$\theta^{(3)}$
$\theta^{(2)}$
$\theta^{(1)}$
...
$\Delta\theta$!
Neural-field-encoded kinematics
Training dynamic neural field self-supervised

Figure 1. Overview of contributions (shaded): When a kinematic model is unavailable for the robot, 1) our method to collect annotated depth-free image data can be used instead to train 2) a high-DOFs dynamic neural density field that the robot uses as self-model whose 3) forward and inverse kinematic capabilities enable motion planning applications.

We target the task of robot self-modeling, the (robot's) ability to learn a representation of the robot's kinematics from observing its behavior without human interference, in any state of its existence. Similar to a mental image of oneself, these models can continually be updated to reflect the state of the robot, rendering them advantageous over classical geometric kinematic models which are usually engineered once, may be mismatched to the current state of the robot, and are unavailable for unknown robots [29]. For these reasons, learning-based approaches to robot self-modeling emerged. Despite being successful, a major drawback is their dependence on supervised samples or, in the self-supervised case, depth data in the training distribution. These requirements hinder the readiness of target applications of self-modeling in real-world scenarios where this information is not available, such as after damage to the robot's body during deployment. Particularly, a recent approach [6] to learn a full-body kinematic forward model as neural-implicit representation requires images annotated with depth values from an RGB-D camera. In this work we propose to solve this obstacle by learning neural fields in a self-supervised manner directly from 2D images, only annotated with camera parameters and the dynamic configuration. Consequently, we approach the task of learning a neural-implicit full-body kinematic model from unlabeled kinematic data through training a dynamic neural field that offers downstream compatibility (Figure 1).

We achieve this by introducing a new type of *dynamic* neural fields. Previous work [27] has extended the static-scene setup of neural radiance fields [23] by establishing time as the fourth input dimension next to 3D coordinates, which together are mapped to density and color values. In contrast, in this work we introduce a high number of degrees of freedom (DOFs) as conditioning variables for a coordinate-to-density map that in complex interdependence change local parts of the scene, which has not been done for the purpose of robotic applications. Different from other work relying on deformation from a canonical representation [26], we propose a DOF-encoder-based dynamic neural density field, which is amenable to modeling the shapes of complex changing objects beyond robotics.

In summary, this work contributes the following:

- We introduce a curricular data sampling method and neural network architecture to represent high-DOFs object-centric scenes as dynamic neural density fields.
- We use our method to visually learn the first robot self-model without depth information and from a single camera view, and quantify its quality experimentally.
- Extending [6], we discuss and demonstrate downstream applications of neural-field self-modeled kinematics in motion planning.

We discuss preliminaries and prior work in section 2, and introduce our method and its applications in section 3. We subsequently detail the experimental setup for a sample robot in section 4, and discuss the results in section 5, before concluding with regards to future work in section 6.

2. Background and related work

Robot self-modeling. A self-model is a task-agnostic, general-purpose representation of a robot's physical shape and dynamics that can be acquired and continuously updated at any time without a human in the loop [16, 10]. The objective to enable machines to produce a cognitive model of themselves to guide their behavior has been inspired by similar behavior in human beings [28]. Practically, whenever a geometric kinematic model, which captures the spatial relations and physical constraints of the robot's links and joints manually as result of simulation and engineering, is unavailable, the ability to self-model is required. In particular, when the robot's kinematics are altered, for instance through damage or undocumented body manipulation, the robot can learn an updated model and without the need for human engineering [4, 5].

Early approaches to robot self-modeling have leveraged analytical, probabilistic, and evolutionary methods [4, 11, 5]. Learning-based approaches to implicitly represent self-models were first presented in [17], necessitating training samples that are labeled with end effector positions. Similarly, certain approaches [12, 13] pre-determine the set of parameters to learn for a system, identified based on prior knowledge about the shape or function. The most recent, partially self-supervised approach which constructs an agnostic self-model without such information, [6], still requires depth information, which is used to learn an SDF-based occupancy query model. In all of these approaches, data acquisition plays a crucial role with strategies ranging from entirely random [6], to interactive [3], and targeted-exploratory [12, 2]. This work builds on the agnostic, neural-implicit class of representation proposed in [6] and alleviates the depth requirement using neural fields, while introducing a curricular-random data acquisition strategy for its training.

