A new benchmark for group distribution shifts in hand grasp regression for object manipulation. Can meta-learning raise the bar?

Théo Morales
Trinity College Dublin
Dublin, Ireland
moralest@tcd.ie

Gerard Lacey
Maynooth University
Maynooth, Ireland
gerard.lacey@.mu.ie

Abstract

Understanding hand-object pose with computer vision opens the door to new applications in mixed reality, assisted living or human-robot interaction. Most methods are trained and evaluated on balanced datasets. This is of limited use in real-world applications; how do these methods perform in the wild on unknown objects? We propose a novel benchmark for object group distribution shifts in hand and object pose regression. We then test the hypothesis that meta-learning a baseline pose regression neural network can adapt to these shifts and generalize better to unknown objects. Our results show measurable improvements over the baseline, depending on the amount of prior knowledge. For the task of joint hand-object pose regression, we observe optimization interference for the meta-learner. To address this issue and improve the method further, we provide a comprehensive analysis which should serve as a basis for future work on this benchmark.

1 Introduction

Joint hand-object pose regression methods – and hand pose regression in object grasping – are not commonly benchmarked for in-the-wild data or a wide variety of object grasps [15]. They aim to generalize to unseen poses on the same objects by learning from a large collection of diverse poses and grasps [5, 9, 10]. This may be due to the cost of annotating 3D pose and the limited availability of 3D scanned objects, which encourages researchers to reuse objects' meshes. In addition, synthetic datasets of realistic object grasps are hard to produce and lead to domain adaptation challenges. However, such models have limited use if they are only accurate for a limited set of objects.

To accurately predict the pose of a hand occluded by an object, a deep learning model must learn prior knowledge of various grasps of diverse objects. Neural networks are usually trained to keep a balance between generalization and specialization as their knowledge is frozen when deployed. However, temporarily learning at test time would allow them to specialize their parameters for a specific object grasp while remaining generalisable by forgetting this specialized knowledge. Test-time adaptation (TTA) methods demonstrate improvements in regression accuracy for distribution shifts and in-the-wild data in other domains, such as human pose estimation [12], gaze estimation [20], mesh reconstruction [19] and video object segmentation [25]. We propose to achieve this on the grasp prediction problem with a meta-learning algorithm, where the goal is to quickly learn new tasks from a few examples at test time [1, 17, 24]. We then evaluate this method on a novel benchmark for group distribution shifts in hand-object pose regression for object grasping. How does the performance of a CNN pose predictor evolve as the test set grasps diverge from the training set? We answer this question and look at the advantages and limitations of meta-learning for this application via experiments and empirical analysis.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Contributions In this work, we: (a) reformulate grasp prediction in the context of multi-task learning such that meta-learning can be applied, (b) propose a new benchmark for object group shifts in hand grasp regression based on the DexYCB dataset [5], and (c) prove that meta-learning is effective at tackling group distribution shifts for hand grasp regression.We also find an increase in accuracy for unknown objects from 6 training objects upwards. We compare the relative error of the meta-learning with a baseline on our benchmark and provide a comprehensive analysis of the limitations of the method.

2 A benchmark for object group shifts in hand grasp regression

In this section, we explain the task set creation process which is necessary so that meta-learning can be applied to the grasp prediction problem.

Task set creation To cast pose prediction as a multi-task problem and apply meta-learning, we must create a dataset of tasks from a dataset of samples. For such a task set, we require that:

- Each task is composed of a support set of K randomly sampled images corresponding to one manipulation sequence of an object by one subject, and a query set of Q distinct random samples from the said sequence.
- A series of Ω-objects-left-out splits is used, where Ω is the number of objects absent in the training split and placed in the test split. Thus, all images associated with $\Omega + \min(5, \frac{\Omega}{2} + (\Omega \bmod 2))$ objects are removed from the training split: the samples associated with 5 or fewer objects are used for validation while the ones for other Ω objects are used for testing. This ensures that there is no overlap between the training, validation and test splits.
- For all values of Ω, the objects are randomly sampled with a fixed random seed for reproducibility across experiments.

We use the DexYCB dataset to create our task set and run our experiments, although the procedure is applicable to any object manipulation dataset (see the survey of [15] for an overview of hand-object pose datasets). Along with a similarity study of object grasps from this dataset, we provide more implementation details as well as visual examples of tasks in Appendix A, and the code on GitHub [1].

