Learning-based Real-time Torque Prediction for Grasping Unknown Objects with a Multi-Fingered Hand

Dominik Winkelbauer[1,2], Berthold Bäuml[1,3], Rudolph Triebel[1,2]

***Abstract*— When grasping objects with a multi-finger hand, it is crucial for the grasp stability to apply the correct torques at each joint so that external forces are countered. Most current systems use simple heuristics instead of modeling the required torque correctly. Instead, we propose a learning-based approach that is able to predict torques for grasps on unknown objects in real-time. The neural network, trained end-to-end using supervised learning, is shown to predict torques that are more efficient, and the objects are held with less involuntary movement compared to all tested heuristic baselines. Specifically, for $90\,\%$ of the grasps the translational deviation of the object is below $2.9\,\mathrm{mm}$ and the rotational below $3.1°$. To generate training data, we formulate the analytical computation of torques as an optimization problem and handle the indeterminacy of multi-contacts using an elastic model. We further show that the network generalizes to predict torques for unknown objects on the real robot system with an inference time of $1.5\,\mathrm{ms}$. Website: dlr-alr.github.io/grasping/**

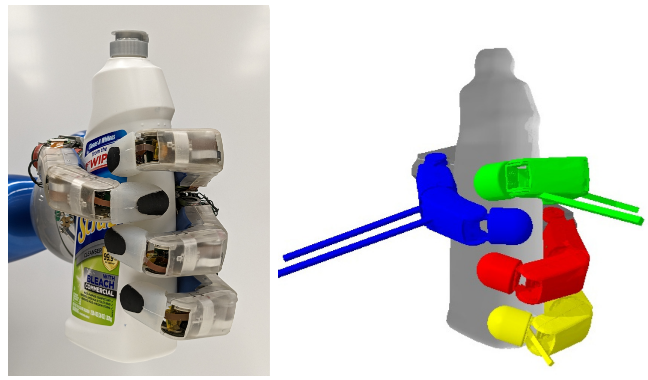


Fig. 1. A grasp on the YCB bleach bottle performed using the torques predicted by our learning-based approach. The scale and negative direction of the forces that are applied via each finger are visualized via colored lines in the right image together with the shape completed 3D model.

I. Introduction

Data-driven grasping allows planning grasps for unknown objects based on incomplete observations. While many approaches target parallel jaw grippers, also approaches applicable to multi-finger hands become more prominent. Usually, the grasp planner only determines the configuration of the grasp, meaning the pose of the hand base and the joint angles. However, to actually perform a grasp, it is also necessary to set at each joint the torque that should be applied. Especially for multi-finger hands, the correct setting of these torques is crucial for the stability of the grasp (see Fig. 1). A wrong set of torques can lead to involuntary shifting or rotation of the object or even to dropping it. Additionally, by choosing suitable torques for a given situation, unnecessary high torques for light objects can be reduced. This leads to less energy consumption and less strain on the hand and object. Nevertheless, most grasping approaches neglect this topic and just use standard torque settings across all objects.

There exist methods that are able to find a set of torques that balance the external wrench for a given grasp. Most of them target precision grasps which require that each finger touch the object in at most one contact. However, we do not want to restrict the space of possible grasps and also target power grasps which have many interlinked hand object contacts. Approaches that allow such power grasps, either linearize the problem or require high amounts of computation which make them unsuitable for real-time applications on the real robot. Also, they require precise information about contact positions and normals which are not available when grasping unknown objects.

In this work, we propose a learning-based approach to counter these issues. Similar to how learning-based methods allowed predicting grasps on unknown objects in real-time, we train a neural network to directly learn to predict the right torques for a given grasping situation. The input to the model consists of the hand pose, joint configuration, magnitude of the external wrench[1] and the observation of the object. We show that our network is able to provide torques that are efficient and hold the object still, which makes it superior to the tested baseline heuristics. To do so, the network does not need any detailed contact information, but only makes use of the global shape of the object. The prediction only requires one forward pass which makes our approach suitable for real-time applications.