Neural (radiance) fields. Neural fields are continuous maps from any spatial coordinate in 3D space $\mathbf{x} = (x, y, z)$ to a scalar or vector. In neural radiance fields (NeRF), each point is assigned a tuple of density and color. The map is parameterized via a neural network Φ, such as a multi-layer perceptron (MLP), overfit to the specific scene,

$$f_\Phi : (\mathbf{x}, \mathbf{d}) \rightarrow (\sigma, \mathbf{c}) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ is the coordinate vector, $\mathbf{d} \in [0,1]^3$ is the viewing direction, $\sigma \in [0, \infty)$ is the predicted density, and $\mathbf{c} \in [0,1]^3$ is the predicted RGB color. Both to march a ray through the scene to obtain point coordinates $\mathbf{x}$ and to compute the viewing direction $\mathbf{d}$, the camera pose ${}^{\boldsymbol{w}}\boldsymbol{T}_{\boldsymbol{c}}$ is used.

NeRFs are neural-implicit representations of a scene since novel views can be rendered by querying the learned map, without the need to store 3D information explicitly, such as in point clouds or voxels. From the field over the 3D space, 2D projections to images from arbitrary camera views are rendered via volume rendering [14, 22]: The color at a pixel $\mathbf{C}$ is computed by integrating the product of color, density, and visibility of the points residing on the ray that was marched through the scene from the projection plane within the depth view bounds. The visibility $\hat{T}_i$ of a point depends on the density values of the points between the projection plane and that point. Using quadrature, the integral is approximated on N points, which are sampled in stratified manner from bins on the ray, as follows:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \hat{T}_i \; \alpha(\sigma_\Phi(\mathbf{x_i})\delta_i) \; \mathbf{c}_\Phi(\mathbf{x_i}, \mathbf{d}) \tag{2}$$

$$\hat{T}_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_\Phi(\mathbf{x_j})\delta_j\right) \tag{3}$$

Here, $\mathbf{r} = \mathbf{o} + t\mathbf{d}$ is the pixel-corresponding ray marched through the scene scaled by depth t; $\alpha(\sigma) = 1 - \exp(-\sigma)$ maps density values into the range $[0, 1]$; and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent points on the ray.

The differentiable nature of volume rendering allows to train the MLP via minimizing a photometric reconstruction loss. Given training images from different views centered on the same object, rays are marched through the scene using the provided camera poses. The predictions on the rays are rendered into images from the same views, allowing to use the mean-squared error (MSE) loss to update the MLP parameters via backpropagation.

Dynamic neural fields have emerged to represent scenes with moving parts, for instance over the time dimension [27]. In extension, numerous works have aimed at modeling human bodies, customarily using prior knowledge about their shape or multi-view video training data [7, 30]. Related to our work, we propose a new dynamic neural field architecture geared towards shape-unknown objects with many DOFs that are interdependent, trained on single-view images only.

3. Method

3.1. High-DOFs dynamic neural density field

We propose to extend neural fields to dynamic scenes in which changes are anchored in interdependent DOFs of the object in the scene. For this purpose, we condition the map f_Φ on the k-dimensional configuration of the object $\boldsymbol{\theta} = \left[\theta_0, \ldots, \theta_{k-1}\right]^T$ that drives the observed changes.

$$f_\Phi : (\mathbf{x}, \boldsymbol{\theta}) \rightarrow (\sigma) \tag{4}$$

To model the shape, the field does not assign color, and thus is independent of the viewing direction. Nonetheless, color can be included for the purpose of training the model via a photometric loss.

Specifically to learn a self-model of a robot, the map is conditioned on the joint configuration of the robot composed of k joint values and thus learned as $3 + k$-dimensional neural field. We can subsequently compute density fields in novel configurations, and render these into projections from novel views, assigning an arbitrary color.

Encoder-based architecture. We parameterize the map via a neural network based on [23], that is an MLP with ReLU activations. Trivially extending this architecture by inserting $\boldsymbol{\theta}$ as additional input parameters leads to unsatisfactory results. Consequently, we extend [6]'s approach and introduce separate encoders for the spatial and the new conditioning input variables. This harnesses the independence of the spatial coordinates and the DOFs configuration and promises to learn useful representations resulting from the combinations of the constituents of each group.

Applying the shuffled curriculum learning approach introduced below, however, results in continual forgetting of previously learned relationships of DOFs as the training progresses, reducing performance on previously well reconstructed samples. For this reason, we introduce DOF-individual encoders, MLPs that encode each input variable. Since gradients flowing back to the weights of these MLPs will be zero when the joint values are zero in curriculum-learning batches in which exclusively other DOFs are sampled, we argue this improves the memorability of useful features for the behavior introduced by each DOF individually. The outputs of these individual encoders are concatenated and chained to the group-based encoders. Hence the overall model is described by:

$$f_\Phi(\mathbf{x}, \boldsymbol{\theta}) = \\ \Phi_{\text{MLP}}\left(\Phi_{\text{Enc}_\mathbf{x}}\left(\bigoplus_{l=1}^{3} \Phi_{\text{Enc}_{x_l}}(x_l)\right), \Phi_{\text{Enc}_{\boldsymbol{\theta}}}\left(\bigoplus_{l=0}^{k-1} \Phi_{\text{Enc}_{\theta_l}}(\theta_l)\right)\right) \tag{5}$$

where $\oplus$ is the concatenation operator and $\Phi(x) = h_n \circ \ldots \circ h_1(x)$ is an n-layer MLP with ReLU activations. The architecture is shown in Figure 2.