3 Evaluating the effectiveness of meta-learning

In this section, we describe our experimental framework used to assess the hypothesis that TTA of a pose prediction model through meta-learning improves the generalization of unknown objects. We give an introduction to optimization-based meta-learning in Appendix B.

3.1 Experimental framework

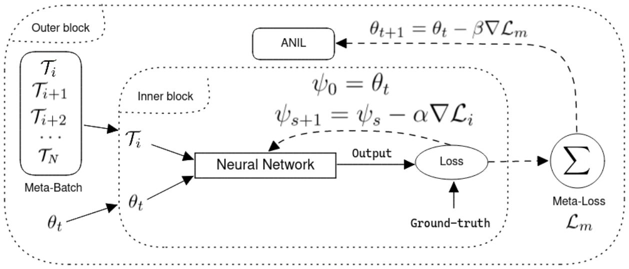


Figure 1: **An overview of the meta-learning system.** – The outer block optimizes the inner block's parameters by back-propagating the second-order gradients of the meta-loss. This cost function is the cumulative loss of the inner block on all N tasks, computed with the adapted parameters ψ_S obtained after S optimization steps and with the same initial parameters θ_t.

[1] https://github.com/DubiousCactus/meta-learning-HOPE

To evaluate the effectiveness of meta-learning at tackling object group shifts and dealing with highly imbalanced dataset splits, we apply ANIL [21] with the improvements of MAML++ [2] to a ResNet18 [13] baseline. As seen in Fig. 1, our framework consists of an outer block (ANIL) and an inner block (ResNet18).

Outer block The outer block trains the inner block to learn good parameters that are amenable to fast adaptation with few gradient descent steps and examples. It is based on MAML and is a re-implementation of ANIL [21] which brings performance improvements to the former by only adapting the head of the network and learning a common feature extractor for all tasks. We further implement some of the improvements proposed by [2], such as the Multi-Step Loss and Derivative-Order Annealing. Furthermore, to deal with the meta-overfitting phenomenon that arises from the non-mutual-exclusiveness of our regression tasks, we implement the regularization method of [26]. We use the *learn2learn* [3] library to facilitate the implementation and minimize mistakes.

Inner block For the inner block, the baseline pose regression neural network, we use the same backbone for both experiments: ResNet18 [13] pre-trained on ImageNet [23]. The choice of a shallow ResNet architecture is motivated by the speed of training and low memory footprint for meta-learning. It allows a fair comparison without being impacted by the limitations of meta-learning methods regarding more complex architectures. Combining these more complex and efficient architectures with meta-learning should be the focus of future work. For a fair comparison, the baseline is trained with weight decay regularization to minimize overfitting.

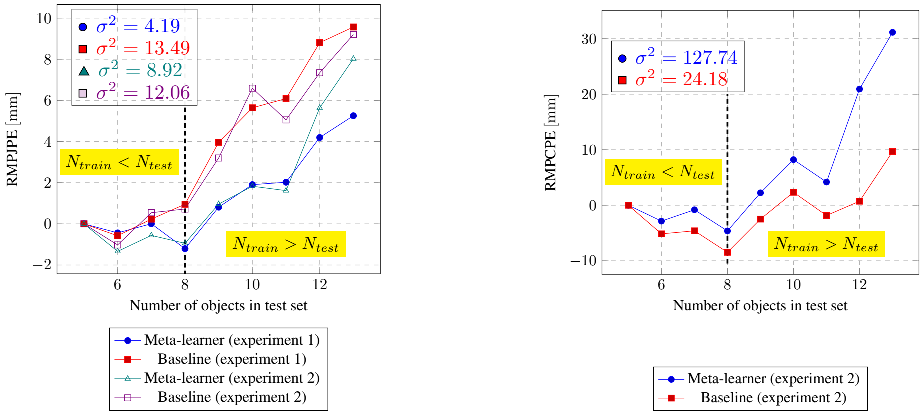
For the baseline, we use a batch size of 64 and a learning rate $\alpha = 10^{-3}$ for 100 epochs. For the meta-learner, we use a meta-batch size of 8 and found learning rates $\alpha = 10^{-5}$ and $\beta = 10^{-2}$ via hyper-parameter search using *Weights & Biases* [4], with 300 epochs. For both methods, we normalize the inputs according to ImageNet's statistics and align the ground-truth poses to the root of the hand. Using wrist-aligned pose labels greatly improves the convergence of both methods, and the absolute position in 3D space is not in the scope of this work. For the meta-learner, we use $K = 10$ for the support set, $Q = 50$ for the query set, and 15 adaptation steps. This implies that during evaluation, there are $K \times (\frac{T}{K+Q})$ samples – with T being the size of the test split – virtually removed from the test split, which is a slight disadvantage for the baseline. It is acceptable because the goal is to measure error changes relative to various test splits. In fact, the pose prediction error of the meta-learner is roughly $1.5\times$ that of the baseline for both experiments. When evaluating the meta-learner, we average all metrics over 5 runs to account for the run-time stochasticity coming from the support and query sets sampling. For all experiments, we normalize the error against each model's performance on the easiest setting (5 test objects) to compare the relative error changes.

