Optimization Beyond the Convolution: Generalizing Spatial Relations with End-to-End Metric Learning

Philipp Jund, Andreas Eitel, Nichola Abdo and Wolfram Burgard[1]

***Abstract*— To operate intelligently in domestic environments, robots require the ability to understand arbitrary spatial relations between objects and to generalize them to objects of varying sizes and shapes. In this work, we present a novel end-to-end approach to generalize spatial relations based on distance metric learning. We train a neural network to transform 3D point clouds of objects to a metric space that captures the similarity of the depicted spatial relations, using only geometric models of the objects. Our approach employs gradient-based optimization to compute object poses in order to imitate an arbitrary target relation by reducing the distance to it under the learned metric. Our results based on simulated and real-world experiments show that the proposed method enables robots to generalize spatial relations to unknown objects over a continuous spectrum.**

I. Introduction

Understanding and leveraging spatial relations between objects is a desirable capability of service robots to function in human-centered environments. However, our environments are rich with everyday objects of various shapes and sizes, making it infeasible to pre-program a robot with sufficient knowledge to handle all arbitrary relations and objects it might encounter in the real world. Instead, we should equip robots with the ability to learn arbitrary relations in a lifelong manner and to generalize them to new objects, see Fig. 1. For example, having learned how to place a book inside a drawer, a robot should be able to generalize this spatial relation to place a toy inside a basket.

In this work, we propose a novel, neural-network-based approach to generalize spatial relations from the perspective of distance metric learning. Rather than considering a pre-specified set of relations and learning an individual model for each, our approach considers a continuous spectrum of pairwise relations and learns a metric that captures the similarities between scenes with respect to the relations they embody. Accordingly, we use this metric to generalize a relation to two new objects by minimizing the distance between the corresponding scenes in the learned metric as shown in Fig. 2. Following the metric-learning approach by Chopra *et al.* [1], we use a variation of the siamese architecture [2] to train a convolutional neural network as a function that maps an input point cloud of a scene consisting of two objects to the metric space such that the Euclidean distance between points in that space captures the similarity between the spatial relations in the corresponding scenes.

This work has been supported by the German Research Foundation under research unit FOR 1513 (HYBRIS) and the priority program SPP 1527.

[1]All authors are with the Department of Computer Science, University of Freiburg, Germany. {jundp, eitel, abdon, burgard}@cs.uni-freiburg.de

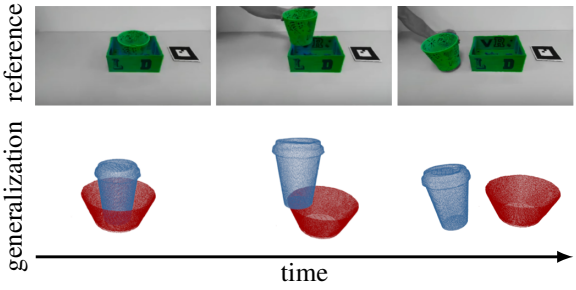


Fig. 1: The goal of our work is to enable a robot to imitate arbitrary spatial relations between pairs of objects and to generalize them to objects of different shapes and sizes. Top: three consecutive, arbitrary relations we presented our approach with, which we perceive using a Kinect2 camera. Bottom: the corresponding generalization of the relations using two new objects as computed by our approach.

Our deep metric learning approach allows the robot to learn rich representations of spatial relations directly from point cloud input and without the need for manual feature design. Furthermore, to generalize spatial relations in an end-to-end manner, we introduce a novel, gradient-descent based approach that leverages the learned distance metric to optimize the 3D poses of two objects in a scene in order to imitate an arbitrary relation between two other objects in a reference scene, see Fig. 2. For this, we backpropagate beyond the first convolution layer to optimize the translation and rotation of the object point clouds. Our gradient-based optimization enables the robot to imitate spatial relations based on visual demonstrations in an online and intuitive manner. In summary, we make the following contributions in this work: (1) an end-to-end approach to learning a metric for spatial relations from point clouds, (2) a differentiable projection to depth images to reduce the input dimensionality of point clouds, (3) a network architecture that models a differentiable metric function using a gradient approximation that allows for optimization beyond the first convolution layer, and (4) a demonstration that this technique enables gradient-based optimization in the learned feature space to optimize 3D translations and rotations of two new objects in order to generalize a demonstrated spatial relation.