Following [23], we train two models of this type to separately model spatially coarse and fine predictions. The second model evaluates points on the ray sampled from regions of t where the first model has resulted in higher density predictions. All points are used to render the projection of a ray. To equip the model with the ability to better predict fine, high-frequency structures, [23] employ a sine-cosine positional embedding method. While being effective for that purpose, the high dimensionality of the embedding hinders the learning of the true physical movement associated with traversing the DOF value ranges. Consequently, we remove the embedding and substitute it with the normalized original values, resulting in $3 + k$-dimensional inputs to the network which are passed into the DOF encoders.

Learning from a one-camera setup. To circumvent the need for multiple cameras observing the robot to produce the neural-field self-model, which limits real-world applications, we harness the mobility of the robot's base. Given the configuration of the robot $\boldsymbol{\theta}$ and the camera pose as camera-to-world transform ${}^{\boldsymbol{w}}\boldsymbol{T_c}$, we enforce the first DOF to be the base rotation. Since rotating the object at its base is equivalent to rotating the camera about the upward axis, multi-

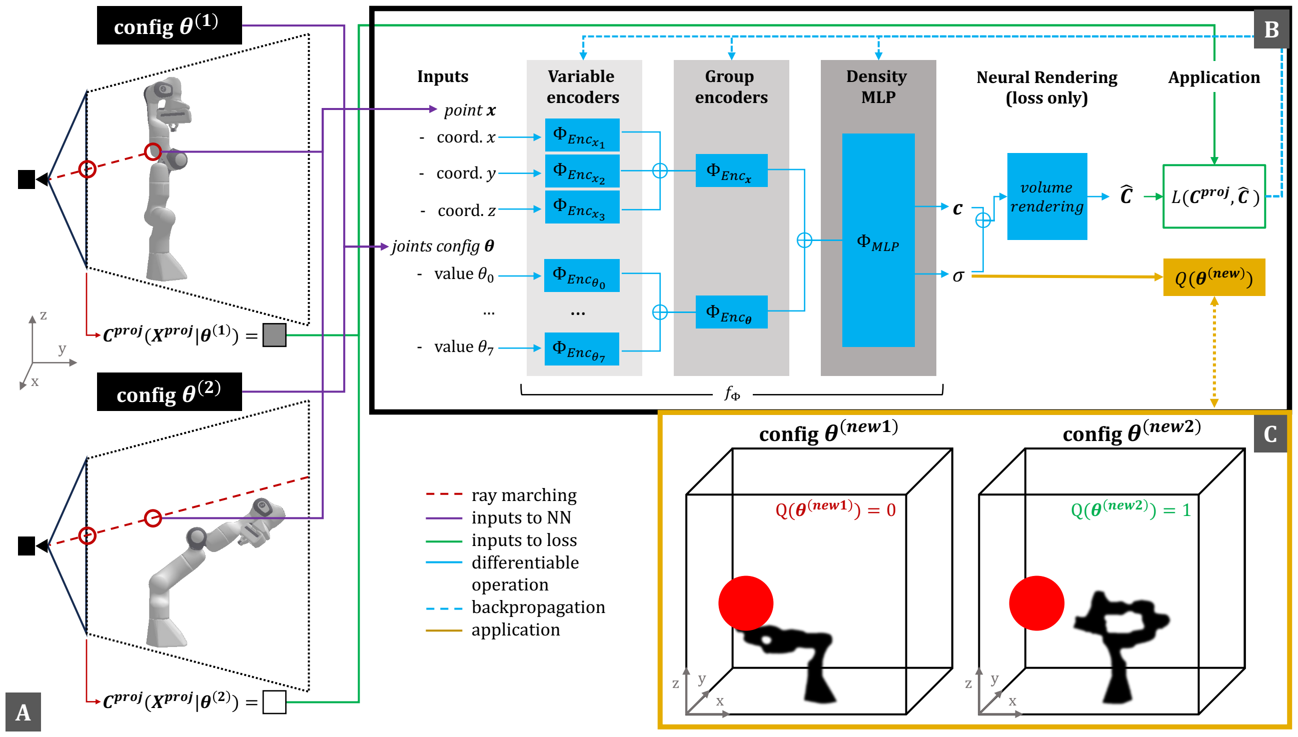


Figure 2. Overview of proposed method. A - Training images of robot in different configurations: While the point coordinates and the annotated configuration are the inputs to the neural network, the true projected color of the pixel is used for the reconstruction loss. B - Neural network architecture: The individual DOFs, alongside the spatial coordinates, are individually encoded, concatenated and group-wise encoded, and concatenated and processed to an output density. C - The trained neural density field is used to evaluate the membership of a configuration in the configuration space with respect to an obstacle by predicting the densities of points queried from its volume.