II. Related Work

A. Torque Optimization

Calculating the required torques to resist a given external force has been intensely studied. Early approaches simplify the optimization problem by linearizing the friction constraint [1]. Buss et al. [2] found that the non-linear constraints are equivalent to the positive definiteness of a specific matrix. Han et al. [3] reformulated positive definiteness into linear

[1]DLR Institute of Robotics & Mechatronics, Germany
[2]Technical University of Munich (TUM), Germany
[3]Deggendorf Institute of Technology (DIT), Germany
Contact: Dominik.Winkelbauer@dlr.de

*This work is supported by the Helmholtz Association under the joint research school “Munich School for Data Science - MUDS“.

[1]The external wrench can either be given or can be estimated by slowly lifting the grasped object using a torque-controlled arm. For simplicity, we further use a conservative friction coefficient that covers all our scenarios.

 DOI: 10.1109/IROS55552.2023.10341970

matrix inequalities which make GFO a convex optimization problem that can be solved in polynomial time. To further speed up the optimization process, Zheng and Qian [4] proposes to split up the computation in an offline and an online part. However, all of the approaches mentioned above only support precision grasps.

In power grasps, there can be multiple contacts on one finger or even on one link. Therefore, as shown by Bicchi [5], contact forces cannot be individually controlled anymore. To solve this issue, Bicchi [5] proposed to build an elastic model of the grasp by placing springs in all contacts. This concept was later extended to support postural synergies [6] and underactuated hands [7]. However, Haas-Heger and Ciocarlie [8] pointed out that the linear formulations used in the aforementioned approaches lead to physically implausible solutions. Haas-Heger and Ciocarlie [8] therefore proposed to formulate the problem as a Mixed Integer Program that takes the non-linearities into account. However, this formulation becomes intractable for higher number of contact points. We propose to use a non-linear evolutionary optimizer in combination with a root finder instead. In this way, we are able to handle a high number of contact points and found solutions are always valid with respect to the full non-linear constraints.

B. Torque setting in data-driven grasping

Recently, many learning-based approaches have been developed to efficiently predict stable grasps for unknown objects. However, most of them use a simple heuristic to determine the forces to apply when executing the grasp. Most approaches close each finger with the same force using a position or an impedance controller [9, 10, 11, 12] Many methods also do not describe at all in which way they determine the grasp forces [13, 14, 15, 16, 17], leading to the perception that there is not much focus put on force control when grasping unknown objects. This stands in contrast to the observation that more sophisticated force-control strategies might lead to better grasping performance [11]. Based on that observation, we propose to set the torques based on the given grasp and object to exactly counter the external wrench. As computing these torques online is impossible due to missing contact information and high computational demands, we propose a learning-based approach that directly predicts the required torque at each joint in real-time.

There also exist methods based on reinforcement learning, which learn jointly the positioning of the hand and the forces to apply [12]. While this allows more advanced force control strategies, reinforcement learning-based approaches need well-calibrated simulator and need more training time, as the network also needs to master motion planning. Furthermore, the training might get stuck in local minima which leads to sub-optimal grasp positions and torque configurations

III. Torque Optimization

Given a grasp on a specific object, we look at the problem of finding a set of torques $\tau \in \mathbb{R}^L$ that when applied, balance a given external wrench $w \in \mathbb{R}^6$. In this work, we are using the torque-controlled DLR-Hand II [18], shown in Fig. 2. Here, each finger has four joints, however middle and distal joints are coupled via a tendon, which leads to $L = 12$ actuated joints in total.

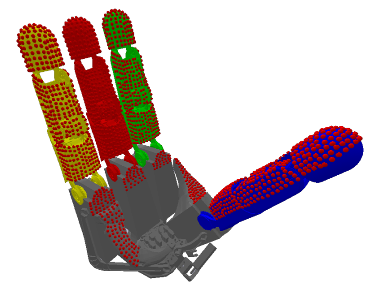


Fig. 2. The DLR-Hand II [18] which we use in our experiments. Each red dot represents a potential contact point.