II. Related Work

Learning spatial relations provides a robot with the necessary capability to carry out tasks that require understanding object interactions, such as object manipulation [3], human-

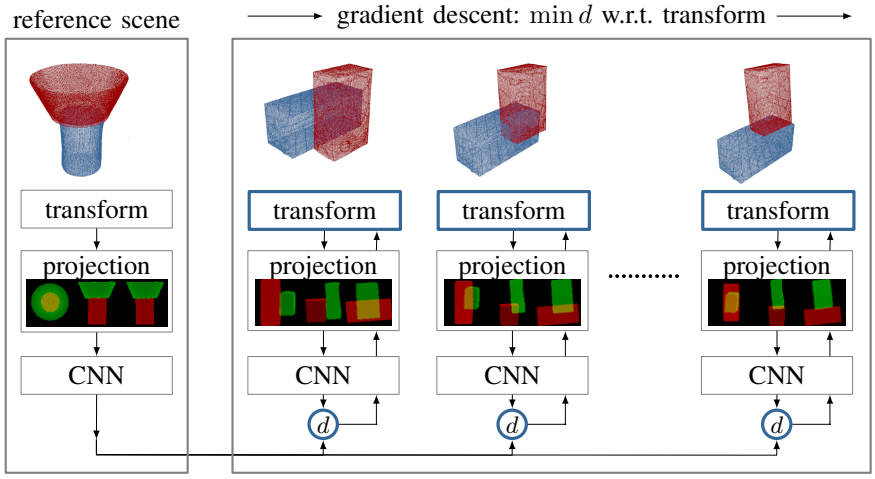


Fig. 2: An overview of our approach to generalize a relation. The transformation to the metric space consists of a function applying the 3D transformations, a projection of the point clouds to depth images, and a convolutional network pre-trained on pairs of relations. During test time, we backpropagate the error of the euclidean distance between the test scene's embeddings and the reference scene's embeddings, to optimize 3D translation and rotation of two objects to resemble the reference scene's spatial relations. Supplementary video: `http://spatialrelations.cs.uni-freiburg.de`

robot interaction [4], [5], [6] and active object search in complex environments [7]. In the context of robotics, learning spatial relations between objects has previously been phrased as a supervised classification problem based on handcrafted features such as contact points and relative poses [8], [9], [10]. Spatial relations can also be learned from human-robot interaction using active learning, where the robot queries a teacher in order to refine a spatial model [11]. However, the above techniques require learning an individual model for each relation and are thus limited in the number of relations they can handle. In contrast to these works, our metric learning approach allows us to reason about a continuous spectrum of known relations. Learning of object interactions from contact distributions has been addressed in the context of object grasping [12] and object placing [13]. While those approaches perform classification, our metric learning approach enables generation of spatial relations between objects solely based on visual information and without explicit modeling of contacts or physical object interaction. Visuospatial skill learning can imitate goal configurations with objects based on spatial relations, but the imitation does not generalize to objects of various sizes and shapes [14].

In our previous work we introduced a novel method that leverages large margin nearest neighbor metric learning in order to generalize spatial relations to new objects [15]. For this, we relied on hand-crafted 3D features to describe a scene. In contrast to this, we learn representations in an end-to-end fashion based on 2D image projections of scenes. Additionally, the previous work employed a grid search-based optimization with one object kept fixed whereas our approach optimizes on the full continuous spectrum of possible poses for both objects.

Our approach is related to deep learning techniques that learn similarity metrics directly from images, such as siamese [16] and triplet networks [17]. In comparison, our network takes point clouds as input and processes them using our differentiable point cloud to depth image projection layer for input dimensionality reduction. In addition, we leverage the gradient of the metric to optimize the translation and rotation of objects in the point cloud space. This strategy is in spirit similar to previous works that manipulate input images by backpropagating with respect to the input to visualize representations [18], and trick neural networks into making wrong classifications [19]. We explore and analyze the utility of the metric's gradient for optimization of 3D translation and rotation in Section V-B.

III. Problem Formulation

We aim to learn a continuous representation for spatial relations between everyday objects in the form of a metric function, i.e., a representation that is not restricted to a finite set of relations, and to use this metric to enable a robot to imitate these spatial relations with new objects. Our goal is to learn this representation directly from geometric information in the form of raw point clouds, without requiring any semantics and without relying on handcrafted features.
For this, we consider pairwise spatial relations between objects. We denote these objects by o_m and o_n and we represent them as point clouds $\mathbf{P}_m$ and $\mathbf{P}_n$. Together with the respective translation vectors $\mathbf{t}_m$, $\mathbf{t}_n$ expressed relative to a global world frame and rotation quaternions $\mathbf{q}_m$, $\mathbf{q}_n$, we define a scene $\mathbf{s}_i$ as the tuple $\mathbf{s}_i = \langle o_{m,i}, o_{n,i}, \mathbf{t}_{m,i}, \mathbf{t}_{n,i}, \mathbf{q}_{m,i}, \mathbf{q}_{n,i} \rangle$. As a reference frame, we assume that the gravity vector $\mathbf{g}$ is known and oriented in the opposite direction of the global z-axis.

