Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 3, 445–458
DOI: 10.61822/amcs-2026-0029

CONTACT MODELS IN ROBOTIC GRASP LEARNING

MAREK KOPICKI [a]

[a]Institute of Robotics and Machine Intelligence
Poznan University of Technology
ul. Piotrowo 3A, 60-965 Poznań, Poland
e-mail: marek.kopicki@put.poznan.pl

Autonomous robotic manipulation faces persistent challenges in dynamic, unstructured environments, where traditional physics-based models struggle with incomplete or inaccurate object and contact data. Machine learning has enabled adaptive manipulation skills, but purely end-to-end approaches often make contact-level reasoning implicit and rely on extensive training data. Structured methods that incorporate segmentation, depth, and planning offer complementary, more interpretable alternatives. Contact modelling plays a central role in predicting object motion and enabling realistic simulations, especially at close-range interactions. Early forward models demonstrated generalization to unseen objects, but struggled with action generation and exemplar dependence. Later advances introduced generative contact models that infer grasps directly from depth images, enabling one-shot learning and robust manipulation without prior object knowledge. This paper reviews and compares such contact modelling approaches, highlighting their role in grasp synthesis, generalization, and integration into dynamic and perception-driven manipulation pipelines.

Keywords: robotic grasping, contact models, machine learning.

1. Introduction

Autonomous robotic manipulation has emerged as a critical field in modern robotics, offering a wide range of applications but posing significant challenges for researchers and engineers for a long time (Sun *et al*., 2022). One of the core aspects of autonomous robotic manipulation is the ability to grasp and manipulate objects, also in a dynamic and unstructured environment. This capability is essential for a wide range of applications, such as pick-and-place and packing tasks in warehouses or assembly operations (Sun *et al*., 2022).

In an ideal world, one could use the laws of mechanics to predict the outcome of actions (Lynch and Mason, 1996) to select, e.g., stable grasps (Bicchi and Kumar, 2000). Even for rigid bodies, this would require models of the shapes, masses, mass distributions, and friction coefficients of the manipulated objects. It would also demand a model of the gripper kinematics and the contact forces that can be exerted in different configurations. The real world, however, is dynamic and unstructured, presenting significant challenges for robots to operate autonomously (Roy *et al*., 2021). Estimating the parameters of such models can be very difficult. Still model approximations can result in inaccurate predictions. Successful applications of robotic systems for manipulation tasks have largely been constrained to well-structured environments with pre-programmed behaviours or extensive training data available (Liao *et al*., 2017; Kroemer *et al*., 2020). To address these challenges, researchers have turned to *machine learning* techniques, which hold the promise of enabling robots to learn adaptive manipulation skills from experience (Roy *et al*., 2021).

Recent end-to-end methods, including diffusion and foundation-model approaches, can perform strongly in cluttered or large-scale settings when supported by large and diverse training sets (Ball *et al*., 2023; Ke *et al*., 2024; Jang *et al*., 2024). Their limitation does not lie in the fact that they cannot work, but that their predictions are often difficult to attribute to explicit contact hypotheses and the data or simulation burden remains high for new hands, objects or tasks. Structured information such as depth, segmentation and planning trajectories can complement such methods in autonomous bin picking (Zeng *et al*., 2019) or autonomous driving (Zhou *et al*., 2019).

1.1. Contact learning. Arguably, most manipulation tasks are about changing and/or maintaining kinematic relations between a robot, objects and object parts, where contacts and collisions between the objects involved are the key factors determining their behaviour, for both rigid and non-rigid objects (Mason, 2001; Battaglia *et al*., 2016). Accurate modelling of contacts is essential for realistic physics simulations, particularly at short distances, where objects are in close proximity (Mason, 2001; Andrews *et al*., 2021).

Our early work on *contact modelling and learning* was related to *prediction learning* of object motion during robotic manipulation (Kopicki *et al.*, 2011; 2017). The core idea was to use pairs of local reference frames at object-robot and object-environment contacts. The relative poses between these frames were represented as rigid-body transformations and used to create independent *forward models*, i.e., learned predictors of object pose changes. The object's motion was then predicted as the product of multiple local predictions generated by these forward models. We showed that the approach generalised to previously unseen objects and outperformed state-of-the-art methods, including physics simulators (Kopicki *et al*., 2017).

One of the major problems with this work on contact models was the requirement to use object exemplars to select appropriate forward models, which was difficult to achieve in unstructured environmental conditions. Another issue was the lack of possibility to directly generate manipulation actions, apart from expensive action planning by sampling the action results using forward models. In the work of Kopicki *et al*. (2014), novel contact models were introduced that can represent (almost) arbitrary robot-object contact relations and can be trained from just a single example. It is a *learned generative model of contacts* that allows inferring contacts directly from depth images, without any prior knowledge of the object exemplars or even image segmentation. This paper is a focused review of this line of work, from one-shot generative contact models to underactuated hands, evaluative models, dynamic grasping and active vision, with *robotic grasp learning* as the unifying application domain (Kopicki *et al.*, 2024; 2019; 2015; 2014; Arruda *et al.*, 2016; Aktaş *et al.*, 2022).

1.2. Contact models in the literature.

1.2.1. Modelling contacts. Analytical models, such as point contact with friction and soft contact formulations, dominate classical grasp analysis (Bicchi and Kumar, 2000). These rely on predefined contact types and force closure metrics but lack adaptability to novel geometries or noisy data. Physics-based simulation models (e.g., MuJoCo, Bullet) simulate contact dynamics using rigid-body physics and penalty methods (Todorov *et al*., 2012; Coumans, 2015). While useful for control and planning, they are computationally intensive and not inherently learnable or generalizable.

Our approach uses statistical learned contact models, encoding the relative pose between hand links and local object geometry. These models are probabilistic and modular, allowing flexible composition across grasp types (Kopicki *et al*., 2019).

1.2.2. Learning geometry. Deep learning approaches such as Dex-Net, GQCNN and PointNetGPD learn grasp success or grasp configurations from synthetic depth images, point sets and analytic metrics, and have shown robust performance under controlled sensing (Mahler *et al*., 2017; Liang *et al*., 2019). Their strength is scalable data-driven evaluation; for the questions addressed here, the limitation is that link-specific contact geometry is usually implicit and training relies on large generated datasets. Feature-based methods (e.g., FPFH, SHOT descriptors) encode surface geometry explicitly but require careful tuning and are sensitive to noise (Rusu *et al*., 2009; Tombari *et al*., 2010). They are often used in conjunction with clustering or matching algorithms. Our model learns local surface geometry implicitly through contact distributions, enabling generalization across object shapes and poses (Kopicki *et al*., 2019).

1.2.3. Learning from few or single examples. Few-shot learning methods have been explored in robotic manipulation, often focusing on imitation or task adaptation rather than explicit contact modelling. For example, task-embedded control networks enable few-shot imitation learning by embedding task representations into control policies (James *et al.*, 2018). Similarly, Gaussian process implicit surfaces offer shape estimation from sparse data, though they are rarely used for grasp synthesis directly (Dragiev *et al*., 2011).

Our model supports one-shot learning of contact models, enabling grasp transfer from a single demonstration. This is a major advantage in data-scarce or online learning scenarios, allowing rapid adaptation to novel objects without requiring large datasets or retraining (Kopicki *et al*., 2015).

1.2.4. Application to robotic grasping. Analytic models are used in grasp stability analysis and planning but require precise object models and contact assumptions (Bicchi and Kumar, 2000). Learning-based models often predict grasp pose or success directly from images or point clouds; uncertainty-aware and explainable variants improve diagnostics, but usually do not expose explicit link-specific contact models that can be composed and

reused (Mahler *et al*., 2017; Liang *et al*., 2019; Xie *et al*., 2023).

Our model is directly integrated into grasp generation and optimization pipelines, supporting both static and dynamic grasping. It enables query density evaluation, grasp type selection, and collision-aware refinement (Kopicki *et al*., 2019; Marturi *et al*., 2019).

1.2.5. Modularity and interpretability. End-to-end neural models can be powerful, but their internal representations are typically less modular, which can make grasp failures harder to adapt or debug (Liang *et al*., 2019; Xie *et al*., 2023). Hybrid approaches combining simulation with learning offer flexibility, although often with substantial tuning or data-generation effort (Todorov *et al*., 2012; Coumans, 2015).