With this framework, we design two experiments: (1) we evaluate the relative error of the 3D hand pose regression for manipulation samples of unknown objects, (2) we reiterate on the joint hand-object pose regression by incorporating the object bounding cuboid coordinates in the targets, as in [7, 14, 18].

3.2 Results

Experiment 1: hand pose only Fig. 2a shows the meta-learner's ability to deal with object group shifts on a macro scale. The size of the training set decreases inversely and in proportion to the size of the test set, thus the error is expected to go up for the baseline as the test set grows. The error of both models rises steeply from 8 objects in the test set, where the gap starts to widen, but less steeply for the meta-learner. It corresponds to 8 training objects as well (with 4 for validation); a more or less balanced dataset. By fitting a linear regression model on both curves, we prove that the difference in the two slopes is statistically significant with a p-value of 0.0031.

We look at the micro scale by freezing the training split for 3 levels of prior knowledge, and progressively adding objects to the test split; the results are shown in Fig. 3. Both methods are trained and evaluated on 3 train/test splits; the average curve is shown for the 3 sizes. The meta-learner and the baseline behave identically in Fig. 3a since the training set is too small for the meta-learner to collect sufficient prior knowledge for adaptation: 3 training objects are not sufficient prior knowledge. As the training set size and variability grow for Fig. 3b, the meta-learner reduces the error significantly better than the baseline: 6 objects are enough prior knowledge. However, with 9 training objects in Fig. 3c, both curves have a similar horizontal slope and a variance roughly a third of that of Fig. 3a.



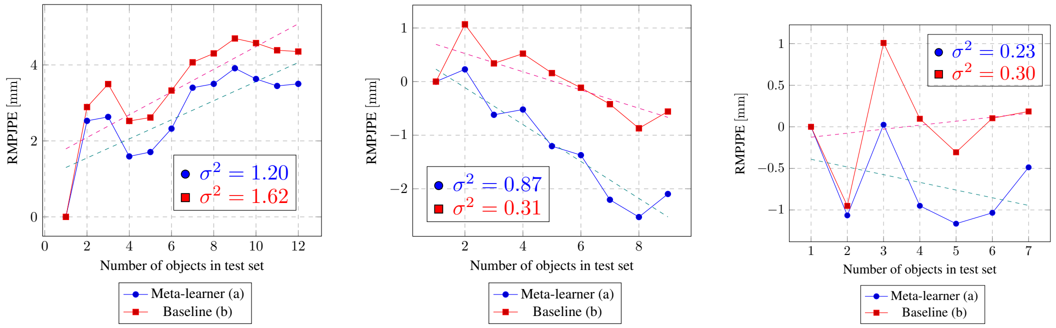
(a) **Hand only** – For both experiments, the error increases less steeply and has lower variance σ^2 as the number of test objects grows for the meta-learner. The latter is better at adapting to dataset imbalance.

(b) **Object only** – Both methods exhibit a high variance but the meta-learner is worse by an order of magnitude. The latter isn't able to reduce the object pose error.

Figure 2: **Relative Mean Per-Joint Pose Error (RMPJPE) & Mean Per-Corner Pose Error (RMPCPE) as functions of the imbalance level** – All curves are aligned to the origin to show the relative changes and compare the generalisability of both methods. The more objects are in the test set, the more deprived the training set (see Tab. 1 in Appendix A); the dashed line shows the equilibrium between the two. Both experiments are plotted in Fig. 2a since they both regress the hand pose, while only experiment 2 predicts the object pose and is plotted in Fig. 2b on its own. The meta-learner is overall better at handling object class imbalance, and thus generalisability to unknown grasps. However, this behaviour is not seen for the prediction of the object corners coordinates.