To learn the metric, we require a set of training scenes $S = \{\mathbf{s}_0, ..., \mathbf{s}_n\}$ accompanied by labels in the form of a similarity matrix $\mathbf{Y}$ where the entry $\mathbf{Y}_{ij}$ denotes the similarity of the

spatial relation between scenes $\mathbf{s}_i \in S$ and $\mathbf{s}_j \in S$. That is $\mathbf{Y}_{ij}$ should be small for similar relations and large for dissimilar relations. Note that we do not require all possible scene combinations to be labeled, i.e., $\mathbf{Y}$ does not need to be fully specified. To ease labeling, we allow the entries of $\mathbf{Y}$ to be binary, i.e., $\mathbf{Y}_{ij} \in \{0, 1\}$, where 0 means similar and 1 dissimilar.

Our goal is to learn a metric function $f(\mathbf{s}_i, \mathbf{s}_j) = d$ that maps two scenes to a distance d such that the following properties hold: (1) d captures the similarity of the spatial relations depicted in scenes $\mathbf{s}_i$ and $\mathbf{s}_j$, that is d is small for similar relations and large for dissimilar relations, and (2) f is differentiable. The latter ensures that we can employ gradient based optimization on the metric function.

Instead of directly learning the metric function f, we learn a mapping function Γ that maps each input scene into a low-dimensional space such that the Euclidean distance captures the similarity of spatial relations. Concretely, we define our metric function f as

$$f(\mathbf{s}_i, \mathbf{s}_j) = ||\Gamma(\mathbf{s}_i) - \Gamma(\mathbf{s}_j)||_2. \tag{1}$$

As a smaller distance denotes higher similarity, we formulate the problem of generalizing a spatial relation in a reference scene $\mathbf{s}_r$ to a test scene $\mathbf{s}_t$ as finding the translations and rotations of both objects in $\mathbf{s}_t$ which minimize the distance between the two scenes under the learned metric, i.e., we seek to solve the following problem:

$$\underset{\mathbf{t}_{m,t}, \mathbf{t}_{n,t}, \mathbf{q}_{m,t}, \mathbf{q}_{n,t}}{\text{minimize}} \; f(\mathbf{s}_r, \mathbf{s}_t). \tag{2}$$

In this work, we focus on computing the poses of both objects to imitate the semantics of a reference relation, and do not consider the physical feasibility of the resulting scene, e.g., collision checks.

IV. Approach

Our method consists of two phases. In the first, we learn a distance metric function to capture the semantics of spatial relations. In the second, we leverage the gradient of this function to imitate spatial relations in a continuous manner via optimization of the object poses. The key challenges are to learn a rich representation from high dimensional input and to backpropagate the gradient information into the raw point cloud.

A. Distance Metric Learning: Metric Composition and Training

The goal of learning the distance metric is to express the similarity between spatial relations. As input we use high-dimensional 3D point cloud data.

Therefore we seek a mapping function Γ that reduces the dimensionality from the point cloud space to a low-dimensional metric space. We implement Γ as a composition of three functions, which we will now outline briefly and then describe two of the functions more detailed. First, the transformation function ψ applies the corresponding rotation $\mathbf{q}$ and translation $\mathbf{t}$ to each object point cloud. Second, we project the point clouds to three orthogonal planes to create three depth images. We denote this projection function by ρ. Third, we apply a mapping function $G_{\mathbf{W}}$ parameterized by $\mathbf{W}$, which maps the three projections to the metric space, see Fig. 2 for an overview. More formally, we compose the mapping Γ as

$$\Gamma \coloneqq G_{\mathbf{W}} \circ \rho \circ \psi. \tag{3}$$

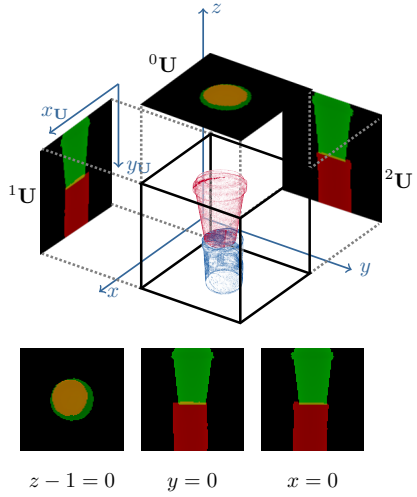


Fig. 3: Projections to three orthogonal planes. We project each point cloud to the three orthogonal planes defined by $y = 0$, $x = 0$, and $z - 1 = 0$. We create a depth image by setting the value of a pixel to the smallest distance of all points that are projected on this pixel multiplied by 100. To contrast the objects from the background we add a bias of 100 to all object pixels. Each projection of the two objects is in a separate channel and we randomly choose which object is positioned in the first channel. For this visualization, we added an all-zero third channel.