A. Problem formulation

A grasp is defined as a 6D hand pose h and a joint configuration $q \in \mathbb{R}^L$. It results in N hand-object contacts, each defined by its position p_i and object normal n_i. For a grasp to stay stable, the contact forces $\{f_1, ..., f_N\}$ need to resist the external wrench w being applied to the object. This is represented via the constraint

$$w = -Gf\,, \tag{1}$$

with $G \in \mathbb{R}^{6\times 3N}$ being the grasp matrix, mapping a vector of all stacked contact forces f to the resulting wrench applied to the object. Furthermore, the torques τ applied in the joints need to balance the contact forces via

$$\tau = J^T f\,, \tag{2}$$

with each torque being constraint to stay inside its limits $\tau \in [\tau_{\text{min}}, \tau_{\text{max}}]$. To model the dependence between contact force and to solve the indeterminacy of the force distribution, we make use of the fact that hand and object are not completely rigid, but slightly elastic (this is esp. true for the silicone fingertips of the DLR-Hand II). So similar to Bicchi [5], we place springs at each contact. The springs are loaded by applying small displacements to the object $\Delta x \in \mathbb{R}^6$ and to the joints $\Delta q \in \mathbb{R}^L$. Specifically, the displacement vector d, being a stack of the displacements $d_i \in \mathbb{R}^3$ of the springs at each contact point i, is defined as

$$d = G^T \Delta x - J \Delta q\,. \tag{3}$$

The displacements d_i can each be decomposed into a displacement $d_{\text{n},i}$ along the corresponding contact normal and a displacement $d_{\text{t},i}$ along the contact tangentials. Using the stiffness k of the springs[2], the displacements can be related to the contact forces. The normal forces f_n are defined as

[2]For simplicity, we assume the same stiffness for all contacts, however it is also possible to use different values for different contacts or normal and tangential force.

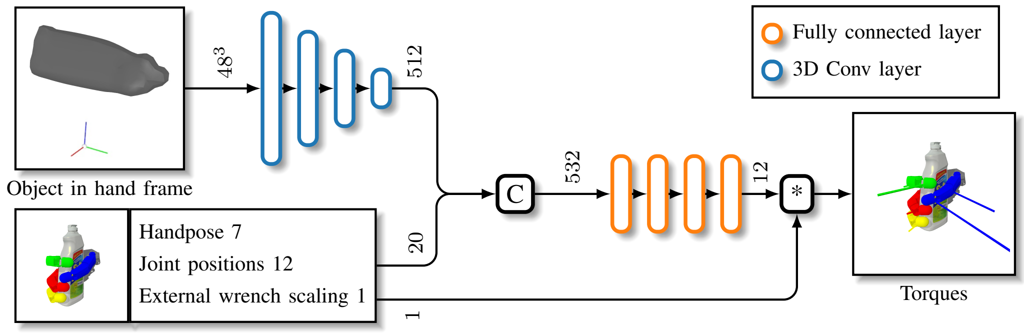


Fig. 3. Visualization of our torque prediction network architecture. The given SDF voxelgrid is processed by four 3D convolutional layers and the resulting feature vector is concatenated to the remaining inputs. Based on the result, four fully connected layers predict the torques to apply.

$$f_{\mathrm{n},i} = \begin{cases} k d_{\mathrm{n},i} & \text{if } n_i \cdot d_{\mathrm{n},i} < 0 \\ 0 & \text{if } n_i \cdot d_{\mathrm{n},i} \geq 0 \end{cases} . \tag{4}$$

The cut-off here is necessary, as the contacts are not sticky, meaning one can only push the object along the contact normal, but not pull it. The tangential forces f_{t} are defined as

$$f_{\mathrm{t},i} = \begin{cases} k d_{\mathrm{t},i} & \text{if } \|d_{\mathrm{t},i}\| < \mu \|d_{\mathrm{n},i}\| \\ k\mu \|d_{\mathrm{n},i}\| \frac{d_{\mathrm{t},i}}{\|d_{\mathrm{t},i}\|} & \text{else} \end{cases} . \tag{5}$$

Here the tangential component is limited by the maximum friction force based on the normal force $f_{\mathrm{n},i}$ and the friction coefficient μ. This constraint is defined according to the Coloumb friction model for hard contacts.