Our framework is modular: contact models are link-specific, composable, and interpretable. This supports grasp type generalization and diagnostic evaluation (Kopicki *et al*., 2019).

2. Contact model learning

2.1. Robotic manipulation scenarios. The general approach is one of learning from demonstration (LfD), to learn a *generative model of contacts from demonstration*. This model of contacts can represent robotic grasps of fully actuated dexterous robotic hands, but also underactuated dexterous grippers (Kopicki *et al*., 2024) or simple parallel grippers (Marturi *et al*., 2019).

The grasping of novel objects is still an active research area. The scenario considered here is one in which a novel object must be grasped using typically a dexterous hand, after being seen from just a single viewpoint. This scenario is referred to as dexterous, single-view grasping of novel objects. This is essentially the grasping problem solved by humans and establishes a high bar for robots. The combination of scenario features means that it is hard to apply planning methods based on analytic mechanics. This is because they require knowledge of frictional coefficients, object mass, and complete object shape to evaluate a proposed grasp. None of these are either known a priori nor easily recovered even from multiple camera views.

Alternatively, there are learning methods. Broadly speaking, these are divided into those that learn *generative models* and those that learn *evaluative models*. The former take sensor data as input and produce one or more candidate grasps, while the latter take sensor data and a grasp candidate as input and produce an estimate of the quality of that grasp on the target object. We discuss here how to improve generative model learning, but also how to train evaluative models.

First, an example grasp is presented, and then a model of that grasp is learned. This grasp model

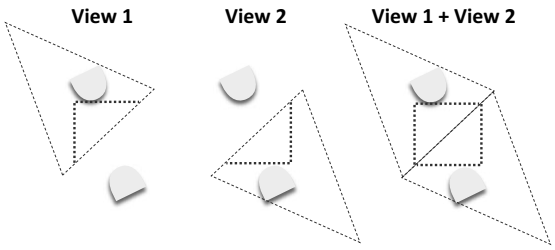


Fig. 1. Two-fingered grasp of an object, shown in cross-section. Object view models can be created from partial point clouds captured from single-camera view points (View 1 or View 2), but also from point clouds combined from multiple views (far right figure).

comprises two different types of sub-models—probability densities. The first one is a *contact-model* of the spatial relation between a hand-link and the local object shape at the point of contact. One contact-model is learned for every hand-link in contact with the object for every view of the grasp that observes that contact. Thus, if there is an average of N_L links in contact, there are N_G grasps, and each contact can be seen from N_V views, then there are $N_L \times N_G \times N_V$ contact models. The second type of model learned from each grasp example is a *hand-configuration one*, which captures the overall hand-shape. There is just one such model learned for each grasp, so that there are N_G in total.

The second stage of learning is clustering and then compression of the multiple contact models. The motivation is that many of these models are similar. First, they are clustered according to a similarity metric. Second, a single *contact model prototype* is created for each cluster. This results in a small set of more general contact models.

This grasp-view specific formulation was first presented by Kopicki *et al*. (2019) and is more general than the previous formulations (Kopicki *et al*., 2015). This is because it does not necessarily require point clouds captured from a single view, but simply those that are consistently constructed during both training and test (inference). This means that they could also be complete point clouds of objects as first used by Kopicki *et al*. (2014), or, e.g., point clouds that are merged (and, for instance, subsampled) from one or more different but fixed viewpoints (Fig. 1).

2.2. Representations.

2.2.1. Kernel density estimation. All contact models, query densities and hand configuration models are probability densities represented by *kernel density estimation*. A density is supported by particles, i.e., N weighted

samples x_j, each carrying a smoothing kernel $\mathcal{K}$:

$$\mathbf{pdf}(x) \simeq \sum_{j=1}^{N} w_j \mathcal{K}(x|x_j, \sigma), \tag{1}$$

where $w_j \in \mathbb{R}^+$, $\sum_j w_j = 1$, and σ denotes the bandwidth of the generic KDE expression. In the pose-descriptor kernels below, σ is written explicitly as a component-wise tuple. Throughout the paper, pose variables such as u, v and s denote individual rigid-body transformations, i.e., elements of $SE(3)$, usually parameterised as $x = (p, q)$ or $x = (p, q, r)$, with $p \in \mathbb{R}^3$, a unit quaternion $q \in S^3$ representing an element of $SO(3)$ up to antipodal equivalence, and optionally a surface descriptor $r \in \mathbb{R}^{N_r}$:

$$\mathcal{K}(x|\mu, \sigma) = \mathcal{N}_3(p|\mu_p, \sigma_p)\Theta(q|\mu_q, \sigma_q)\mathcal{N}_2(r|\mu_r, \sigma_r), \tag{2}$$

where μ and $\sigma = (\sigma_p, \sigma_q, \sigma_r)$ are respectively the kernel mean point and bandwidth tuple, $\mathcal{N}_n$ is an n-variate isotropic Gaussian kernel, and Θ corresponds to a pair of antipodal von Mises–Fisher, Gaussian-like distributions on S^3, representing rotations in $SO(3)$ (Kopicki *et al.*, 2014).

2.2.2. Object model and views. Let v denote a surface-frame pose and r the corresponding surface descriptor. There are N_G object-grasp training examples $g = 1 \dots N_G$, each observed from N_{V_g} *generalised views* $m = 1 \dots N_{V_g}$. A model of the grasped object from this view, denoted as $V_{m,g}(v, r)$, is computed from a single point cloud m as a set of $N_{m,g}$ features $\{(v_{j,m,g}, r_{j,m,g})\}$. This set defines, in turn, a joint probability distribution, which we call the *object-view model*:

$$\mathbf{pdf}^V_{m,g}(v, r) \equiv V_{m,g}(v, r) \simeq \sum_{j=1}^{N_{m,g}} w_{j,m,g} \mathcal{K}(v, r|x_{j,m,g}, \sigma_x), \tag{3}$$

where $\mathcal{K}$ is defined as in Eqn. (2). The grasp models are organised by view. At inference time this means that the grasp optimisation procedure will not try to force all hand-links which were in contact in the training grasp into contacts with visible surfaces, instead relying on the hand shape model to implicitly guide hand-links to hidden back surfaces.

2.3. Receptive field. We define a contact receptive field around each hand-link to quantify the relevance of nearby surface features for contact modelling (Kopicki *et al.*, 2014). It maps each surface feature pose v to a relevance weight:

$$F_i(v) \in [0, 1]. \tag{4}$$

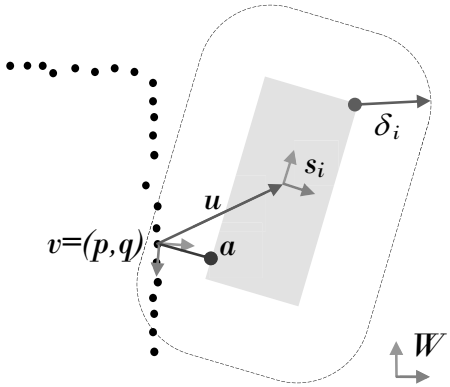


Fig. 2. Illustration of the contact receptive field F_i for hand-link i (light grey), showing surface samples (black dots), cutoff boundary δ_i (black oval), and relative pose u.

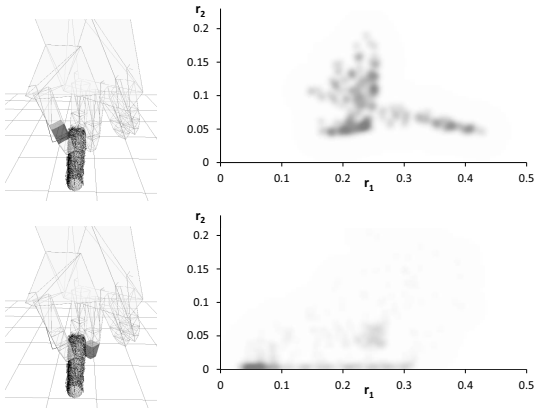


Fig. 3. Contact models by Kopicki *et al.* (2015). Left: a "pinch" training grasp with selected hand-links (dark grey), right: distributions of curvature descriptors r for features in each link's contact model.