In short, the baseline and the meta-learner generalize equally well. For this case, the improvement of the meta-learner may nonetheless be revealed with more test objects if available. The findings are different than in Fig. 2a since the models are here compared in terms of accuracy for a given training set size, and not their ability to deal with dataset imbalance.

Experiment 2: joint hand-object pose Analogously to the previous experiment, Fig. 2a shows a mild improvement in the hand pose regression for the meta-learner when trained for the task of joint hand-object pose regression. This improvement however is lesser than for the first experiment, as reflected by the smaller gap between the two curves and a p-value of 0.3186. The null hypothesis cannot be rejected: the meta-learner shows no significant improvement in generalization. Fig. 2b



(a) **3 training objects** – Both methods have a similar variance and error curves, with similar slopes ($p = 0.61$). There is insufficient prior knowledge for the meta-learner to be effective.

(b) **6 training objects** – The error decreases more steeply for the meta-learner ($p = 0.02$), although its variance is higher. There is sufficient prior knowledge for the meta-learner to be effective.

(c) **9 training objects** – The meta-learner has a lower variance but both error slopes are similar ($p = 0.38$). Both models generalise well with a non-significant improvement from the meta-leaner.

Figure 3: **Relative Mean Per-Joint Pose Error (RMPJPE) as function of the test split for experiment 1 with 3 training set sizes.** The dashed lines are the linear regression models.

confirms this phenomenon and shows even worse results than the baseline. We tried increasing the network's capacity but obtained similar results with underfitting. Therefore we can hypothesize that this joint regression causes interference in the adaptation to both tasks, and this problem would have to be processed separately in future work. In Sec. 4.1, we run more experiments to assess the hypotheses made on the meta-learner in an attempt to explain the results.

4 Empirical analysis and conclusions

Takeaways From these experiments, we can conclude three things: (1) for hand grasp prediction, the meta-learner is better able to deal with dataset imbalance than the baseline, as shown in Fig. 2a this is especially true when using less than 8 training objects (or more than 8 test objects), (2) the meta-learner improves the accuracy with enough training data (i.e. 6 objects) on unknown objects, and (3) this method is ineffective for joint hand-object pose regression, and is even worse than the baseline for object pose prediction (see Fig. 2b). This weakness is investigated in the following section.

4.1 Empirical analysis

Hypothesis 1: the model learns specialized object-specific parameters ANIL and the like learn prior knowledge of related tasks during training so that they can adapt to specialized parameters for a novel task with few optimization steps [16]. But does it translate to learning object- or grasp-specific parameters in the proposed framework? To verify this, we plotted the t-SNE embeddings of the network's head parameters post adaptation (see Appendix C.1). We would expect to find clusters of tasks for the same manipulated object but none appeared. It reveals that the parameters do not specialize to a specific grasp or object after adaptation, thus refuting this hypothesis. This may well be due to the *non-mutual exclusiveness* of tasks incurring memorization overfitting [22], for which the regulariser of [26] did not help. In future work, the problem should be properly formulated in the meta-objective to encourage specialization.

Hypothesis 2: Using Oriented Bounding Box (OBB) coordinates in the training signal constrains the hand-object pose In experiment 2, we rely on this hypothesis and make the conjecture that OBBs differ enough to lead away from the initialization in the loss landscape. In truth, Fig. 2b shows that it limits the adaptation for the hand pose and leads to worse generalization than the baseline for the object pose. [8] define the shape as *"all the geometrical information that remains when location, scale and rotational effects are filtered out from an object"*. Therefore we can consider that most OBBs have almost identical shapes, except for unusually thin or wide objects such as *large marker* or *pitcher base*. We postulate that due to this, the adaptation phase should produce small gradients for the object keypoints and thus the hand keypoint regression should not be severely impacted. However, we observed a $2 - 3\times$ increase in gradient norms from experiment 1 to 2 (see Appendix C.2 for more details). We hypothesize that this is due to the complex hand-object relationship which is not expressed in the meta-objective, causing interference in the meta-optimization landscape. In future work, these constraints should be explicitly defined in the objective, or the two problems decoupled.