If a solution to the formulated constraints above can be found, usually an infinite number of solutions exist. To make the solution unique, we add the objective

$$\tau^* = \arg\min_{\tau} \sum_{i=1}^{L} \|\tau_i\|^2 . \tag{6}$$

In this way, the solution with minimal torques is used. However, our formulation is also compatible with other objectives like minimizing the contact forces.

B. Optimizer

In summary, an optimization problem can be constructed to find Δx and Δq that minimize the objective (6) while considering the constraints (1) to (5). We solve the full non-linear problem without potentially physically implausible linearizations. Due to the cut-off in the two non-linear constraints, the gradient can become zero and make gradient-based optimizers get stuck. Therefore, we chose gradient-free evolutionary optimization. However, it turns out that applying the optimizer to the full problem leads to suboptimal results. Instead, we propose to split up the problem into an outer evolutionary optimizer only considering Δq and an inner root finder, which finds, if possible, in each evolutionary iteration the Δx which fulfills (1). We run the optimizer with a population size of 120 for 2000 iterations which takes ca. 300 s per grasp on a single CPU core.

IV. Learning and Applying Torque Prediction

The optimizer described in the previous section is able to find the minimal torques that are required to counter a given external wrench. However, this system is not directly applicable to the real robot due to multiple reasons: First, precise contact positions and normals are required which are not available when dealing with unknown objects and incomplete observations. Also, the optimization is computationally expensive and is not able to deliver the solution in real-time. Instead, we propose to learn the mapping

$$\mathcal{F} : \{o, h, q, F_{\mathrm{g}}\} \to \{\tau\} \tag{7}$$

using a neural network $\mathcal{F}$. Next to the observation of the object o, the hand pose $h \in \mathbb{R}^7$ (position and rotation as quaternion) and joint configuration $q \in \mathbb{R}^L$, the network also receives the magnitude of the gravitational force F_{g} as an input. In this work, we focus on learning torques to withstand gravitational forces only. This is relevant for pick and place tasks, one of the most common robotic applications. However, our approach can be extended to support any kind of external wrench.

A. Neural network architecture

The architecture of the neural network is shown in Fig. 3. The object observation is represented as a voxelgrid of size 48^3 containing signed distance values. To make it easier for the network to generalize across different grasps, the object is transformed into the coordinate frame of the hand. For encoding the given voxel grid, four 3D convolutional layers, each followed by a pooling layer, are applied. The resulting feature map is flattened to a feature vector of size 512 and concatenated to the given hand pose, joint configuration and magnitude of the gravitational force. Each of the additional inputs is normalized such that their distributions in the training dataset have a mean of zero and standard deviation of one. The complete feature vector of size 512 is now mapped by four fully connected layers with 1024 neurons each to the output of size 12, representing the torques applied to each joint. We multiply the predicted torques with the given magnitude of the external wrench. This has the effect that the network predicts the normalized torque vector and therefore the norm of the predictions stays similar for different F_{g}.

Nevertheless, the network cannot be indifferent to the given F_g, as, depending on the torque limits, the torques might need to be redistributed for higher F_g.

B. Training procedure

The network is trained using supervised learning. The training dataset is based on ca. 12 000 objects from the ShapeNet dataset [19]. For each object, we generate grasps using our grasping network [20]. Specifically, we let the network predict 1024 grasps per object. Then we split up the grasps into eight groups based on their approach directions. From each group, we take the three grasps that received the highest predicted grasping score. In this way, the dataset contains diverse but still relevant grasps. In total, this results in a dataset of around 280 000 grasps. For each grasp j, the ground truth torques τ_j are generated in the following way: First, for each grasp, we determine the maximum gravitational force magnitude $F_{\text{g},j}^{\text{max}}$ it can withstand using the torque limits of the hand. Based on that, we sample one counterable gravitational force magnitude via $F_{\text{g},j} \sim \mathcal{U}(0.1\,\text{N}, \min(20\,\text{N}, F_{\text{g},j}^{\text{max}}))$. Finally, the optimization procedure described in Section III is again used to find the minimal torques τ_j which can withstand $F_{\text{g},j}$. In this way, we form a training sample $(o_j, h_j, q_j, F_{\text{g},j}, \tau_j)$. As contact points, we use each potential contact point (see Fig. 2) that is closer than $1\,\text{mm}$ to the object's surface. The network is trained for 100 000 iterations using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 64. The loss is calculated using an L2 loss function.