The field is centred on the hand-link and assigns higher values to features closer to the link surface, decaying rapidly with distance (Fig. 2). This allows the model to focus on local geometry while ignoring distant, less relevant features. Various fast-decaying functions can be used to implement this behaviour.

2.4. Contact model density. *Contact model* $M_{i,m,g}(U, R)$ encodes the joint probability distribution of object surface features as "seen" in the frame of hand-link i, for view m and object-grasp example g. Upper-case symbols denote random variables and lower-case symbols their realisations. Thus R models descriptors and U the pose of link i relative to the feature frame, while $M_{i,m,g}(u, r)$ evaluates the density at sample (u, r). Contact models differ between links even within one grasp type, as shown by the marginalised models

$M_{i,m,g}(r)$ in Fig. 3.

The contact model for link i, view m and object g is estimated as

$$M_{i,m,g}(u,r) \simeq \frac{1}{Z}\sum_{j=1}^{N_{m,g}} F_i(v_{j,m,g})\mathcal{K}(u,r|u_{i,j,m,g},r_{j,m,g},\sigma), \quad (5)$$

where $Z > 0$ is a normalising constant, s_i is the pose of hand-link i, $u_{i,j,m,g} = v_{j,m,g}^{-1} \circ s_i$ with pose composition operation $\circ$, $\sigma = (\sigma_p, \sigma_q, \sigma_r)$, $N_{m,g}$ is the number of features in the point cloud, and $\mathcal{K}$ is the kernel from Eqn. (2) applied to the pose-descriptor variable (u, r).

Kopicki *et al*. (2019) introduced the idea of a *contact model norm*, a heuristic score of visible contact support for hand-link i in view m of grasp-object pair g:

$$\|M_{i,m,g}\| = \sum_{j=1}^{N_{m,g}} F_i(v_{j,m,g}). \quad (6)$$

Equation (6) is used in the contact model selection process (Section 2.5) to help estimate which links are reliably engaged in a particular grasp; it is not intended as a formal physical contact area or probability mass.

2.5. Contact model selection. In view-based learning, not all grasp-view-link combinations yield useful contact models—some links may have poor or occluded contact with the object. To address this, a two-phase selection procedure was introduced by Kopicki *et al*. (2019). The first phase uses the contact model norm to evaluate reliability, encoded as binary contact hypotheses:

$$b_{i,m,g} = \begin{cases} 1 & \text{if } \frac{N_L N_{V_g} N_G \|M_{i,m,g}\|}{\sum_{j,k,l}\|M_{j,k,l}\|} > \eta_i, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

In the denominator, j, k and l range over hand-links, views and grasp examples, respectively. The second phase retains grasp-view pairs with enough valid contact models, forming a non-empty index set $(i, m, g) \in \mathcal{M}$. Here, η_i is a link-specific threshold, λ and ζ are additional empirical selection thresholds, and $\mathcal{M}$ is the retained index set.

2.6. Contact model clustering. Memory-based contact models grow linearly with training data, limiting generalization and efficiency during transfer. To address this, contact models can be clustered and merged, as proposed by Kopicki *et al*. (2019), using a similarity measure between kernel density estimators. Each contact model is treated as a set of pose kernels, and the distance between two kernels is defined as a weighted sum of position and orientation differences:

$$d_{\text{k}}(x,y) = w_{\text{lin}} d_{\text{lin}}(p_x, p_y) + w_{\text{ang}} d_{\text{ang}}(q_x, q_y). \quad (8)$$

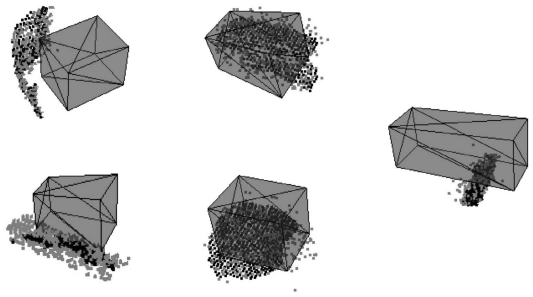


Fig. 4. Prototypes (dark grey) and members (light grey, black) of five contact model clusters after affinity propagation (Kopicki *et al*., 2019).

The constants w_{lin} and w_{ang} are fixed empirical scale factors balancing translational and angular distances. The distance from a kernel to a density is computed using only the nearest neighbor, $\min_y d_{\text{k}}(x, y)$, which improves efficiency and robustness to point cloud density variations. Kernel weights are ignored, as their influence is already captured by spatial proximity. Surface descriptors and full frame comparisons are also omitted, since shape information is already encoded in the kernel distribution.

To compare any two contact models, denoted here by M_a and M_b, an asymmetric divergence $d_{\text{dd}}(M_a, M_b) = \frac{1}{N_a}\sum_{x\in M_a} \min_{y\in M_b} d_{\text{k}}(x, y)$ is computed over their kernel centres. This is symmetrized using a max operator to yield a proper distance metric for clustering:

$$d(M_a, M_b) = \max\{d_{\text{dd}}(M_a, M_b), d_{\text{dd}}(M_b, M_a)\}. \quad (9)$$

Affinity propagation is then employed to cluster contact models and identify exemplars, forming a partition $\mathcal{C} = \{\mathcal{C}_1 \ldots \mathcal{C}_l \ldots \mathcal{C}_{N_\mathcal{C}}\}$. Each cluster groups similar models across grasps, views, and links, enabling more compact and generalizable representations.

2.7. Contact model prototypes. To improve generalization—especially for small clusters—contact model prototypes $M_l^{\mathcal{C}}$ are introduced to replace cluster exemplars (Kopicki *et al*., 2019). Each prototype is a mixture model over its cluster members, weighted by their similarity to the exemplar:

$$M_l^{\mathcal{C}}(u,r) = \sum_{M_{l,k}^{\mathcal{C}} \in \mathcal{C}_l} P(k|\mathcal{C}_l) M_{l,k}^{\mathcal{C}}(u,r). \quad (10)$$

Here, k indexes contact models in cluster $\mathcal{C}_l$. Weights are computed using an exponential function of the density distance, normalized to form a multinomial distribution. This allows efficient sampling and smooth interpolation across similar contact models. Visualizations (Figs. 4 and 5) show coherent, well-separated clusters, reflecting distinct link-to-surface contact types.

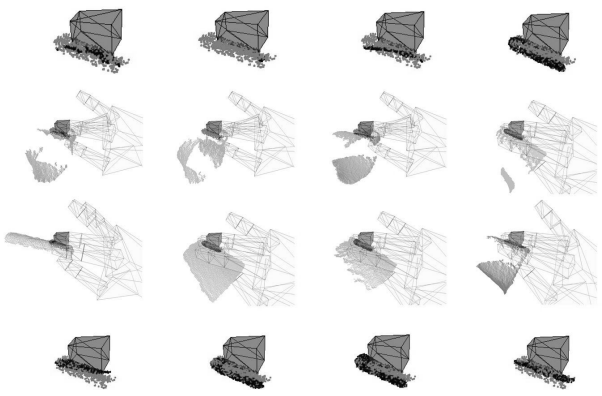


Fig. 5. Cluster members and point cloud data for one contact type (Kopicki *et al*., 2019). Prototype (dark grey), members (black), hand-link (light grey).

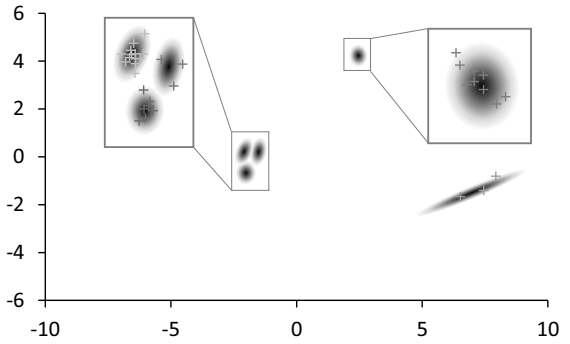


Fig. 6. Hand configuration model visualized via weighted PCA for five training grasps (Kopicki *et al*., 2015). Black density shows learned joint distributions while grey crosses indicate transferred grasps on test objects.