view consistency for the density predictions can be ensured by assigning:

$$({}^{w}\boldsymbol{T_c})' \leftarrow \boldsymbol{R_z}(\theta_0)\,{}^{w}\boldsymbol{T_c} \tag{6}$$

$$\theta_0' \leftarrow 0 \tag{7}$$

Curricular training data. Due to the large space of configurations and the serial dependence among the k DOFs, the most distant-from-the-base joint's position depending on all previous $k-1$ DOFs, learning a high-DOFs neural field is difficult. Thus, the data generation approach is crucial to the success of the model inferring the correct marginal influence of each DOF. We propose to use a curriculum-learning-inspired sampling approach. For the set of all DOF indices $\Theta = \{l\}_{l=1}^{k-1}$, we compute the powerset, that is the set of all subsets of Θ, and sort it in ascending order by magnitude, excluding the empty set: $S_\Theta = \{s\}_{s\subseteq\Theta}$. For each set of DOF indices $s \in S_\Theta$, training samples are generated by randomly sampling joint values from the permissible ranges of the DOFs in s. The values of the remaining DOFs are fixed to zero. In this manner we encourage learning the contribution of each DOF first by itself, and then in combination with other DOFs in order of increasing complexity, until all DOFs are interacting.

To encourage learning the behavior across the full ranges of the considered DOFs, we sample from uniform distributions. For example:

$$\boldsymbol{\theta}^{(1,4)} = \begin{bmatrix} 0 & \theta_1 \sim D_1 & 0 & 0 & \theta_4 \sim D_4 & \dots & 0 \end{bmatrix} \tag{8}$$

$$D_i = U(\theta_i^{(max)}, \theta_i^{(min)}) \tag{9}$$

We find that shuffling the training images such that images with different numbers of active DOFs lie in the same batch improves the training performance.

Training. We optimize our model via the MSE photometric loss between ground-truth pixels and rendered pixels. In the given setup, our experiments have suggested that keeping the RGB output improves training performance. Nonetheless, the density prediction is the only output kept to be used in the self-model once training has concluded. To train density-output-only high-DOFs neural fields, the MSE loss may be used between binarized images and volume renderings with $\mathbf{c}$ set to black.

3.2. Neural-field self-model and applications

Self-Model. The trained map f_Φ is a neural-implicit kinematic model of the robot since it enables to reconstruct the robot's shape conditioned on its joint configuration. By learning this model only from annotated 2D images, this replaces the need to know the robot geometry altogether. Such a neural-implicit model gives rise to applications in motion planning, similar to the use of a classical geometric kinematic model.

Motion planning: Touching a target object via inverse kinematics. Extending [6], we demonstrate an immediate downstream use case of the model as motion planning. Due to the differentiable nature of the model's forward prediction of the density of a point given the configuration, we can compute the inverse kinematics, that is the configuration such that a point is occupied. For this purpose, the MLP parameters are fixed and the input joint values are optimized via projected gradient descent (PGD) to minimize the deviation from the target density level.

By choosing appropriate points, the robot can, for example, compute how to reach an object. Furthermore, by selecting the initial configuration of the optimization to be the current configuration of the robot and acting in an obstacle-free environment, the inverse kinematics optimization steps can be cast as a path to reach the target. Precisely, given information about the target, we uniformly sample N surface points $O_s = \{\mathbf{x_i}\}_{i=1}^{N}$ and query the robot's density on them, leveraging that density on the surface above a threshold τ indicates touch. In the fine model, starting from a no-touch configuration $\boldsymbol{\theta}^{(0)}$, we minimize the following loss, which will be ≤ 0 when the target is reached:

$$L(\boldsymbol{\theta}, O_s) = \min_{\mathbf{x} \in O_s} \left[-\alpha(f_\Phi(\mathbf{x}, \boldsymbol{\theta}))\right] + \tau \tag{10}$$

The final ReLU activation at Φ_{MLP}'s output unit for σ is removed to produce non-zero gradients.