C. Combination with an impedance controller

In reality, there always exist small errors in the determination of the external wrench, the observation of the object's geometry and the prediction of the network. When only applying the raw torques, such small errors could in the worst case break the grasp. So to still be able to stably hold the object, we apply the torques via a joint-level impedance controller. The controller can be described as $\tau = k_\text{p}(q_\text{d} - q_\text{m}) - k_\text{d}\dot{q}$, with q_d being the desired and q_m being the measured joint configuration. As we assume a static situation, where the fingers are already placed on the surface of the object, we can ignore the velocity term $k_\text{p}\dot{q}$. Thus, to set q_d based on the torques τ predicted by the neural network, the controller can be configured via

$$q_\text{d} = \frac{\tau}{k_\text{p}} + q_\text{m}\,. \tag{8}$$

In this way, the impedance controller will apply the commanded torques τ in the perfect case, but can still slightly adapt them based on small errors or external influences that occur on the real system.

V. Experiments

A. Experimental setup

We extensively evaluate the trained network in a physics-based simulator on a set of 536 grasps across 65 household objects (see Fig. 4). To simulate a given grasp together

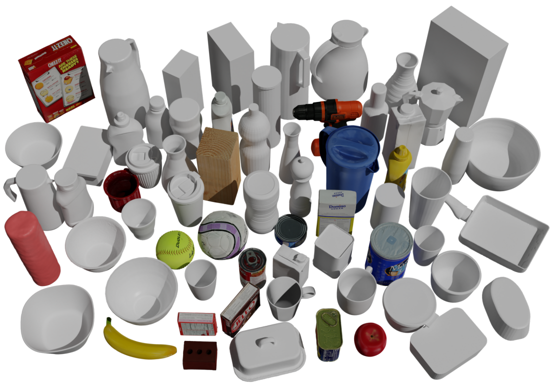


Fig. 4. The 65 objects used in the evaluation. The 20 colored objects are from the YCB dataset [21].

with a set of torques we follow the following procedure: The simulation starts with the object being placed on a table and starts moving along the approach direction towards the object and during the last $2\,\text{cm}$, the fingers are closed simultaneously according to the grasp until they get in contact. Based on the resulting joint configuration, the trained network predicts a set of torques τ suitable for the given gravitational force F_g. We assume F_g given except for in the experiments in Section V-D. Now we set the desired offsets q_d of the impedance controller as described in (8). Afterward, the table is removed, such that the full gravitational force acts on the object (see Fig. 6 for two qualitative examples). The grasp is marked as breaking if the objects moves more than $6\,\text{mm}$ down against gravity.

We compare our approach to two heuristic baselines: The first heuristic, called *baseline constant*, applies the same torques for each object and at each finger. The torques τ_bc are set high enough such that even heavy objects can be held. Specifically, we set the torque of each second joint to $1.6\,\text{N}\,\text{m}$ and the torque of the coupled third/fourth joint to $0.8\,\text{N}\,\text{m}$. The second heuristic, called *baseline scaled*, extends this strategy by taking the magnitude of the gravitational force into account: $\tau_\text{bs} = (F_\text{g}/10\,\text{N})\tau_\text{bc}$. Both baselines are also combined with an impedance controller.

B. Accuracy in countering the external wrench

First, we evaluate how well the torques predicted by the neural network can withstand the given gravitational force F_g. Therefore, the grasp together with the predicted torques is simulated multiple times each time using a different gravitational force. In detail, we test each gravitational force in the range $[0\,\text{N}, F_\text{g} + 10\,\text{N}]$ using a step size of 0.5 and in the end, we determine the maximum force F'_g at which the grasp did not break. This maximum non-breaking external force should be as close as possible to the given gravitational force. If $F'_\text{g} < F_\text{g}$, the given torques are too small or are not distributed correctly, such that the grasp under influence of F_g would be not stable. If $F'_\text{g} > F_\text{g}$, the given torques are larger than necessary and are not the minimal torques.