We tackle the dimensionality of the input data using a function ρ that projects the point clouds to three depth images. This serves as a non-parameterized reduction of the input dimensionality in comparison to 3D representations such as octrees or voxels. Concretely, we scale the scene to fit in a unit cube, see Fig. 3. We then project each point to three orthogonal image planes of size 100×100 pixels fit to the top, front, and side faces of the cube such that the image plane normals are either parallel or orthogonal to the gravity vector $\mathbf{g}$. We place the projection of each object in a separate channel. In this work, we will refer to a single orthogonal projection of one object as projection image.

To learn the parameters $\mathbf{W}$ of the mapping function $G_{\mathbf{W}}$ we use a triplet network [17], a variation of the siamese convolutional network that features three identical, weight-sharing networks $G_{\mathbf{W}}$. We train the network on input triplets of projections $\langle(\rho \circ \psi)(\mathbf{s}_i), (\rho \circ \psi)(\mathbf{s}_j^+), (\rho \circ \psi)(\mathbf{s}_k^-)\rangle$ where $\mathbf{s}_i \in S$ is a reference scene and $\mathbf{s}_j^+, \mathbf{s}_k^- \in S$ are similar and dissimilar to $\mathbf{s}_i$, respectively, that is, in the case of binary labels, $\mathbf{Y}_{ij} = 0$ and $\mathbf{Y}_{ik} = 1$. We run each plane

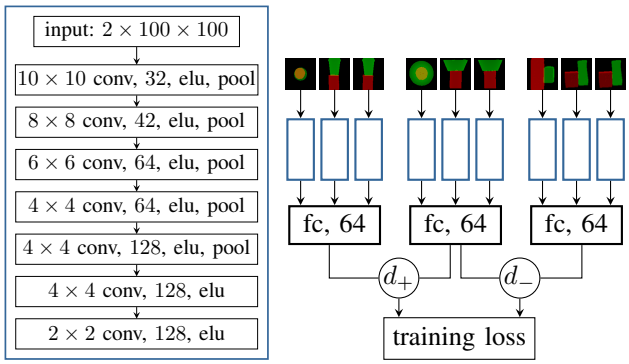


Fig. 4: The hyperparameters of the subnet of a sibling $G_{\mathbf{W}}$, found with random search. Each subnet receives one projection of one plane. The convolution layers of each subnetwork share the weights. All three subnets are fused with a fully connected layer. The sibling network $G_{\mathbf{W}}$ is then cloned three times into a triplet network when training the distance metric and cloned two times into a siamese network when generalizing a relation at test time.

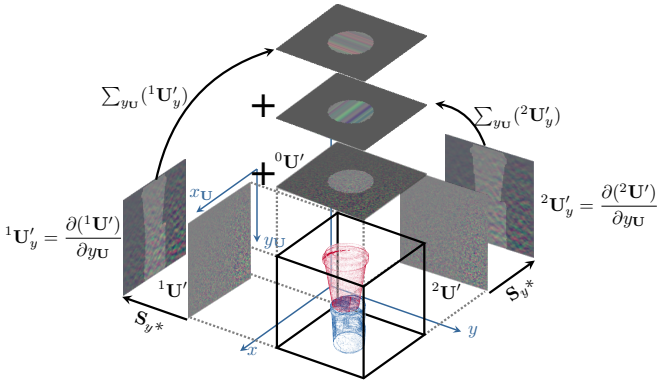


Fig. 5: Implementation of the partial derivative w.r.t. the world z-axis. For each projection we compute the partial derivatives w.r.t. pixels, $\mathbf{U}'$. For both projections ${}^1\mathbf{U}$,${}^2\mathbf{U}$ the world z-axis corresponds to the image axis $y_{\mathbf{U}}$. We compute the partial derivatives w.r.t. $y_{\mathbf{U}}$ by convolving ${}^1\mathbf{U}'$,${}^2\mathbf{U}'$ and the Sobel kernel $\mathbf{S}_y$. We then sum the resulting errors over the $y_{\mathbf{U}}$-axis and add them to the top projection, propagated over the axis of ${}^0\mathbf{U}$ that corresponds to the resepctive depth axis of ${}^1\mathbf{U}$,${}^2\mathbf{U}$. Then, for each pixel, we assign this error to the z-coordinate of the closest point the pixel in ${}^0\mathbf{U}$ originated from. Note that, assuming a solid point cloud with uniformly-distributed points, this is equivalent to the computation described in IV-A.

of the projection through its own sub-network with the sub-networks also sharing the weights and we fuse them with a fully-connected layer, see Fig. 4 for architecture details.