2.8. Hand configuration model. The hand configuration model C^g, introduced by Kopicki *et al*. (2014), encodes joint angle configurations specific to a grasp g, excluding wrist pose. It restricts the grasp search space during transfer to configurations near those observed in training. The model is learned from example grasps and captures both contact and approach phases. Configurations are interpolated and extrapolated between pre-contact (h_c^t) and contact (h_c^g) joint angles:

$$h_c(\gamma) = (1-\gamma)h_c^g + \gamma h_c^t, \tag{11}$$

where γ is the interpolation/extrapolation scalar. A kernel density estimate is built over these samples, weighted by proximity to h_c^g:

$$C(h_c) \equiv \sum_{\gamma\in[-\beta,\beta]} w(h_c(\gamma))\mathcal{N}_D(h_c|h_c(\gamma),\sigma_{h_c}). \tag{12}$$

Here, $\mathcal{N}_D$ is a D-variate Gaussian kernel, β sets the sampled range of γ, and σ_{h_c} is the configuration-space bandwidth. Learning complexity scales with the number of links, grasps, and interpolation steps. Figure 6 visualizes the learned models and transferred grasps using PCA-reduced joint data across five grasp types.

2.9. Kinaesthetic optimisation of contact models. While the approach for contact model learning presented so far works well with position-controlled robots, it cannot be directly applied to tendon-driven underactuated grippers, in particular with limited proprioception capabilities. This is because the relation between motor command and gripper posture can only be estimated experimentally and depends on many additional factors, such as the properties of the manipulated object. Real demonstrations allow the robot to learn motor commands and stable grips. However, the robot's links poses were not always accurately estimated, mainly due to sensor model limitations and imperfect camera calibration. As a result, some contact models were not created because the target object was too far away, or they were created, but colliding with the object.

The *kinaesthetic contact optimisation* procedure first presented by Kopicki *et al*. (2024) adjusts gripper joint configuration h_c and its 3D pose h_w w.r.t. the training object (Eqn. (3)), minimising collisions while maximising link-object contacts locally, in neighbourhood ϵ of a training grasp example $h^g \equiv (h_w^g, h_c^g)$:

$$\begin{aligned} h^{g*} &= \\ &\underset{h\in(h^g-\epsilon,h^g+\epsilon)}{\arg\max} (\mathcal{M}^g(h_w,h_c) - \zeta \sum_{i=1..D^h} W_i(h_w,h_c)) \\ \mathcal{M}^g(h_w,h_c) &= \sum_{i=1,\ldots,D^h} \sum_{j=1,\ldots,N^g} w_j^g F_i(v_j^g|h_w,h_c), \end{aligned} \tag{13}$$

where $h = (h_w, h_c)$, with wrist pose h_w and hand configuration h_c, D^h is the number of links, v_j^g is the j-th feature pose of training object g, $\mathcal{M}^g$ is the contact mass for grasp g, ζ is a collision weight, and W_i is the collision model (Eqn. (20)).

3. Inference with contact models

3.1. Contact query density. Contact query density defines a distribution over poses of a hand-link relative to a new object, guiding both grasp generation and evaluation. It is computed by convolving a contact model prototype with the object's surface model, encouraging contacts at regions with similar local geometry to those seen during training. Using the factorisation $M_l^{\mathcal{C}}(u,r) = M_l^{\mathcal{C}}(u|r)M_l^{\mathcal{C}}(r)$, the query density for link l is (Kopicki *et al*., 2015)

$$Q_l^{\mathcal{C}}(s) = \iiint T(s|u,v)V(v,r)M_l^{\mathcal{C}}(u|r)M_l^{\mathcal{C}}(r)\mathrm{d}v\mathrm{d}u\mathrm{d}r. \tag{14}$$

Here, $s = v \circ u$ is the world-frame pose of the link and $T(s|u,v)$ is a deterministic transformation kernel enforcing this pose composition, not a stochastic model. Furthermore, $M_l^{\mathcal{C}}(u|r)$ is the contact model prototype conditioned on curvature r and $V(v,r)$ is the object model. The term $M_l^{\mathcal{C}}(r)V(v,r)$ encourages the sampling of curvatures common to both the contact model and the object. To improve generalization, especially when contact models are merged, a more conservative formulation replaces $M_l^{\mathcal{C}}(r)$ with a likelihood over full surface patches:

$$P(N^{(v)}|s, M_l^{\mathcal{C}}) \propto \exp\left(-\phi d_{\text{dd}}(s \diamond M_l^{\mathcal{C}}, V)\right). \quad (15)$$

Here, $N^{(v)}$ is the local surface neighbourhood at v, ϕ is a penalty gain, and $s \diamond M_l^{\mathcal{C}}$ denotes $M_l^{\mathcal{C}}$ transformed by pose s. This term compares the transformed contact model to the local object surface. Substituting it into the integral yields the refined query density:

$$Q_l^{\mathcal{C}}(s) = \iiint T(s|u,v)V(v,r)M_l^{\mathcal{C}}(u|r)P(N^{(v)}|s, M_l^{\mathcal{C}})\mathrm{d}v\mathrm{d}u\mathrm{d}r. \quad (16)$$

Define candidate poses $\hat{s}_{lk} = \hat{v}_k \circ \hat{u}_{lk}$, which become kernel centres in the final density; in practice, this is approximated via importance sampling, where $\sim$ means "sampled from". Samples $(\hat{v}^k, \hat{r}^k) \sim V$, $\hat{u}_{lk} \sim M_l^{\mathcal{C}}(u|\hat{r}^k)$ define candidate poses $\hat{s}_{lk} = \hat{v}_k \circ \hat{u}_{lk}$, which become kernel centres in the final density:

$$Q_l^{\mathcal{C}}(s) \approx \sum_k w_{lk}\mathcal{N}_3(p|\hat{p}_{lk}, \sigma_p)\Theta(q|\hat{q}_{lk}, \sigma_q), \quad (17)$$

and weights such that

$$w_{lk} \propto \exp\left(-\phi d_{\text{dd}}(\hat{s}_{lk} \diamond M_l^{\mathcal{C}}, V)\right). \quad (18)$$

The hats denote sampled quantities and k indexes importance samples. This formulation ensures that the query density favours contact regions on the test object that closely resemble those seen in training, both in curvature and in full local geometry (Fig. 7).

3.2. Product of experts. Once query densities $Q_l^{\mathcal{C}}$ are created for each contact model prototype on a new object, an initial set of grasps is generated via random sampling (Kopicki *et al.*, 2019). A grasp-view pair (g,m) and hand-link i are sampled, pointing to a cluster $\mathcal{C}_l$ and its query density $Q_l^{\mathcal{C}}$. A link pose s_i is sampled from $Q_l^{\mathcal{C}}$ and a hand configuration h_c from the corresponding hand configuration model C^g. The sampled link pose and h_c determine h_w, yielding a full grasp $h = (h_w, h_c)$; repeating this gives seed grasps $\mathcal{H}^1 = \{h(j)\}$.

Grasp optimisation proceeds through K refinement steps, producing sets $\mathcal{H}^k$. The goal is to maximise the product of query densities and hand configuration likelihood:

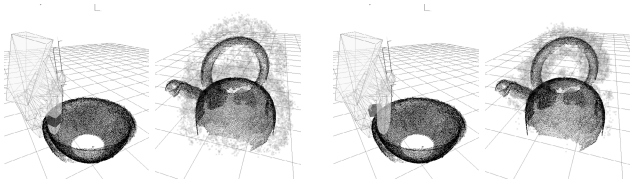


Fig. 7. Visualisation of two query densities (right image in each panel) for two contact models (left image in each panel) of a "pinch with support" grasp (Kopicki *et al.*, 2015). The links to which the contact models are associated are in dark grey. A query density is a distribution over poses of the corresponding link (light grey cloud) for a new query kettle.

$$\begin{aligned} h^* &= \arg\max_h \mathcal{L}^{gm}(h) \\ &= \arg\max_h \mathcal{L}_C^g(h)\mathcal{L}_Q^{gm}(h) \\ &= \arg\max_{(h_w,h_c)} C^g(h_c) \prod_{Q_{l(i)}^{\mathcal{C}} \in \mathcal{Q}^{gm}} Q_{l(i)}^{\mathcal{C}}\left(\text{kin}_i^{\text{for}}(h_w,h_c)\right). \end{aligned} \quad (19)$$

Here, $\mathcal{L}^{gm}$ is the full likelihood, $\mathcal{L}_C^g$ the hand-configuration factor, and $\mathcal{L}_Q^{gm}$ the query-density product, while $\text{kin}_i^{\text{for}}(h_w,h_c)$ is the direct-kinematics map from wrist pose and joint configuration to the pose of hand-link i in the object frame. Simulated annealing optimises grasps in $\mathcal{H}^k$, decreasing T from T_1 to T_K and retaining only the top 10% at selection steps.