The resulting deviations $F'_\text{g} - F_\text{g}$ for all tested objects are shown in the density diagram of Fig. 5. We first evaluate

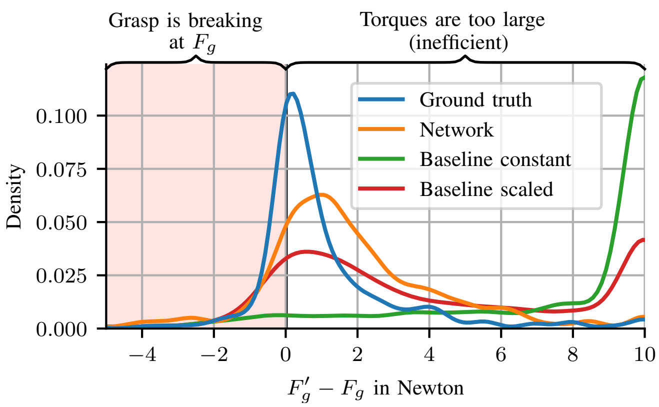


Fig. 5. A density diagram showing the deviation of the actual breaking force magnitude F_g' from the desired force magnitude F_g in newton. All grasps that did not break in the tested interval are collected at the maximum tested deviation of $10\,\mathrm{N}$. So, the two baselines, which have an additional peak there, cause deviations that are even larger than shown in the figure.

the ground truth torques found by the optimizer described in Section III. The peak here is clearly around zero, which is the desired outcome. The network's predictions have a similar peak, however, they tend to withstand more external force than necessary. This might be caused by ambiguities in the training labels, which make the network learn to predict higher torques than necessary. The constant baseline is nearly always using much higher torques than necessary. However, based on the density around zero, there are still a few situations where such high torques are necessary. The baseline that scales the torques proportional to the F_g is more efficient, however in some situations still too high. Decreasing the scaling factor would lead to shifting the peak around zero into the negative region which is not desired. This shows that the size of the torques is not only dependents on the magnitude of the external wrench but also on the grasp itself. Our proposed neural network is capable of taking both into account and therefore superior to the tested baselines.

Even when using the ground truth torques found by the optimizer, there exists a difference between desired and actual breaking force. This can be explained by multiple factors: The simulator uses different contact points than the ones that are used in the torque optimization. Furthermore, it can happen that a grasp breaks but immediately slips into a new stable configuration. In this way, it might seem that a grasp withstands more gravitational force than its static evaluation actually allows.

C. Object movement

To compare, how well the torques predicted by our neural network can preserve the pose of the object, we measure the difference in object pose between before and after applying the predicted torques. Here we only look at the grasps which are non-breaking under the desired gravitational force F_g.

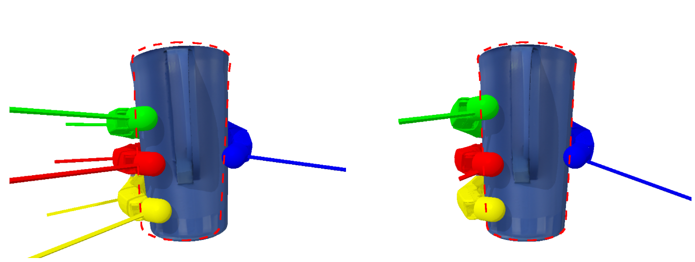
Fig. 7a shows a density plot over the translational deviation and Fig. 7b shows the rotational deviation of the object from the pose it had before applying the provided torques. In both cases, the network performs very similarly to the ground truth torques found by the optimizer. It is able to keep for $90\,\%$ of the grasps the translation below $2.9\,\mathrm{mm}$ and the rotation below 3.1°. Both baselines completely fail compared to that. Fig. 6 illustrate this behavior in more detail: In Fig. 6a the baseline, applying the same torques at each finger, involuntarily rotates the object, as the force applied by the ring finger is not countered by any opposing contact. The neural network is aware of this and therefore chooses to mainly apply torques in the ring finger and middlefinger. Fig. 6b depicts a usual precision grasp, where the thumb is acting contrary to the combined force of the other three fingers. When using the heuristic baseline, this leads to shifting the object towards the thumb, until the thumb is deflected to a point where the forces equal out again. The torques predicted by the network are correctly scaled such that this equilibrium already exists in the beginning and the object stays still.