In contrast to an actual triplet network we do not employ a ranking loss but adapt the hinge loss function as in the approach by Chopra *et al.* to enforce an upper bound on the distance [1]. This upper bound ensures that the learning rate used for optimizing the poses of a test scene can be tuned independently of a specific set of learned parameters W. Concretely, we compute the loss function

$$C\big(\Gamma(\mathbf{s}),\Gamma(\mathbf{s}^+),\Gamma(\mathbf{s}^-)\big) = \frac{1}{2}(d_+)^2 + \frac{1}{2}\big(\max(0,1-d_-)\big)^2, \tag{4}$$

where $\Gamma(\mathbf{s})$ denotes the embedding of the scene $\mathbf{s}$, i.e., $\Gamma(\mathbf{s}) = G_{\mathbf{W}}\big((\rho\circ\psi)(\mathbf{s})\big)$, and d_+, d_- denote the Euclidean distance of $\Gamma(\mathbf{s}^+)$ and $\Gamma(\mathbf{s}^-)$ to the embedding $\Gamma(\mathbf{s})$ of the reference scene, respectively. During optimization, this results in d_+ being minimized towards 0 and d_- being maximized towards 1.

B. Generalizing Spatial Relations Using the Backward Pass

Having learned the neural network mapping function $G_{\mathbf{W}}$, we can now leverage the backpropagation algorithm to imitate a relation by optimizing the parameters of the 3D transformations applied to the point clouds. As stated in (2), we formulate the generalization of a spatial relation as a minimization problem with respect to the rotations and translations in the scene. Note that this differs from the representation learning process in that we keep the parameters $\mathbf{W}$ fixed and instead optimize the transformation parameters $\mathbf{t},\mathbf{q}$.

To employ gradient based optimization, the transformation Γ must be differentiable. While the functions $G_{\mathbf{W}}$ and ψ are differentiable, the gradient of the projection ρ needs a more thorough consideration. When backpropagating through the input layer of the first convolutional operation of $G_{\mathbf{W}}$, we need to consider that an input pixel not only contains depth information, but also discrete spatial information. Projecting a point onto the image plane discretizes two dimensions, which makes gradient-based optimization on these two axes impractical. Although projecting a scene to three sides sustains one continuous gradient for each axis, in our application the important information is contained in the location of the pixel, i.e., in the discretized dimensions.

As an example, consider a top-view projection image ${}^0\mathbf{U}$ of a 'cup on top of a can', i.e., a pixel value $u_{y,x}$ corresponds to the z-value of the point in the world frame, see Fig. 5. With only the depth information one cannot conclude if the cup is resting on the can or if it is hovering above it, as no information of the bottom of the cup is available. In contrast, the side view captures this information on the $y_{\mathbf{U}}$-axis of the image which also corresponds to the z-axis in the world coordinate frame. However, the gradient of the loss with respect to $y_{\mathbf{U}}$ is not well defined. The crux here is that the function $G_{\mathbf{W}}$, expressed as a convolutional neural network, only computes partial derivatives with respect to the input $\frac{\delta C}{\delta u_{y,x}}$, i.e., the partial derivative only depends on the magnitude $u_{y,x}$ but not on the position of a pixel. However, the hierarchical structure of $G_{\mathbf{W}}$ retains the spatial context of the error, which we will use to propagate the error of a single projection image back to all three coordinates of the 3D points.

Therefore, we convolve the matrix containing the error with respect to the input image of $G_{\mathbf{W}}$ with a Sobel-derived kernel. This procedure approximates the rate of change of

Method	3-out-of-5 acc.	5-out-of-5 acc.
LMNN [15]	$86.52\% \pm 1.98$	N/A
GBLMNN [15]	$87.6\% \pm 1.94$	N/A
Our NN-based metric	$\mathbf{91.21\% \pm 2.78\%}$	$76.25\% \pm 7.21\%$

TABLE I: Nearest neighbor performance on the Freiburg Spatial Relations dataset. We report results for correctly retrieving 3-out-of-5 and 5-out-of-5 target neighbors of a query scene.