To enforce that the joint configuration found and every step of the optimization is within the joint limits, after each step of size η the joint values are projected back into the k-dimensional ball representing the permissible ranges:

$$\boldsymbol{\theta}^{(j+1)} = \Pi_{\boldsymbol{\theta}^{(min)}}^{\boldsymbol{\theta}^{(max)}} \left[\boldsymbol{\theta}^{(j)} - \eta \frac{\partial L(\boldsymbol{\theta}^{(j)}, O_s)}{\partial \boldsymbol{\theta}^{(j)}}\right] \tag{11}$$

If only the inverse kinematic problem is considered, random initializations of the start configuration can accelerate the optimization process to find a configuration that results in a target point's occupation.

Motion planning: Configuration space. For more complex constraints and in the presence of obstacles, customary motion planning algorithms can be used with the self-model. Any planning algorithm using configuration space information, that is a binary map over the k-dimensional space of possible configurations indicating which configuration is collision-free, is compatible with the neural-field-implicit kinematic model. Given the high-DOFs neural density field and information about obstacle(s) in the scene, a configuration is valid if the maximum density of the robot among N uniformly sampled points from the volume of the obstacle O_v is below a threshold. Consequently, sampling-based motion planning methods that search the configuration space such as Probabilistic Road Map [15] or Rapidly-Expanding Random Trees [18], can be used. The membership in the configuration space can be evaluated as:

$$Q(\boldsymbol{\theta}) = \begin{cases} True & \text{if: } \max_{\mathbf{x} \in O_v} \left[\alpha(f_\Phi(\mathbf{x}, \boldsymbol{\theta}))\right] < \tau \\ False & \text{else} \end{cases} \tag{12}$$

4. Experimental Setup

We experimentally demonstrate our method on a 7-DOF robot in simulation.

Training distribution. We apply the curriculum data generation described above with 16 different image samples per set s and 6 random base rotations sampled anew for each image, totalling $5,588$ annotated image samples. We generate the samples in simulation of the Panda robot [8] with 7 joints and a rotatable base ($k = 8$), using the Pybullet simulator [9]. We group batches of 15 samples, and, to avoid overfitting to one batch, only process $10,240$ rays per sample image in the batch.

Training. We use the described architecture with 3-layer DOF-individual encoder MLPs, 1-layer coordinate encoder MLPs, 2-layer group encoder MLPs, and the final 7-layer density MLP. We train using the Adam optimizer for 1,320,000 steps with a learning rate of $4e-5$ and optimize the parameters of all MLPs together.

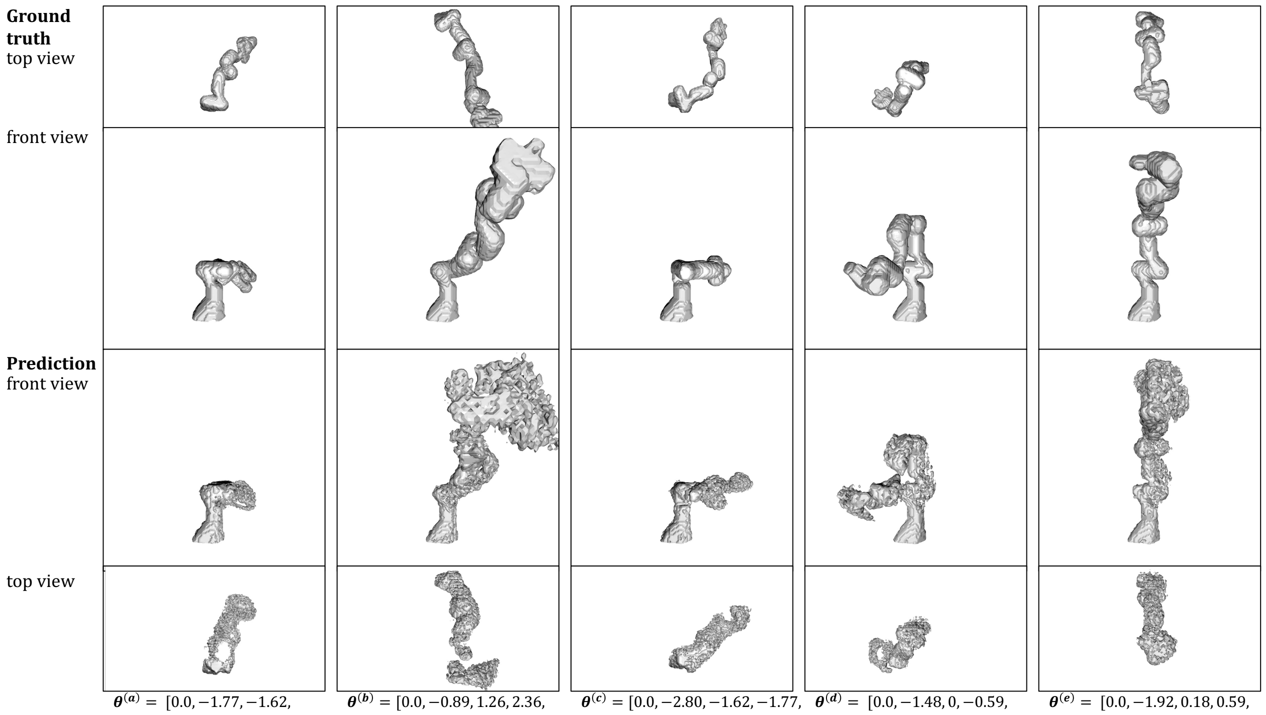
Predicted visualizations. We produce point clouds by querying the field from two camera poses at the front of the scene on the x- and y-axes. Points with alpha values above 0.015 are kept, determining the isolevel. On the fused point cloud, marching cubes [20] reconstruction is applied to generate a triangle mesh, followed by hole repair and Taubin smoothing [31] algorithms.