Conclusions. Our results show measurable improvements over the baseline, where it reduces the error rise with less than 8 training objects. On the other hand, it can generalize better than the baseline from 6 training objects. For the task of joint hand-object pose regression, the meta-learner's ability to deal with object group shifts is dampened and is worse than the baseline for the object pose. In our empirical analysis, we show that this may be due to interference in the optimization. We further propose solutions to address this issue and to encourage the model to learn object-specific parameters during adaptation. This should in turn improve the effectiveness of this method.

Acknowledgments and Disclosure of Funding

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3988–3996, 2016. ISBN 9781510838819. 1, 10

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *International Conference on Learning Representations*, 2018. 3, 10

[3] Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. *ArXiv*, August 2020. URL `http://arxiv.org/abs/2008.12284`. 3

[4] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com. 3

[5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 8

[6] Dawson-Haggerty et al. trimesh, 2019. URL `https://trimsh.org/`. 8

[7] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. Hope-net: A graph-based model for hand-object pose estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6607–6616, 2020. 3

[8] Ian L. Dryden and Kanti V. Mardia. Statistical shape analysis. In *Statistical Shape Analysis*, 1998. 5

[9] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[10] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 1, 8

[11] Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. Ho-3d_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *ArXiv*, abs/2107.00887, 2021. 8

[12] Miao Hao, Yizhuo Li, Zonglin Di, Nitesh B. Gundavarapu, and Xiaolong Wang. Test-time personalization with a transformer for human pose estimation. *ArXiv*, abs/2107.02133, 2021. 1

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[14] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. HOT-Net: Non-Autoregressive Transformer for 3D Hand-Object Pose Estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3136–3145, Seattle WA USA, October 2020. ACM. ISBN 978-1-4503-7988-5. doi: 10.1145/3394171.3413775. URL `https://dl.acm.org/doi/10.1145/3394171.3413775`. 3

[15] Lin Huang, Boshen Zhang, Zhilin Guo, Yang Xiao, Zhiguo Cao, and Junsong Yuan. Survey on depth and rgb image-based 3d hand shape and pose estimation. *Virtual Reality & Intelligent Hardware*, 3(3):207–234, 2021. ISSN 2096-5796. doi: https://doi.org/10.1016/j.vrih.2021.05.002. URL `https://www.sciencedirect.com/science/article/pii/S2096579621000280`. 1, 2, 8

[16] Mike Huisman, Jan N. van Rijn, and Aske Plaat. A survey of deep meta-learning. *Artif. Intell. Rev.*, 54:4483–4541, 2021. 5

[17] Brenden M. Lake, Tomer David Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2016. 1

[18] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. *ArXiv*, abs/2109.05488, 2021. 3

[19] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, X. Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *ArXiv*, abs/2012.03196, 2020. 1

[20] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9367–9376, 2019. 1

[21] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2019. 3

[22] Janarthanan Rajendran, Alex Irpan, and Eric Jang. Meta-Learning Requires Meta-Augmentation. *arXiv:2007.05549 [cs, stat]*, November 2020. URL `http://arxiv.org/abs/2007.05549`. arXiv: 2007.05549. 5

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 3

[24] Sebastian Thrun and Lorien Y. Pratt. Learning to learn. *arXiv: Learning*, pages 354–354, 1998. 1

[25] Huaxin Xiao, Bingyi Kang, Yu Liu, Maojun Zhang, and Jiashi Feng. Online meta adaptation for fast video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:1205–1217, 2020. 1

[26] Mingzhang Yin, G. Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. *ArXiv*, abs/1912.03820, 2020. 3, 5

Appendix

Table of Contents

A Task set creation

Table 1: **Image samples per split as the number of unseen objects grows in the test split** – As the test set size increases for each added object, the training set size decreases proportionally. We cap the validation objects so as to maximise the number of dataset versions.

# objects in test split	5	6	7	8	9	10	11	12	13
Training	$242,704$	$221,688$	$183,104$	$163,608$	$117,824$	$98,704$	$80,072$	$58,016$	$42,288$
Validation	$60,248$	$62,552$	$79,232$	$82,240$	$105,040$	$101,744$	$99,952$	$100,896$	$99,072$
Test	$100,960$	$119,672$	$141,576$	$158,064$	$181,048$	$203,464$	$223,888$	$245,000$	$262,552$