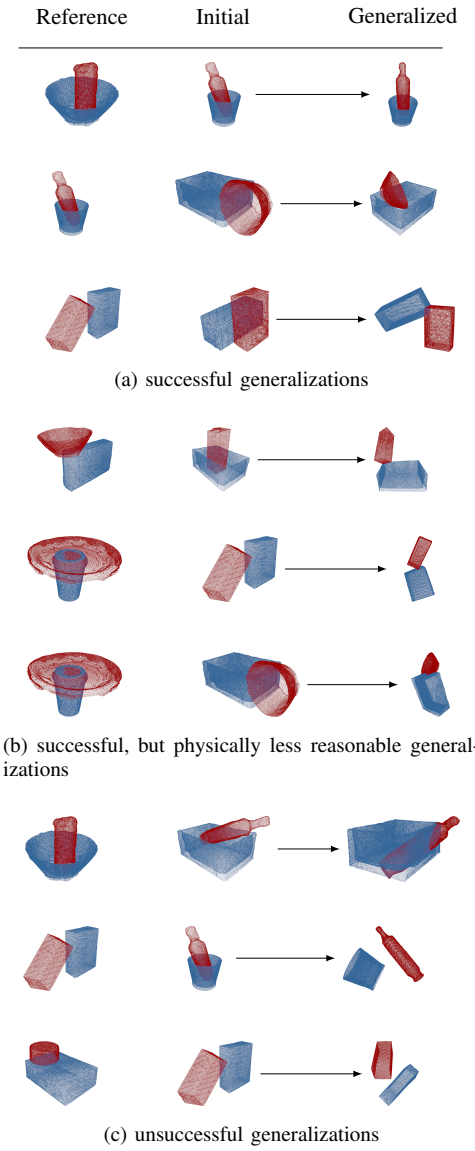
the error of the input image with respect to $x_{\mathbf{U}}$ and with respect to $y_{\mathbf{U}}$. That is, we approximate the change of the error with respect to shifting a pixel on the $x_{\mathbf{U}}$ and $y_{\mathbf{U}}$ axis and use this as the error of the respective spatial coordinates of the pixel. This error can then be backpropagated to the 3D point it resulted from via applying the inverse orthogonal projection and all three coordinates of the point are updated. More formally, for a projection image $\mathbf{U}$, i.e., one input channel of the function $G_{\mathbf{W}}$, the partial derivative can be formulated as follows. Let $\mathbf{U}'$ be the gradient of this projection image with respect to the loss where the entry $u'_{y,x} = \frac{\delta C}{\delta u_{y,x}}$ denotes the partial derivative of the loss with respect to the input pixel at position (x, y). We compute the matrices of partial derivatives with respect to the y and x position $\mathbf{U}'_y$ and $\mathbf{U}'_x$ as $\mathbf{U}'_y = \mathbf{S}_y * \mathbf{U}'$ and $\mathbf{U}'_x = \mathbf{S}_x * \mathbf{U}'$ with $\mathbf{S}_y, \mathbf{S}_x$ being the Sobel kernels $\mathbf{S}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ and $\mathbf{S}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$. In practice, we found that using a larger Sobel-derived kernel to approximate the derivatives achieves better results, which likely results from the gradients being sparse due to maxpooling. Once we have computed an error on the pixel $u_{y,x}$ with respect to $x_{\mathbf{U}}$ and $y_{\mathbf{U}}$ we assign the error values to the respective coordinates of all points whose projection would result in the pixel y, x, assuming the object is solid. For each point, we sum all the errors associated with it. In summary, this ensures that each coordinate of a point is assigned an error from each projection. The remaining partial derivatives for the translation and rotation have an analytical solution. Fig. 5 depicts an overview of the gradient implementation for a single axis of the points. Our code is available at `https://github.com/philjd/generalize_spatial_relations`.

V. Experimental Results

In this section we conduct several quantitative and qualitative experiments to benchmark the performance of our approach. Hereby, we demonstrate the following: 1) our learned feature representation is able to generalize over a rich set of different spatial relations and yields improved performance for a spatial nearest neighbor retrieval task with respect to a state-of-the-art method, 2) using our novel gradient-based optimization method we are able to generalize spatial relations to new objects and to capture the intention of the reference scenes being imitated without prior semantic knowledge about the relation they embody.

A. Nearest Neighbor Retrieval

We train the distance metric function on the Freiburg Spatial Relations dataset [15], which features 546 scenes each containing two out of 25 household objects. The dataset contains seven labeled relations, which we transform into a binary similarity matrix. As stated before, we train the metric using a triplet network. The network parameters $\mathbf{W}$ are optimized for $14,000$ iterations with a batch size of 100 triplets. We sample the batches such that the provided class annotations of the scenes are uniformly distributed. Further, we apply data augmentation in a way that does not change the underlying ground truth spatial relation, i.e., we add a small amount of noise on the transformations of the objects and rotate the full scene around the z axis. Additionally, we apply dropout with a probability of 50% on the fully-connected layer. For training we use



(a) successful generalizations

(b) successful, but physically less reasonable generalizations

(c) unsuccessful generalizations

Fig. 6: Examples for successful (Fig. 6a), successful but physically infeasible (Fig. 6b), and unsuccessful generalizations (Fig. 6c). In each row, the leftmost scene depicts the reference scene, the test scene before optimizing, and the rightmost scene depicts the generalized result.