During selection, a soft collision model $W(h_w,h_c)$ penalizes grasps that penetrate the object point cloud:

$$W(h_w,h_c) = \prod_{i=1,\ldots,D^h} \exp(-\gamma W_i(h_w,h_c)), \quad (20a)$$

$$W_i(h_w,h_c) = \sum_{j=1,\ldots,N} \left(\exp(\beta d_{W_i}(p_j,h_w,h_c)^2) - 1\right). \quad (20b)$$

Here, D^h is the number of hand links and N the number of object point-cloud points, p_j is one such point, $d_{W_i}(p_j,h_w,h_c)$ is the mesh-based penetration distance from p_j to the transformed geometry of link i under (h_w,h_c), and γ, β are empirical penalty gains. The overall grasp likelihood becomes

$$\begin{aligned} \mathcal{L}^{gm}(h) &= W(h_w,h_c)\mathcal{L}_C^g(h)\mathcal{L}_Q^{gm}(h) \\ &= W(h_w,h_c)C^g(h_c) \\ &\quad \prod_{Q_{l(i)}^{\mathcal{C}} \in \mathcal{Q}^{gm}} Q_{l(i)}^{\mathcal{C}}\left(\text{kin}_i^{\text{for}}(h_w,h_c)\right). \end{aligned} \quad (21)$$

This formulation evaluates grasp quality based on contact likelihood, hand configuration, and collision cost,

Table 1. Algorithmic variations.

A	Selection	Divergence	Clustering	Features
A1	No	No	No	Curv
A2	Yes	No	No	Curv
A3	Yes	Yes	No	Curv
A4	Yes	Yes	Yes	Curv
A5	Yes	Yes	No	FPFH
A6	Yes	Yes	Yes	FPFH

Table 2. Grasp success rates (succ) for algorithm variations A1 to A6 and training set T1 to T3 (A+T).

A+T	#Succ	%Succ	A+T	#Succ	%Succ
A1+T1	29	59.2	A4+T1	38	77.6
A6+T1	37	75.5	A1+T2	27	55.1
A2+T2	28	57.1	A3+T2	34	69.4
A4+T2	40	81.6	A5+T2	35	71.4
A6+T2	40	81.6	A1+T3	31	63.3
A4+T3	43	**87.8**			

enabling robust selection and refinement across diverse grasp types.

3.3. Evaluation. Extensive tests by Kopicki *et al*. (2019) evaluated grasp success rates using six algorithmic variants (A1–A6) and three training sets (T1–T3), all on the same platform as in the earlier work (Kopicki *et al*., 2015). The goal was to assess the impact of three innovations—contact model selection, query density divergence, and contact model clustering—plus the effect of increasing training data.

Training sets varied in grasp diversity and data volume:

- T1: five grasps from Kopicki *et al*. (2015),
- T2: 10 grasps with 7-view point clouds,
- T3: autonomous grasps generated during testing (leave-one-out regime).

Table 1 outlines algorithmic features and Table 2 reports success rates. The main outcome is that view-based modelling, divergence evaluation and clustering improve performance over baseline A1, especially when combined as A4/A6: success rises from 55.1% for A1+T2 to 81.6% for A4/A6+T2 and to 87.8% for A4+T3.

H1. Innovations improve success even with limited training; A4+T1 (77.6%) and A6+T1 (75.5%) outperform A1+T1 (59.2%), though not significantly.

H2. View-based modelling aids grasping thick objects; A2+T2 (68.75%) significantly outperforms A1+T2 (18.75%) on 16 deep-surface cases ($p = 0.0133$).

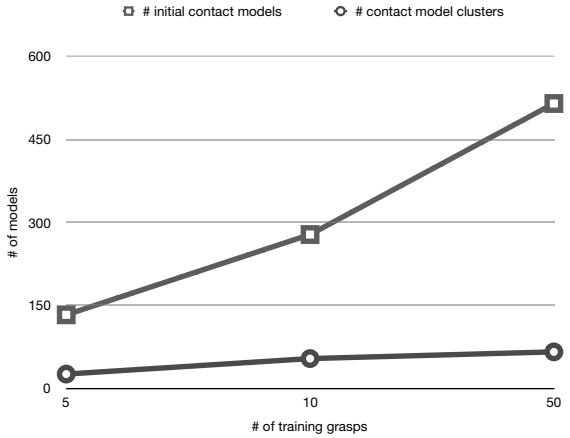


Fig. 8. Number of contact models before and after compression as the number of training grasps rises.

H3. Adding innovations improves performance; success rises from A1+T2 (55.1%) to A4+T2/A6+T2 (81.6%), with statistically significant gains.

H4. More training data improves performance; A4 rises from 77.6% (T1) to 87.8% (T3).

H5. Innovations better exploit added data; A4 improves by 10.2% from T1 to T3, compared to A1's 4.1%.

H6. Full innovation set dominates baseline; A4 and A6 outperform A1 across all training regimes, with highly significant differences.

Figure 8 shows that contact model clustering compresses the number of models as training data grows. This compression correlates with improved grasp success (e.g., A4 vs. A3: 81.6% vs. 69.4%).

In summary, combining selection, divergence, and clustering with richer training data yields substantial and statistically significant improvements in grasp success—up to 87.8% compared to 55.1% for the baseline. Figure 9 shows specific instances of grasps where this ability to grasp hidden back-surfaces of thick objects can be seen, such as the coke bottle, guttering, spray can and stapler.

3.3.1. Experiments with an underactuated hand. The kinaesthetic contact optimisation procedure (Section 2.9) was also validated experimentally by Kopicki *et al*. (2024) on the IKEA object set shown in Fig. 10. The robot was demonstrated with eight training trajectories, each labelled with the numbers 1 to 8 (Fig. 12). Each trajectory is different in terms of the gripper configuration and the local shape of the object, but also the motor commands that were used to control the gripper fingers.

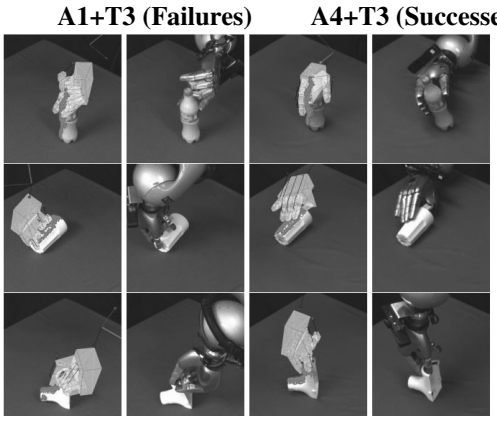


Fig. 9. Comparison of grasps executed by algorithms A1+T3 and A4+T3. These are three of the 15 cases where A1+T3 failed and A1+T3 succeeded. Columns are labelled by the variant.



Fig. 10. IKEA object sets used for training (left) and test (right).

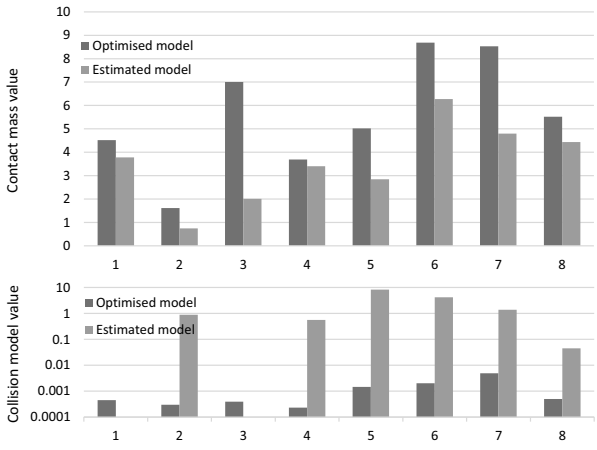


Fig. 11. Contact optimisation for grasp training examples.