Ground truth visualization. We produce ground-truth 3D data by simulating the true geometric model in Pybullet. The ground-truth point cloud for a joint configuration is the fusion of six point clouds from RGB-D images obtained from two camera views per axis at either end of the axis with the view centered at the object. The mesh is produced identically to the predicted meshes.

Metrics. To assess the quality of our model, we compare the ground-truth against the predicted meshes. First, we use the customary Chamfer-L2 distance, the shortest Euclidean distance of each point in a set to any point in the other set, applied symmetrically and averaged over all points. This returns an average spatial offset per point. We generate the point sets by sampling uniformly from the mesh surfaces. In addition, as measures for the spatial similarity of the predicted shape, we compute two intersection over

Distance metric	config a	config b	config c	config d	config e	test set (n=30)
Chamfer-L2 (m) ↓	.017	.019	.053	.013	.013	.024
Chamfer-L2 (% of workspace-z) ↓	1.35	1.51	4.22	1.06	1.07	1.94
Surface area IoU ↑	.501	.479	.408	.571	.572	.496
Hull volume IoU ↑	.685	.607	.336	.714	.690	.573

Table 1. Spatial distances of predictions against ground truth in five different random test configurations.



$\boldsymbol{\theta}^{(a)} = [0.0, -1.77, -1.62, 2.36, -0.62, -1.33, 0.0, 0.89]$, $\boldsymbol{\theta}^{(b)} = [0.0, -0.89, 1.26, 2.36, -0.62, -1.92, 2.47, -1.77]$, $\boldsymbol{\theta}^{(c)} = [0.0, -2.80, -1.62, -1.77, -1.24, -2.80, 2.28, -0.30]$, $\boldsymbol{\theta}^{(d)} = [0.0, -1.48, 0, -0.59, -2.48, -1.33, 1.33, 0.89]$, $\boldsymbol{\theta}^{(e)} = [0.0, -1.92, 0.18, 0.59, -1.24, -2.95, 0.76, 0.89]$.

Figure 3. Predicted vs. ground-truth meshes, smoothed and reconstructed via marching cubes from point clouds generated by querying the high-DOFs neural density field. Please also see the supplementary video [25].

union (IoU) metrics. Both are based on a union point cloud, constructed by fusion, and an intersection point cloud, constructed by keeping points with negative signed distance relative to the mesh defined by the other point cloud. We compute a 2D metric, relating the surface areas of the meshes reconstructed from the two point clouds, and a 3D metric, relating the volumes of the convex hulls of the same two meshes. The hull is generated from a fixed number of uniformly sampled surface points.

5. Results

Neural-field self-model. We show the predicted meshes from the 7-DOF robot self-model for five random test configurations from a fixed view in Figure 3. It can be observed that in each configuration, the prediction follows the shape of the ground truth, subject to small deviations in the rotations of smaller parts of the body. For the shown samples, this indicates that the model learned to correctly approximate the shape from the configuration, despite the large space of possible configurations. We find that density scales with the certainty in the prediction and that despite the solid material of the robot, most of the non-zero density values are in the lower regime as opposed to close to one. Consequently, the threshold selection, that is the marching cube isolevel, is a significant hyperparameter, since it controls the sensitivity with which sampled points are included. A low threshold may lead to a too large shape relative to the ground truth whereas a too high threshold may exclude parts of the body in whose prediction the model is less certain. We find that due to the serial nature of dependence among the DOFs in this scene - the first link's density only depends on the first joint value whereas the seventh link's density depends on all previous joints values - those excluded parts are the higher-up components of the body. Conversely, the highest density values belong to points in the base of