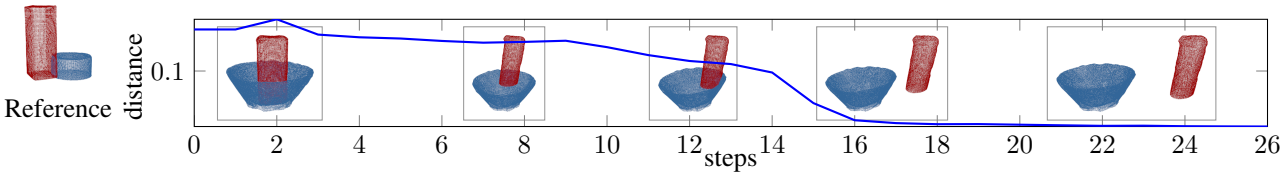


Fig. 7: This figure shows an example of an optimization run to imitate a 'next-to' relation in a reference scene consisting of a box and a can and to generalize it using two new objects: a bowl and a smaller box. We show the intermediate scenes obtained during the optimization and their corresponding distance to the reference scene. Despite the very different shapes, the produced relation resembles the spatial relation of the reference scene.

stochastic gradient descent with a momentum of 0.9 and warm restarts [20] with an initial learning rate of 0.001 and a period length of 1500 steps, which is doubled after each restart. We cross-validate the nearest neighbor performance of our learned metric on the 15 train/test splits of the Freiburg Spatial Relations Dataset provided with the dataset. As performance measure, we compute the mean 3-out-of-5 and 5-out-of-5 accuracy for nearest neighbor retrieval over the fifteen splits. Table I shows that our method yields an accuracy of $91.21\% \pm 2.78\%$, which is a relative improvement of 3.6% compared to the GBLMNN approach by Mees *et al.* [15]. Our results show that the learned metric allows us to retrieve similar scenes from a continuous spectrum of relations in the learned space with high accuracy. Further, they suggest that the learned feature representation of the metric, captured in the last fully-connected layer of the network siblings (see Fig. 4), is rich enough to be leveraged for gradient-based optimization in the next experiments.

B. Generalizing Relations to Known Objects

Next, we quantitatively evaluate the capability of our approach to imitate spatial relations. For testing we randomly selected 13 scenes including 15 different objects such that every scene was similar to at most one other scene. We then considered all 156 combinations of these 13 scenes excluding 31 scenes that cannot be transformed into each other, e.g., a plate and a cup cannot be generalized to an inside relation. From the remaining 125 combinations, we used one scene as a reference to generalize the other scene, as qualitatively depicted in Fig. 6. To compute the generalization, we used the Adam Optimizer with a learning rate of 0.1 to minimize the metric distance by optimizing $\mathbf{t}_m, \mathbf{t}_n$ and $\mathbf{q}_m, \mathbf{q}_n$ of the test scene. Overall, 70 of the imitations successfully generalized the reference scene. 41 of these imitated scenes were physically infeasible scenes, e.g., containing objects placed on their edges. However, we do not account for scene stability or feasibility in this work and therefore consider them successful. 55 of the generalizations converged to a non-optimal solution. Fig. 6 qualitatively depicts exemplary results for each. Among successful generalizations the figure shows a bowl that is correctly optimized into a tray and several inclined object relations. Note that both object transformations are optimized in a continuous manner, without specifying any semantic knowledge, see Fig. 7. Despite the fact that we do not provide knowledge about physical concepts such as collisions during the training process and despite the fact that we approximate the 3D world using our 2D projection technique, our approach is able to leverage the learned metric to generalize relations in the 3D domain.

In addition, we conducted a real-world experiment with a Kinect2 camera, where we demonstrated reference scenes containing common spatial relations, see Fig. 1. To retrieve the point clouds and the poses of the reference objects we used the Simtrack framework [21]. In this experiment we demonstrate the capability of our method to successfully generalize spatial relations in real-time. In a further experiment, we employed our framework on a real PR2 robot and used the robot to manipulate the objects of the test scene, using out-of-the-box motion planning. A video of these experiments is available at `http://spatialrelations.cs.uni-freiburg.de`.

C. Generalizing Relations to Unknown Objects

To evaluate how well our approach handles previously unseen objects, we chose five common 3D models such as the Utah tea pot and the Stanford bunny, as shown in Fig. 8. We emphasize that the shapes of these objects differ notably from the objects of the Freiburg Spatial Relations dataset used during training. As reference scenes we used a subset of the 13 previously used scenes. As test scenes we used all two-permutations without replacement of the five new objects, sampled next to each other. We then considered all 160 transformable combinations of training and test scenes, excluding the test object combinations that cannot form an inside relation. Our approach was able to successfully generalize 68 scenes and failed on 92 scenes. As expected, this is a more challenging task compared to the previous experiment since none of the test object shapes were used in training. Additionally, the dataset used for training covers only a small subset of the space of possible object arrangements, posing a challenge on dealing with intermediate relations the approach encounters during optimization. Nonetheless, our approach is able to imitate the semantics of a given spatial relation with considerably different objects and generalize them without the need for prior knowledge.