Two sets of contact models were trained: *estimated models* based entirely on the estimated gripper configuration and *optimised models* based on kinaesthetic contact optimisation procedure (Section 2.9). Robot-object contacts for example grasps 5 and 6 are shown as grey clouds in Fig. 12(b) for estimated models and in Fig. 12(c) for optimised ones. Figure 11 shows that collision model values for the estimated model of grasp 5 are very high, which can also be seen in Fig. 12(b). Contact optimisation not only almost completely removes collisions (collision model value (Eqn. (20b)), but also increases the contact mass value for all training grasp examples (Fig. 11).

For tests we used 42 IKEA objects at semi-random poses on the table, creating 128 object–pose pairs. We compared estimated and optimised models, each time replicating the same object-pose pair but capturing the test point clouds independently. The robot performed 256 grasp trials fully autonomously, selecting the first most likely kinematically feasible grasp, with the table, robot, cage and cameras treated as obstacles. Experiments used the tabletop/lab hardware background and standard indoor lighting; no extra object occluders or shadow-removal stage were employed.

Reported runtimes should be read as representative research-code timings, not optimized benchmarks. In the one-shot experiments, thousands of grasps were generated in 12 to 23 s on an Intel i7-4960HQ, with query-density computation below 1 s; in the underactuated-hand experiments, query density, generation and optimisation together took about 5 s on an i7-11700 (Kopicki *et al.*, 2015; 2024). The code was optimized for modularity rather than speed, and the core sampling and evaluation steps are amenable to GPU acceleration.

Table 3 shows that optimized models with an 80% success rate perform significantly better than estimated contact models with a success rate of 57%. Grasps #2 and #4 have similar gripper configurations but different surface properties at contacts (rectangular vs. rounded). This affected corresponding query densities, so that grasp #2 was never selected and was dominated by the similar grasp #4, with a more common rounded local shape. Moreover, estimated models often chose pinch grasps #4 and #5 with very poor success rates. The optimized models significantly improved their reliability, but also reduced their selection probability.

We further validated our algorithm with optimized models on the YCB dataset (Calli *et al.*, 2015), with a success rate of 87% on 120 grasps following the protocol by Bekiroglu *et al.* (2019), who showed a 95% rate on this benchmark, although it relies on four views and a powerful industry-grade parallel-jaw gripper.

4. Other applications of contact models

4.1. Evaluative models. Grasp planning involves generating candidate grasps and estimating their success probability. Figure 13 shows a generative model (GM) proposing grasps and an evaluation model (EM) ranking them (Aktaş *et al.*, 2022). Three EM variants were analysed. All take a colour depth image and grasp trajectory parameters through separate channels; EM1 and

Table 3. Number of successful/total grasps by training examples 1–8 and overall, for estimated and optimised models.

Models	1	2	3	4	5	6	7	8	**All**
Estimated	5/7	–	29/37	7/25	2/19	14/16	6/8	10/16	**73/128**
Optimized	2/2	–	45/58	2/2	4/7	13/13	12/12	24/34	**102/128**
YCB+Optimized	–	–	68/81	–	–	21/22	11/11	4/6	**104/120**

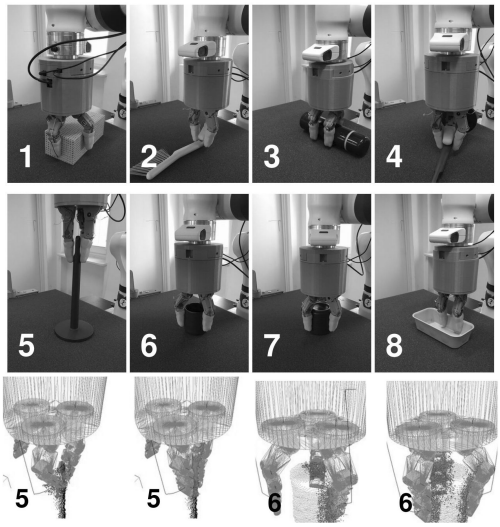


Fig. 12. Training examples 1–8 (top and middle rows), estimated and optimised contact models for training examples 5 and 6 (bottom row).

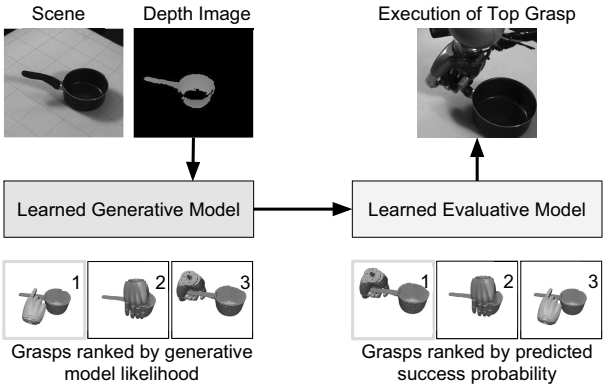


Fig. 13. Basic architecture of a generative-evaluative learner.

EM3 use VGG-16 while EM2 employs ResNet-50, and the outputs are combined by addition (EM1, EM2) or concatenation (EM3) before softmax success prediction. The architecture details are reported by Aktaş *et al.* (2022); here we focus on the evaluative models' role as evaluators of contact-model-generated grasp candidates.

Three EMs (EM1, EM2, EM3) were tested with two GMs: GM1 (A2+T2) and GM2 (A4+T2), where A2 and A4 are algorithms with the same T2 training sets (Kopicki *et al.*, 2019). Each EM processes the visual and trajectory data separately before combining them for the final prediction. To train the system, over 2.4 million simulated grasps were generated using domain randomization on 294 objects across 20 classes in MuJoCo (Todorov *et al.*, 2012). Success for each grasp was defined as lifting and holding the object above the table. The gains should not be attributed to a single factor. The EM benefits from a very large simulated dataset, while the contact-based GM contributes structured candidate grasps before neural ranking. This complementarity is suggested by GM2 alone outperforming GM1 (79.05% vs. 69.53%), by GM-EM combinations reaching 87.85%, and by real-world gains from 57.1% to 75.5% for GM1+EM2 and from 81.6% to 87.8% for GM2+EM2.

4.2. Dynamic trajectory planning for robotic grasping. While previous sections focused on static object manipulation, many real-world scenarios—such as human-robot handovers or grasping from moving platforms—require dynamic grasping capabilities. Marturi *et al.* (2019) introduced a five-phase algorithm that enables a robot to track and grasp arbitrarily moving objects in real time. This approach is particularly suited for collaborative tasks, such as a human handing an object to a robot, or manipulation in unstable environments like mobile platforms.

4.2.1. Grasp generation. Offline, a diverse set of grasp trajectories $\mathcal{T}_j^{v^w} = \{(v^w, c^h)_i\}_{i=1}^{m_j}$ is generated using learned contact and hand configuration models (Kopicki *et al.*, 2015). Each trajectory spans from a pre-grasp waypoint to a final contact pose. Only the top $\mathcal{N}$ hypotheses are retained.

4.2.2. Tracking. As the object moves, its pose s is continuously estimated via a particle filter-based tracker (Aldoma *et al.*, 2012). This updates trajectories to $\mathcal{T}_j^{sv^w} = \{(sv^w, c^h)_{ij}\}$, where sv^w is the object-updated wrist pose and c^h is the hand configuration.

4.2.3. Selection. Each updated trajectory is transformed into joint space $\mathcal{T}_j^{c^a} = \{(c^a, c^h)_{ij}\}$ by anytime inverse kinematics, where c^a is the arm configuration and c^h the hand configuration. Trajectories

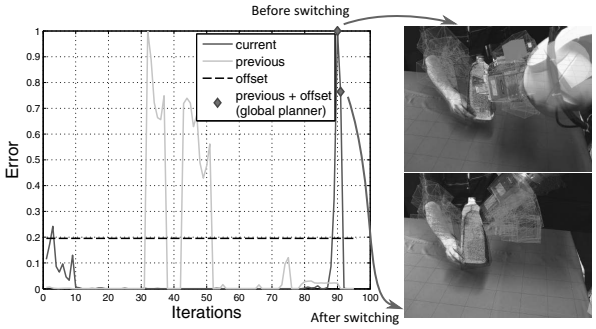


Fig. 14. Planner switching during dish handling. The curves show current and previous trajectory errors; switching occurs when error drops below threshold (dotted line). Diamonds mark switch points with corresponding robot poses.