D. Evaluation on the real robot

We evaluate our approach on our precisely calibrated [22] humanoid robot *Agile Justin* [23] by incorporating it into our grasping pipeline: For a given object, we first observe an incomplete 3D model using a Kinect depth camera [24]. Via shape completion, the observation is completed and used to predict a stable grasp via our grasping network [20]. Using a learning-based motion planner [25], the hand now approaches the object as specified by the predicted grasp. Afterward, the fingers are closed until they contact the object. Based on the completed object and the resulting grasp, our proposed network now predicts suitable torques. As we do not know the mass of the unknown object, we start with a small estimate of the gravitational force and then slowly increase it. At each step, we increase the lifting force applied by the torque-controlled arm accordingly until at some point, our estimate of the gravitational force exceeds the actual gravitational force and the object is lifted up. We now stop increasing our estimate any further. We found that the predicted torques are able to stably hold the object and reduce involuntary object movement compared to the heuristic approaches. The forward pass of the network takes about $1.5\,\mathrm{ms}$ making it suitable for real-time applications.

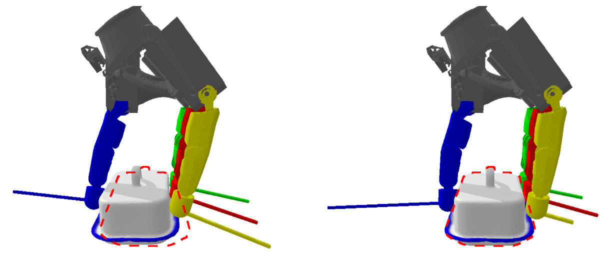
VI. Conclusions

In this work, we presented a novel learning-based approach for predicting torques that balance for a given grasp the external wrench. We showed that a neural network is capable of learning the mapping from grasping pose, joint configuration, external wrench magnitude and object observation to the corresponding set of torques. The predicted torques are efficient and hold the object without involuntary object shifting or rotation. In both regards our methods clearly outperform the two tested baseline heuristics. The network generalizes to unknown objects without requiring detailed contact information. The prediction only requires one forward pass taking $1.5\,\mathrm{ms}$ which makes its application suitable for real-time applications. To generate training data, we formulate the analytical computation of torques as an



Baseline scaled　　Neural network

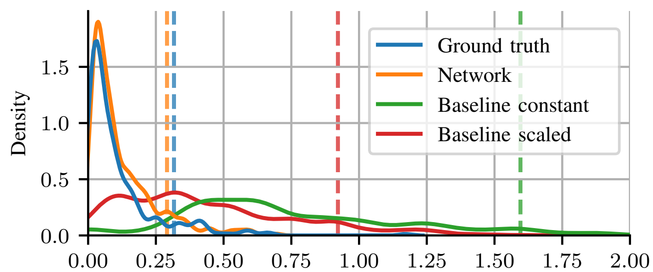
(a) YCB pitcher: The baseline involuntarily rotates the object.



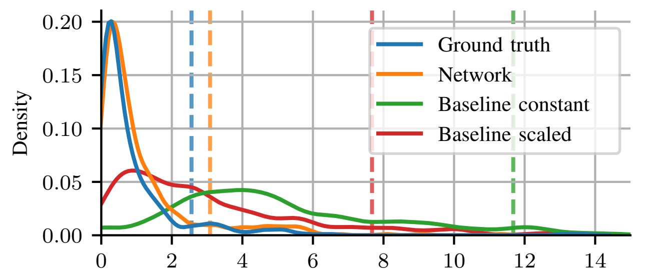
Baseline scaled　　Neural network

(b) Butter box: The baseline shifts the object to the left.

Fig. 6. Multiple examples comparing how the objects deviate from their initial position when using the heuristic baseline compared to our torque prediction network. The red dashed line represents the object's outline in its initial pose.



(a) Translational deviation in centimeter.



(b) Rotational deviation in degree.