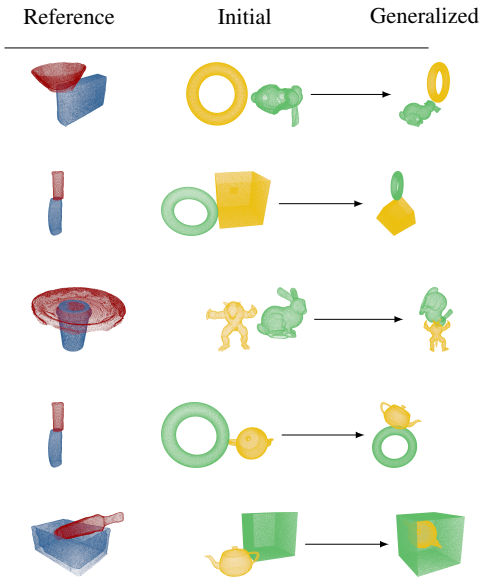


Fig. 8: Examples of four successfully and one unsuccessfully generalized relations to objects unseen during training. Despite the complex, unknown shapes, the optimized scenes capture the semantic of the relation. The last example is counted as unsuccessful because the opening of the cube should point upwards.

VI. CONCLUSIONS

In this paper, we presented a novel approach to learning the similarity between pairwise spatial relations in 3D space and to imitate arbitrary relations between objects. Our approach learns a metric that allows reasoning over a continuous spectrum of such relations. In this way, our work goes beyond the state of the art in that we do not require learning a model for each new spatial relation. Furthermore, our work enables learning the metric and using it to generalize relations in an end-to-end manner and without requiring pre-defined expert features. For this, we introduced a novel approach for backpropagating the gradient of the metric to optimize the 3D transformation parameters of two objects in a scene in order to imitate an arbitrary spatial relation between two other objects in a reference scene. We evaluated our approach extensively using both simulated and real-world data. Our results demonstrate the ability of our method to capture the similarities between relations and to generalize them to objects of arbitrary shapes and sizes, which is a crucial requirement for intelligent service robots to solve tasks in everyday environments. To incorporate physical constraints such as object collisions, it would be interesting to add differentiable physics in the future [22].

ACKNOWLEDGMENT

We thank Tobias Springenberg for fruitful discussion and feedback on leveraging 2D projections and Oier Mees for providing the data set.

REFERENCES

[1] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 539–546, IEEE, 2005.

[2] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 7, no. 04, pp. 669–688, 1993.

[3] K. Zampogiannis, Y. Yang, C. Fermüller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1389–1396, IEEE, 2015.

[4] S. Guadarrama, L. Riano, D. Golland, D. Go, Y. Jia, D. Klein, P. Abbeel, T. Darrell, *et al.*, "Grounding spatial relations for human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1640–1647, IEEE, 2013.

[5] M. Shridhar and D. Hsu, "Grounding spatio-semantic referring expressions for human-robot interaction," *arXiv preprint arXiv:1707.05720*, 2017.

[6] R. Schulz, "Collaborative robots learning spatial language for picking and placing objects on a table," in *Proceedings of the 5th International Conference on Human Agent Interaction*, pp. 329–333, ACM, 2017.

[7] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2818–2824, IEEE, 2011.

[8] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 11, pp. 1328–1342, 2011.

[9] K. Sjöö and P. Jensfelt, "Learning spatial relations from functional simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1513–1519, IEEE, 2011.

[10] S. Fichtl, A. McManus, W. Mustafa, D. Kraft, N. Krüger, and F. Guerin, "Learning spatial relationships from 3d vision using histograms," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 501–508, IEEE, 2014.

[11] J. Kulick, M. Toussaint, T. Lang, and M. Lopes, "Active learning for teaching a robot grounded relational symbols.," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[12] O. Kroemer, S. Leischnig, S. Luettgen, and J. Peters, "A kernel-based approach to learning contact distributions for robot manipulation tasks," *Autonomous Robots*, pp. 1–20, 2017.

[13] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1021–1043, 2012.

[14] S. R. Ahmadzadeh, F. Mastrogiovanni, and P. Kormushev, "Visuospatial skill learning for robots," *arXiv preprint arXiv:1706.00989*, 2017.

[15] O. Mees, N. Abdo, M. Mazuran, and W. Burgard, "Metric learning for generalizing spatial relations to new objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[16] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *IEEE International Conference on Computer Vision*, 2015.

[17] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1386–1393, IEEE, 2014.

[18] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[20] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, 2017.

[21] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1300–1307, IEEE, 2015.

[22] J. Degrave, M. Hermans, J. Dambre, *et al.*, "A differentiable physics engine for deep learning in robotics," *arXiv preprint arXiv:1611.01652*, 2016.