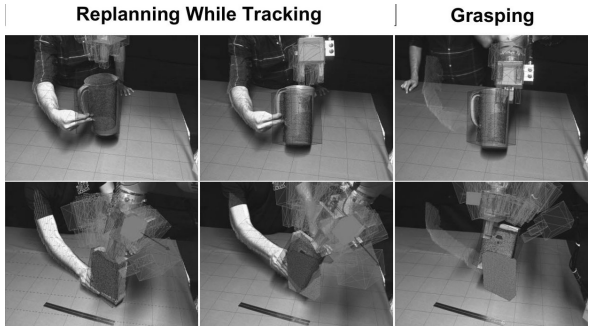


Fig. 15. Grasp adaptation examples. As objects move, grasp candidates are updated. The active grasp and next-best candidate are highlighted. The final column presents successful grasps.

are evaluated by the task space error,

$$\begin{aligned}\epsilon_{ij} = (1-a)\|\texttt{lin}(sv_{ij}^{w}) - \texttt{lin}(\mathcal{K}^{for}(\widetilde{c}_{ij}^{a}))\|^2 \\ + a(1 - |\texttt{ang}(sv_{ij}^{w}) \cdot \texttt{ang}(\mathcal{K}^{for}(\widetilde{c}_{ij}^{a}))^{-1}|).\end{aligned} \tag{22}$$

Here, `lin` and `ang` extract position and orientation, $\mathcal{K}^{for}$ stands for forward kinematics, a balances the two terms, and $\widetilde{c}_{ij}^{a}$ is the IK candidate. The best trajectory r is selected if its error ϵ_k improves over the previous by at least δ_w:

$$\begin{aligned}\epsilon_j &= \frac{1}{m_j}\sum_{i=1}^{m_j}\epsilon_{ij}, \\ \{r, \epsilon_r^*\} &= \begin{cases} k, \epsilon_k & \text{if } \epsilon_k + \delta_w < \epsilon_r^*, \\ r, \epsilon_r^* & \text{otherwise.}\end{cases}\end{aligned} \tag{23}$$

4.2.4. Planning. A tracking trajectory $\mathcal{T}_r^{c^c}$ is planned from the current configuration c^c to the approach waypoint c_{1r} using either a local or global planner:

$$\mathcal{T}_r^{c^c} = \begin{cases} \texttt{global}(c^c, c_{1r}) & \text{if } \|c^c - c_{1r}\|^2 > \delta_c, \\ \texttt{local}(c^c, c_{1r}) & \text{otherwise.}\end{cases} \tag{24}$$

4.2.5. Execution. A human operator triggers the final grasp execution. The selected trajectory $\mathcal{T}_r$ is guaranteed to have error $\epsilon_r^* \leq \epsilon_k + \delta_w$, ensuring feasibility.

4.2.6. Experiments. The method was validated through human–robot handover scenarios with diverse objects moving along arbitrary 6DoF paths. Results show high repeatability and robust adaptation via planner switching and trajectory selection (Marturi *et al*., 2019). Figures 14 and 15 illustrate dynamic grasp adaptation and successful executions.

4.3. Active vision for grasping. Grasp failures often stem from either unmodelled scene objects causing collisions or insufficient object reconstruction leading to unstable contact points. To address this, Arruda *et al*. (2016) proposed an active sensing approach that guides grasp planning through targeted view selection.

The robot begins with a single depth view from a fixed wrist-mounted camera (Fig. 16). It then selects additional views to improve the object's partial point cloud. A dexterous grasp planner (Kopicki *et al*., 2019) generates candidate grasps, assuming graspable surfaces on both sides of the visible geometry. Predicted contact locations guide the next view to improve point cloud quality at those regions. If no grasp is found, view selection switches to information gain planning using a 3D occupancy map. Once surface reconstruction is sufficient or a view limit is reached, the grasp is fixed. A second phase of active vision then verifies a safe trajectory to the grasp location by reducing entropy along the planned path using the same occupancy map. This ensures collision-free execution.

The proposed view selection method achieved an 85.7% success rate, outperforming a random view strategy (64.2%), which also produced more unsafe grasps under identical view constraints (Arruda *et al*., 2016).

5. Conclusions

This paper reviewed contact models as a structured alternative and complement to purely end-to-end grasp learning. The discussed methods model probabilistic relations between hand links and local object surfaces, combine them with hand configuration models, and use query densities for grasp generation and evaluation. Extensions to underactuated hands, deep evaluative models, dynamic grasping and active vision show that the same representation can support several stages of a

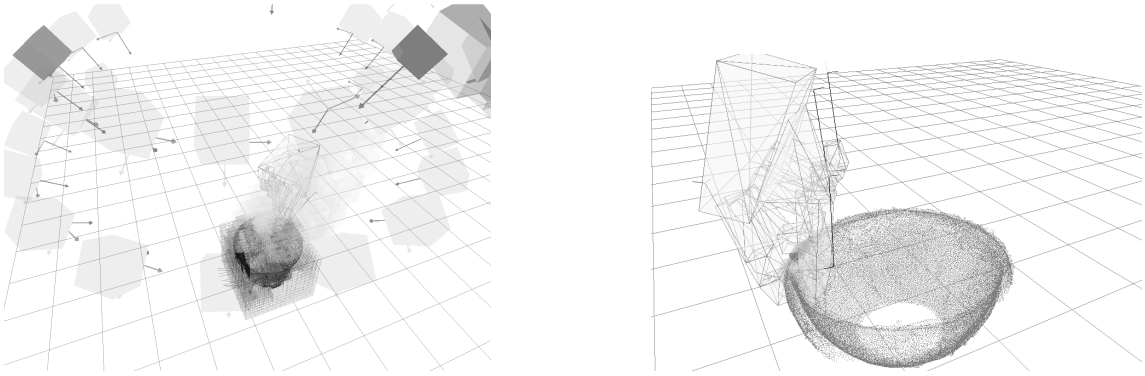


Fig. 16. Left: candidate camera poses Ξ, with visited poses V in darker grey; the voxel cube around the object is sculpted via information gain when no contacts are found. Right: example grasp with shaded contact regions; Active vision selects views to improve reconstruction at predicted contact sites.

manipulation pipeline. The empirical results indicate that contact model selection, clustering and divergence-based evaluation improve grasp success, while the comparison with deep evaluators suggests a complementary role: contact models generate interpretable, structured candidates, and neural models can further rank them when sufficient simulated data are available.

Acknowledgment

This work was supported by the Poznan University of Technology internal grant 0214/SBAD/0252 and the project no. 2021/43/P/ST6/01921, co-funded by the Polish National Science Centre and the European Union's *Horizon 2020* research and innovation programme under the Maria Skłodowska-Curie grant agreement no. 94533.

References

Aktaş, U.R., Zhao, C., Kopicki, M. and Wyatt, J.L. (2022). Deep dexterous grasping of novel objects from a single view, *International Journal of Humanoid Robotics* **19**(02): 2250011.

Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S. and Vincze, M. (2012). Point cloud library: Three-dimensional object recognition and 6 DoF pose estimation, *IEEE Robotics and Automation Magazine* **19**(3): 80–91, DOI: 10.1109/MRA.2012.2206675.

Andrews, S., Erleben, K. and Ferguson, Z. (2021). Contact and friction simulation for computer graphics, *ACM SIGGRAPH 2021 Courses*, Virtual Event, Article no. 2, pp. 1–124, DOI: 10.1145/3450508.3464571.

Arruda, E., Wyatt, J.L. and Kopicki, M. (2016). Active vision for dexterous grasping of novel objects, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea*, pp. 2881–2888, DOI: 10.1109/IROS.2016.7759446.

Ball, P.J., Smith, L., Kostrikov, I. and Levine, S. (2023). Efficient online reinforcement learning with offline data, *Proceedings of the 40th International Conference on Machine Learning, Honolulu, USA,* pp. 1577–1594.

Battaglia, P.W., Pascanu, R., Lai, M., Rezende, D. and Kavukcuoglu, K., (2016). Interaction networks for learning about objects, relations and physics, Available online at `https://arxiv.org/abs/1612.00222`.