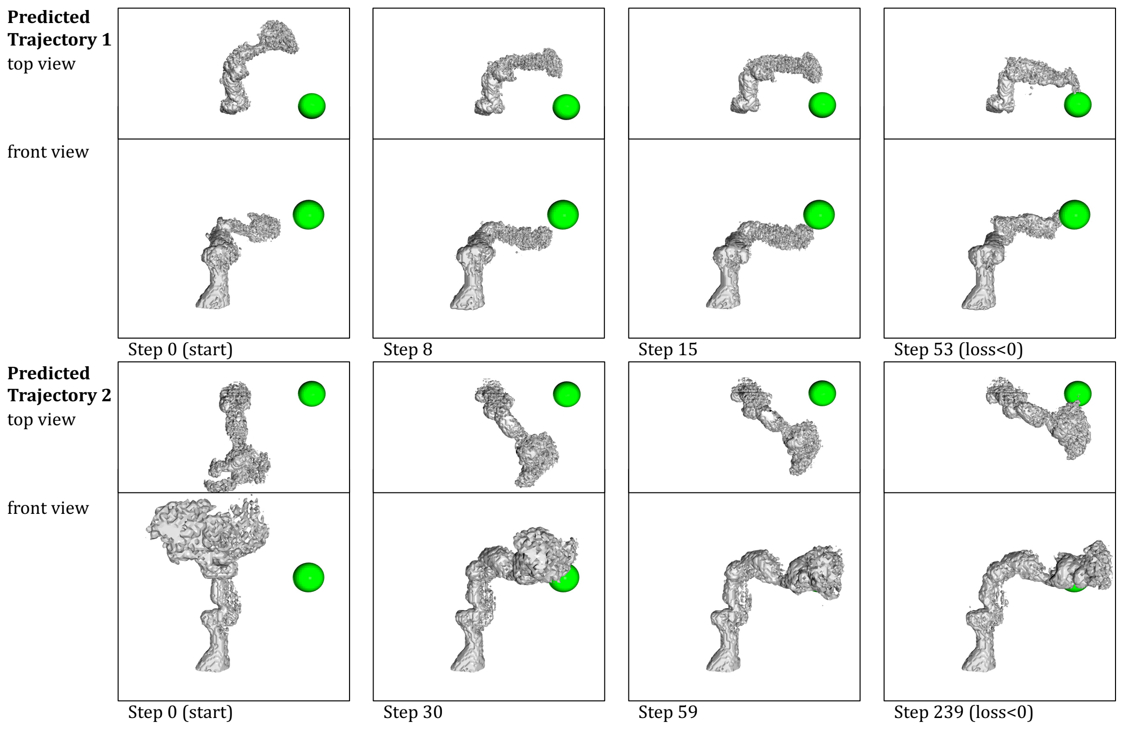


Figure 4. Joint angle optimization at different steps via projected input gradient descent. The robot changes its configuration to minimize a density loss that corresponds to touching a sphere. Please also see the supplementary video [25].

the robot, which does not move in response to the configuration. In addition, the querying resolution plays a crucial role, where the trade-off is computational cost against approximation of the true predicted model. In addition to the qualitative evaluations, numerical results on the spatial quality metrics are provided in Table 1. As most important metric, the mean of the Chamfer-L2 distance for the test set is 1.94% relative to the length of the shortest dimension of the workspace of the robot, 1.254m along the vertical axis. This indicates that, on average, each point on the mesh that was reconstructed from the points in the volume predicted to have sufficiently high density is close to a point on the robot's true surface given the queried configuration. Greater variance can be observed for the two IoU metrics. For the volume-based IoU, this is due to the constraint that the hull must convexly contain all points of the surface point cloud so that individual outliers have a large effect on its shape, and thus, volume. In addition, while marginally off-positioned predicted robot parts can still produce moderate Chamfer-L2 distances, these parts may not or only partially intersect with the ground-truth parts, leading to a lower value. The surface area is similarly sensitive to outliers while additionally depending on the smoothness of the surface.

Motion planning. In Figure 4, snapshots of two generated trajectories to touch an object from a start position in the displayed number of PGD steps are shown. The right images correspond to the first joint configurations yielding a loss lower than zero, that is those in which the robot's density on the sampled target object surface points is above the specified threshold ($\tau = 0.6$). Due to the joint-limit-projected optimization and the absence of obstacles, the optimization steps form a valid trajectory. It can be observed that the robot moves itself successfully into a configuration in which the sphere is touched. A similar approach can be used to evaluate the membership of a configuration in the configuration space if the sphere is treated as obstacle instead. We observe moderate sensitivity to hyperparameters such as learning rate and initial configuration. In addition, τ controls the closeness to the target in the final configuration.