In order to assess the generalisation of a grasp prediction model to unseen objects, we need a dataset of 2D images of the largest amount of manipulated objects with full 3D hand joint annotations, as well as 6D object pose. The review of [15] provides an extensive list of available datasets, but their limited amount of objects was the main factor in choosing the unlisted and recently released DexYCB [5]. We began the experiment with the HO-3Dv3 dataset [10, 11] which contains 10 different objects. This limitation hindered the experimental results, as more objects are needed to have enough training, validation and testing object categories, since they cannot overlap in each split. We further used the DexYCB after realising that the results were not conclusive. It contains $582,000$ annotated frames with 10 subjects and 21 objects, of which the *large clamp* is not annotated and therefore removed. We discard all frames where the hand is not in contact with the object, such that at least two fingertips are within the object bounding box. That oriented cuboid is computed using Trimesh [6] on the associated object mesh, before applying transforms. Tab. 1 shows the final size of each split for every variant of the dataset, and Fig. 4 shows examples of tasks built from DexYCB.

With the DexYCB dataset, there are approximately 20K samples per object with 100 manipulation sequences each, providing the across-task diversity required for the training of neural networks and minimise overfitting. Meta-learning shows its power over standard gradient descent optimisation in the presence of across-task diversity, such that each task is an objective requiring specialisation. We analyse the diversity of object-specific grasps using the Generalised Procrustes Analysis and the Procrustes distance to compare mean hand shape similarity. It is defined as the sum of the squared vertex distances:

$$d = \sum_{i}^{N} [(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2 + (z_{i1} - z_{i2})^2] \tag{1}$$

for two 3D shapes with N matching vertices. The heat map of hand poses similarity in Tab. 2 shows relatively few light cells, hence most grasps are similar to each other. This is partly because subjects may grasp the same object in various and unusual ways, thus resulting in an uninformative mean hand shape. Due to this, we construct tasks from individual manipulation sequences such that there is only one grasp per task, as described in Sec. 2.

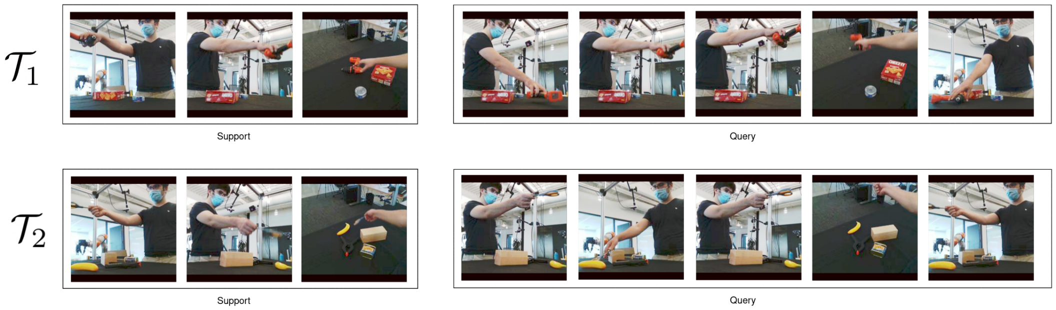
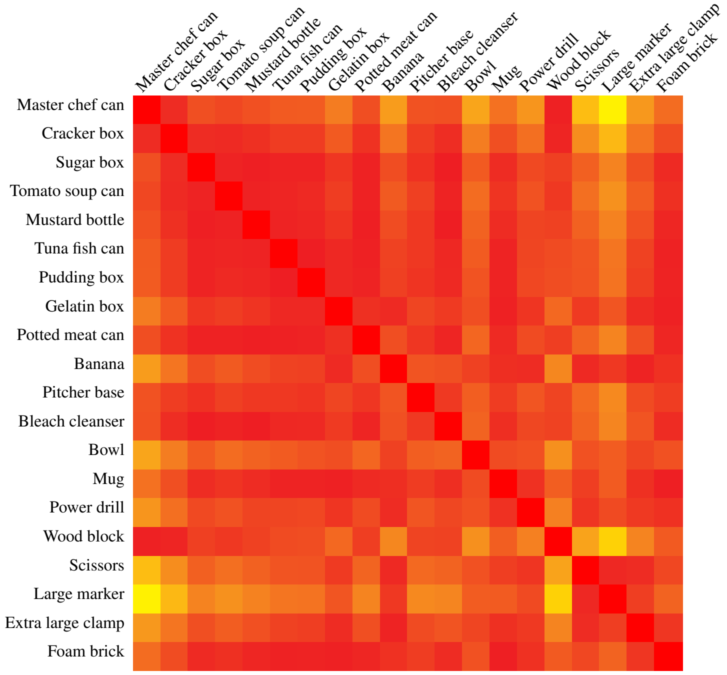


Figure 4: **Example of tasks built from the DexYCB dataset.** – $\mathcal{T}_1$ and $\mathcal{T}_2$ are two tasks composed of a support set of 3 training images, and a query set of 5 test images for 3-shot learning. $\mathcal{T}_1$ is the manipulation of a drill, while $\mathcal{T}_2$ is the grasping of scissors. The tasks contain several viewpoints but only one subject and intent of use in their support and query sets.