Bekiroglu, Y., Marturi, N., Roa, M.A., Adjigble, K.J.M., Pardi, T., Grimm, C., Balasubramanian, R., Hang, K. and Stolkin, R. (2019). Benchmarking protocol for grasp planning algorithms, *IEEE Robotics and Automation Letters* **5**(2): 315–322, DOI: 10.1109/LRA.2019.2956411.

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: A review, *IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA,* Vol. 1, pp. 348–353, DOI: 10.1109/ROBOT.2000.844081.

Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P. and Dollar, A.M. (2015). The YCB object and model set: Towards common benchmarks for manipulation research, *2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey*, pp. 510–517, DOI: 10.1109/ICAR.2015.7251504.

Coumans, E. (2015). Bullet physics simulation, *ACM SIGGRAPH 2015 Courses, Los Angeles, CA, USA*, Article no. 7, pp. 7:1–7:1, DOI: 10.1145/2776880.2792704.

Dragiev, M., Toussaint, M. and Gienger, M. (2011). Gaussian process implicit surfaces for shape estimation and grasping, *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pp. 2845–2850, DOI: 10.1109/ICRA.2011.5980395.

James, S., Bloesch, M. and Davison, A.J. (2018). Task-embedded control networks for few-shot imitation learning, *Conference on Robot Learning (CoRL), Zurich, Switzerland*, pp. 783–795.

Jang, D.-S., Cho, D.-H., Lee, W.-C., Ryu, S.-K., Jeong, B., Hong, M., Jung, M., Kim, M., Lee, M., Lee, S. and Choi, H.-L. (2024). Unlocking robotic autonomy: A survey on

the applications of foundation models, *International Journal of Control, Automation and Systems* **22**(8): 2341–2384, DOI: 10.1007/s12555-024-0438-7.

Ke, T.-W., Gkanatsios, N. and Fragkiadaki, K. (2024). 3D diffuser actor: Policy diffusion with 3D scene representations, Available online at https://arxiv.org/abs/2402.10885.

Kopicki, M., Ansary, S.I., Tolomei, S., Angelini, F., Garabini, M. and Skrzypczynski, P. (2024). Underactuated dexterous robotic grasping with reconfigurable passive joints, *IEEE Robotics and Automation Letters* **10**(1): 48–55, DOI: 10.1109/LRA.2024.3497714.

Kopicki, M., Belter, D. and Wyatt, J.L. (2019). Learning better generative models for dexterous, single-view grasping of novel objects, *International Journal of Robotics Research* **38**(10–11): 1246–1267.

Kopicki, M., Detry, R., Adjigble, M., Stolkin, R., Leonardis, A. and Wyatt, J.L. (2015). One-shot learning and generation of dexterous grasps for novel objects, *International Journal of Robotics Research* **35**(8): 959–976, DOI: 10.1177/0278364915594244.

Kopicki, M., Detry, R., Schmidt, F., Borst, C., Stolkin, R. and Wyatt, J.L. (2014). Learning dexterous grasps that generalise to novel objects by combining hand and contact models, *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, pp. 5358–5365, DOI: 10.1109/ICRA.2014.6907647.

Kopicki, M., Zurek, S., Stolkin, R., Morwald, T. and Wyatt, J.L. (2011). Learning to predict how rigid objects behave under simple manipulation, *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pp. 5722–5729, DOI: 10.1109/ICRA.2011.5980295.

Kopicki, M., Zurek, S., Stolkin, R., Mörwald, T. and Wyatt, J.L. (2017). Learning modular and transferable forward models of the motions of push manipulated objects, *Autonomous Robots* **41**(5): 1061–1082.

Kroemer, O., Niekum, S. and Konidaris, G. (2020). A review of robot learning for manipulation: Challenges, representations, and algorithms, Available online at https://arxiv.org/abs/1907.03146.

Liang, H., Ma, X., Li, S., Görner, M., Tang, S., Fang, B., Sun, F. and Zhang, J. (2019). PointNetGPD: Detecting grasp configurations from point sets, *IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, pp. 3623–3629, DOI: 10.1109/ICRA.2019.8794435.

Liao, Y., Zhu, H., Wu, H., Zhang, G. and Guan, Y. (2017). Real-time safe motion generation through dynamical system modulation with multiple depth sensors, *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China*, pp. 44–49, DOI: 10.1109/ROBIO.2017.8324392.

Lynch, K.M. and Mason, M.T. (1996). Stable pushing: Mechanics, controllability, and planning, *International Journal of Robotics Research* **15**(6): 533–556.

Mahler, J., Matl, M., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Aparicio, J. and Goldberg, K. (2017). Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, *Robotics: Science and Systems XIII, Cambridge, USA*, DOI: 10.15607/RSS.2017.XIII.058.

Marturi, N., Kopicki, M., Rastegarpanah, A., Rajasekaran, V., Adjigble, M., Stolkin, R., Leonardis, A. and Bekiroglu, Y. (2019). Dynamic grasp and trajectory planning for moving objects, *Autonomous Robots* **43**(5): 1241–1256, DOI: 10.1007/s10514-018-9799-1.

Mason, M.T. (2001). *Mechanics of Robotic Manipulation*, MIT Press, Cambridge.

Roy, N., Posner, I., Barfoot, T., Beaudoin, P., Bengio, Y., Bohg, J., Brock, O., Depatie, I., Fox, D., Koditschek, D., Lozano-Perez, T., Mansinghka, V., Pal, C., Richards, B., Sadigh, D., Schaal, S., Sukhatme, G., Therien, D., Toussaint, M. and Van de Panne, M. (2021). From machine learning to robotics: Challenges and opportunities for embodied intelligence, Available online at https://arxiv.org/abs/2110.15245.

Rusu, R., Blodow, N. and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration, *IEEE International Conference on Robotics and Automation, Kobe, Japan*, pp. 2155–2162, DOI: 10.1109/ROBOT.2009.5152473.

Sun, Y., Falco, J., Roa, M.A. and Calli, B. (2022). Research challenges and progress in robotic grasping and manipulation competitions, *IEEE Robotics and Automation Letters* **7**(2): 874–881.

Todorov, E., Erez, T. and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal*, pp. 5026–5033, DOI: 10.1109/IROS.2012.6386109.

Tombari, F., Salti, S. and Di Stefano, L. (2010). SHOT: Unique signatures of histograms for surface and texture description, *Computer Vision and Pattern Recognition (CVPR), San Francisco, USA*, pp. 57–64.

Xie, Z., Liang, X. and Canale, R. (2023). Learning-based robotic grasping: A review, *Frontiers in Robotics and AI* **10**, Article no. 1038658, DOI: 10.3389/frobt.2023.1038658.

Zeng, A., Song, S., Yu, K.-T., Donlon, E., Hogan, F.R., Bauza, M., Ma, D., Taylor, O., Liu, M., Romo, E., Fazeli, N., Alet, F., Chavan Dafle, N., Holladay, R., Morona, I., Nair, P.Q., Green, D., Taylor, I., Liu, W., Funkhouser, T. and Rodriguez, A. (2019). Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching, *International Journal of Robotics Research***41**(7): 690–705, DOI: 10.1177/0278364919868017.

Zhou, B., Krähenbühl, P. and Koltun, V. (2019). Does computer vision matter for action?, *Science Robotics* **4**(30), Article ID: eaaw6661, DOI: 10.1126/scirobotics.aaw6661.



Marek Kopicki is an assistant professor at the Poznan University of Technology, specializing in machine learning for robotic manipulation, with a focus on perception, contact modeling, and planning. He holds a PhD in computer science from the University of Birmingham. With over two decades of experience in both academia and industry, Dr. Kopicki has contributed to algorithmic and hardware innovation in robotic systems. He has held a senior research role at Dyson Ltd., leading work on robotic manipulation technologies, and has collaborated extensively with industrial partners through EU-funded projects. He has also played key roles in several major European robotics initiatives, including CogX, GeRT (with DLR), PaCMan, and RoMaNS, and has recently served as the PI of the MSCA COFUND ProRoc project under *Horizon 2020*. He is the author of a software framework for robot control, motion planning, manipulation learning, and point cloud processing, used in several EU projects.

Received: 30 October 2025
Revised: 6 May 2026
Re-revised: 30 May 2026
Accepted: 2 June 2026