6. Conclusion and outlook

In this work, we propose dynamic neural density fields conditioned on high DOFs. We use this method to learn the first neural-implicit self-model of a robot without a-priori depth or geometry information which can be used in lieu of classical kinematic models. For this purpose, we introduce an architecture that concatenates encoders at different levels of the semantic hierarchy of the input variables, and train it according to a curricular data sampling strategy that allows to learn and memorize the marginal effect of each

DOF present in the robot. By querying the neural density field conditioned on the configuration, we can predict the robot's spatial occupation in arbitrary configurations, resembling full-body forward kinematics. By querying at points of interest and optimizing joint values, the model can be used to compute inverse kinematics as well as the configuration space, enabling compatibility with downstream applications. In experiments with a simulated 7-DOF Panda robot, our model predicts the robot's dynamic occupation of space accurately to over 98% of the workspace's shortest dimension and is successful in a motion planning task.

A robot with the ability to read its joint configuration and access to one calibrated camera pointing at itself is able to learn its neural-field self-model. To advance the real-world readiness of this setup even further, we propose to conduct future work into the directions of limiting the training data to more sparsely observed DOFs configurations and training time to fewer steps, and of automatically estimating the camera parameters, reducing the annotation requirement. In addition, the integration of our approach, which models the robot via neural fields, with previous work, which models the robot's environment, would be beneficial. For instance, downstream tasks such as navigation or localization could be enhanced by considering the robot's current configuration and the robot's configuration could be optimized to help in these tasks. Finally, such a model may extend to multi-robot environments.

We highlight the readiness of our method to be used for dynamic-object scenes outside robotics, such as nature, animals, or humans, or general DOFs-controlled environments.

7. Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) AI Institute for Dynamical Systems (DynamicsAI.org), grant 2112085.

References

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[2] Brandon Amos, Laurent Dinh, Serkan Cabi, Thomas Rothörl, Sergio Gómez Colmenarejo, Alistair Muldal, Tom Erez, Yuval Tassa, Nando de Freitas, and Misha Denil. Learning awareness models. In *International Conference on Learning Representations*, 2018.

[3] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[4] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.

[5] Josh C Bongard and Hod Lipson. Automated damage diagnosis and recovery for remote robotics. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3545–3550. IEEE, 2004.

[6] Boyuan Chen, Robert Kwiatkowski, Carl Vondrick, and Hod Lipson. Fully body visual self-modeling of robot morphologies. *Science Robotics*, 7(68):eabn1944, 2022.

[7] Helena A Correia and José Henrique Brito. 3d reconstruction of human bodies from single-view and multi-view images: A systematic review. *Computer Methods and Programs in Biomedicine*, page 107620, 2023.

[8] Erwin Coumans. Pybullet robots, 2020.

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

[10] Anthony Dearden. *Developmental learning of internal models for robotics*. PhD thesis, Imperial College London, 2008.

[11] Kevin Gold and Brian Scassellati. Using probabilistic reasoning over time to self-recognize. *Robotics and Autonomous Systems*, 57(4):384–392, 2009.

[12] Kaiyu Hang, Walter G Bircher, Andrew S Morgan, and Aaron M Dollar. Manipulation for self-identification, and self-identification for better manipulation. *Science Robotics*, 6(54):eabe1321, 2021.

[13] Zhihong Jiang, Weigang Zhou, Hui Li, Yang Mo, Wencheng Ni, and Qiang Huang. A new kind of accurate calibration method for robotic kinematic parameters based on the extended kalman and particle filter algorithm. *IEEE Transactions on Industrial Electronics*, 65(4):3337–3345, 2017.

[14] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

[15] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[16] Robert Kwiatkowski. *Deep Self-Modeling for Robotic Systems*. Columbia University, 2022.

[17] Robert Kwiatkowski and Hod Lipson. Task-agnostic self-modeling machines. *Science Robotics*, 4(26):eaau9354, 2019.

[18] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.

[19] Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077, 2022.

[20] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, 8(2):1–15, 2003.

[21] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo localization using neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4018–4025. IEEE, 2023.

[22] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[23] 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.

[24] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1347–1356. PMLR, 08–11 Nov 2022.

[25] . `https://youtu.be/M47qy5nWB6Y`.

[26] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021.

[27] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[28] Philippe Rochat. Five levels of self-awareness as they unfold early in life. *Consciousness and cognition*, 12(4):717–731, 2003.

[29] Henry W Stone. *Kinematic modeling, identification, and control of robotic manipulators*, volume 29. Springer Science & Business Media, 1987.

[30] Mingyang Sun, Dingkang Yang, Dongliang Kou, Yang Jiang, Weihua Shan, Zhe Yan, and Lihua Zhang. Human 3d avatar modeling with implicit neural representation: A brief survey. In *2022 14th International Conference on Signal Processing Systems (ICSPS)*, pages 818–827. IEEE, 2022.

[31] Gabriel Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE international conference on computer vision*, pages 852–857. IEEE, 1995.