Table 2: **Distance heat map of mean hand shapes** – Heat map of Procrustes distances of mean hand shapes obtained via Generalised Procrustes Analysis; similarity ranges from yellow to red (best seen in colour). The *Large marker* induces the most different grasps in the dataset, while all box-shaped objects induce similar grasps (as seen by the more uniform upper left square). Note that all objects have several grasps associated with them, which vary depending on the intent of use of the subject.



B Optimisation-based meta-learning

A task is defined as $\mathcal{T}_i = \{p_i(\mathbf{x}), p_i(\mathbf{y}|\mathbf{x}), \mathcal{L}_i\}$ and comprises a distribution of samples $p_i(\mathbf{x})$, a distribution of ground-truth labels $p_i(\mathbf{y}|\mathbf{x})$ and a loss function $\mathcal{L}_i$. By targeting similar tasks, a model can make use of the shared structure in the data across tasks for a meta-learning approach. This allows the model to learn a generic – or across-task – prior, while having the plasticity required to adjust its parameters for task-specific knowledge.

Meta-learning is often presented in the context of few-shot learning (also known as K-shot learning), where the goal is to optimise a new objective from a few examples, referred to as the context set. During training, the meta-learner optimises the learner on various objectives from their context set (of $K \leq 10$ samples typically) and evaluates its performance on a validation set of Q unseen samples called the query set. The model's performance on the tasks aggregate of the latter is a measure of its ability to learn quickly.

In a single-task supervised learning context, we consider a single dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_K\}$ of pairs of images and labels. The optimisation problem is thus to minimise the loss over the entire dataset to find the optimal model parameters θ as:

$$\min_{\theta} \mathcal{L}(\theta, \mathcal{D}), \tag{2}$$

where $\mathcal{L}$ is the *Mean Squared Error* for regression tasks. It is minimised such that the update rule at each optimisation step is given by

$$\theta_{t+1} = \theta_t - \alpha_t \nabla f_{\theta_t} \tag{3}$$

with a static or adaptive learning rate α for each step t and the optimised function f parameterised by θ_t at a step t. The most characteristic aspect of the meta-learning approach is that the update rule is instead learnt end-to-end, such that the parameter update can be reformulated, as defined by [1], to

$$\theta_{t+1} = \theta_t + g_t(\nabla f_{\theta_t}, \phi) \tag{4}$$

where g is the learnt optimiser parameterised by ϕ and f is the optimisee.

Typically, an optimisation-based meta-learning system is composed of an outer block called the optimiser, which learns the update rule of the base learner, and of an inner block called the optimisee, which is the base learner that directly learns the task. The optimisee is thus optimised by the optimiser block to rapidly learn novel tasks. A general definition of the optimisation problem for meta-learning is given as

$$\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_i(\theta, \mathcal{D}_i) \tag{5}$$

for N tasks where each task is a small dataset $\mathcal{D}_i$. For MAML and other optimisation-based meta-learning algorithms, this meta-objective becomes

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\psi}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})}) \tag{6}$$

for N tasks sampled from $p(\mathcal{T})$, with post-adaptation parameters ψ. As for the update rule, it is defined as such:

$$\theta_{t+1} = \theta_t - \beta \nabla_{\theta_t} \sum_{\mathcal{T}_i \sim p(\mathcal{T})}^{N} f_{\theta_t - \alpha \nabla_{\theta_t} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_t})}. \tag{7}$$

In most cases, the learnt optimiser is only effective during training. However, some algorithms propose to learn per-layer and/or per-step learning rates for SGD as the learnable optimiser, and use them at test-time during the adaptation phase [2].

C Empirical analysis

In this section, we run extra experiments on the trained models to support the analysis of the hypotheses made in Sec. 4.1.