Fig. 7. Density diagrams showing the object deviation of its initial pose after applying the torques given by the respective method. Vertical dashed lines show the 90 % quantile of each method.

optimization problem and solve the indeterminacy of multi-contacts by using an elastic model. In the future, we plan to extend the approach to all kinds of external wrenches and incorporate the trained network into a closed-loop controller that is able to actively react to changing disturbances.

References

[1] F. T. Cheng and D. E. Orin, "Efficient Algorithm for Optimal Force Distribution—The Compact-Dual LP Method," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 178–187, 1990.

[2] M. Buss, H. Hashimoto, and J. B. Moore, "Dextrous hand grasping force optimization," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 406–418, 1996.

[3] L. Han, J. C. Trinkle, and Z. X. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 663–674, 2000.

[4] Y. Zheng and W. H. Qian, "A fast procedure for optimizing dynamic force distribution in multifingered grasping," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1417–1422, 2006.

[5] A. Bicchi, "Force distribution in multiple whole-limb manipulation," in *Proceedings - IEEE International Conference on Robotics and Automation*, 1993.

[6] M. Gabiccini and A. Bicchi, "On the role of hand synergies in the optimal choice of grasping forces," in *Robotics: Science and Systems*, 2011.

[7] D. Prattichizzo *et al.*, "On the manipulability ellipsoids of underactuated robotic hands with compliance," *Robotics and Autonomous Systems*, 2011.

[8] M. Haas-Heger and M. Ciocarlie, "Accurate Energetic Constraints for Passive Grasp Stability Analysis," *IEEE Transactions on Robotics*, vol. 36, 2020.

[9] Q. Lu, M. van der Merwe, and T. Hermans, "Multi-Fingered Active Grasp Learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[10] B. Wu *et al.*, "Generative Attention Learning: a "GenerAL" framework for high-performance multi-fingered grasping in clutter," *Autonomous Robots*, vol. 44, no. 6, pp. 971–990, 2020.

[11] U. R. Aktas *et al.*, "Deep dexterous grasping of novel objects from a single view," *IEEE Transactions on Robotics (T-RO)*, 2019.

[12] P. Mandikal and K. Grauman, "DexVIP : Learning Dexterous Grasping with Human Hand Pose Priors from Video," in *CoRL*, 2021.

[13] Z. Yilun and H. Kris, "6DOF Grasp Planning by Optimizing a Deep Learning Scoring Function," 2017.

[14] J. Varley *et al.*, "Shape completion enabled robotic grasping," in *IEEE International Conference on Intelligent Robots and Systems*, 2017.

[15] P. Schmidt *et al.*, "Grasping of Unknown Objects Using Deep Convolutional Neural Networks Based on Depth Images," in *IEEE International Conference on Robotics and Automation*, 2018.

[16] J. Lundell, F. Verdoja, and V. Kyrki, "DDGC: Generative Deep Dexterous Grasping in Clutter," in *IEEE Robotics and Automation Letters*, 2021.

[17] L. Shao *et al.*, "UniGrasp: Learning a Unified Model to Grasp with Multifingered Robotic Hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.

[18] J. Butterfaß *et al.*, "DLR-Hand II: Next generation of a dextrous robot hand," in *Proc. IEEE International Conference on Robotics and Automation*, 2001.

[19] A. X. Chang *et al.*, "ShapeNet: An Information-Rich 3D Model Repository," no. arXiv:1512.03012 [cs.GR], 2015.

[20] D. Winkelbauer *et al.*, "A two-stage learning architecture that generates high-quality grasps for a multi-fingered hand," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[21] B. Calli *et al.*, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[22] J. Tenhumberg *et al.*, "Self-contained calibration of an elastic humanoid upper body using only a head-mounted rgb camera," in *IEEE-RAS 21st International Conference on Humanoid Robots*, 2022.

[23] B. Bäuml *et al.*, "Agile justin: An upgraded member of dlr's family of lightweight and torque controlled humanoids," in *IEEE International Conference on Robotics and Automation*, 2014.

[24] R. Wagner, U. Frese, and B. Bäuml, "Real-time dense multi-scale workspace modeling on a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[25] J. Tenhumberg, D. Burschka, and B. Bäuml, "Speeding up optimization-based motion planning through deep learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.