C.1 Visualisation of the specialised networks' parameters

We plotted the t-SNE embeddings of all parameters of the network's head, after adaptation on the context set, in Fig. 5. The absence of clusters reveals that no object-specific information is encoded in the parameters during adaptation. In Fig. 6, the weights alone of each layer of the network's head are embedded, and the biases in Fig. 7. Only the biases of the final layer seem to reveal some structure after adaptation, although it is unknown what this corresponds to.

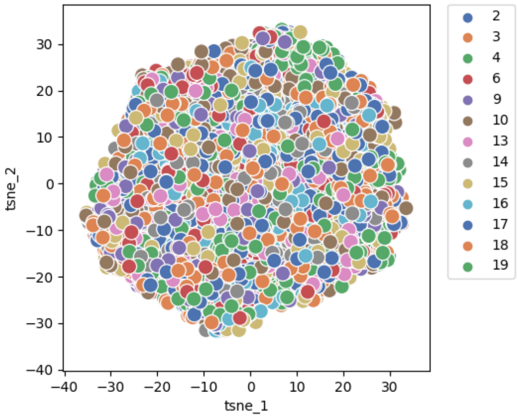


Figure 5: **Visualisation of the specialised parameters for experiment 2 –** The fully-connected layers parameters, after adaptation on the context set, are embedded with t-SNE and labelled by object. This is done for each sampled task of the largest test set (13 objects, 3814 tasks). The absence of clusters indicates that the specialised parameters do not encode object-specific information on the hand-object pose.

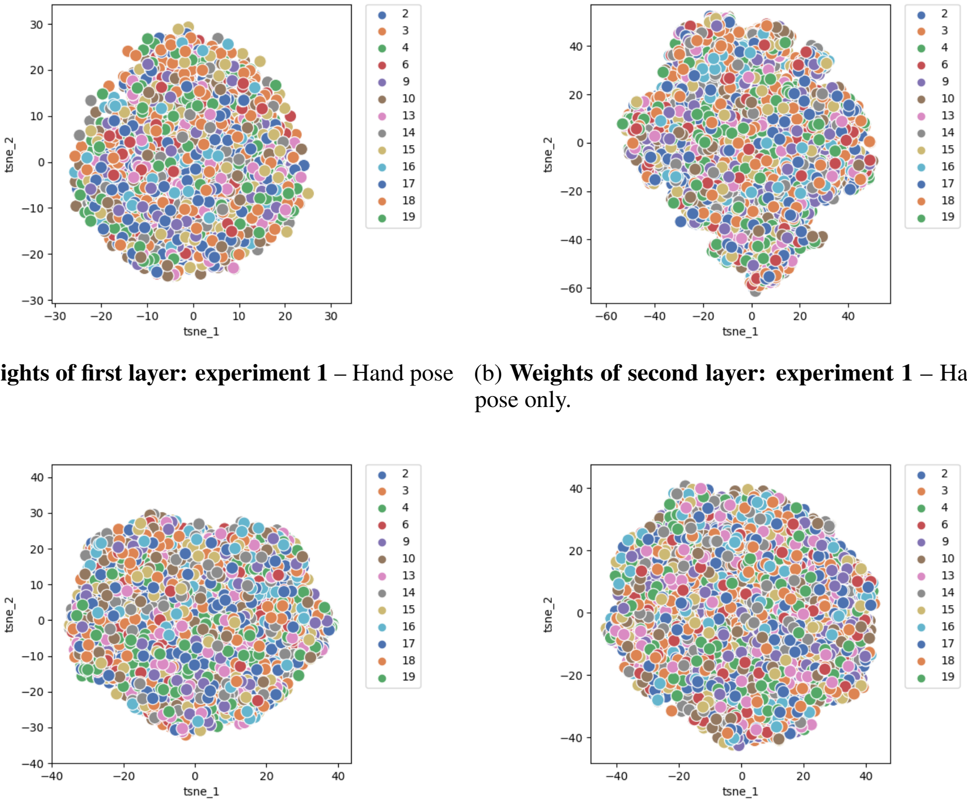
Table 3: **Average gradient norm per adaptation step: exp. 1 vs exp.** – Entries follow an *exp1/exp2* format, with larger values in bold.

Step	Tomato soup can	Banana	Scissors	Foam brick	Mug
1	204/**394**	231/**638**	270/**677**	195/**380**	236/**505**
2	193/**360**	219/**576**	255/**613**	185/**338**	225/**458**
3	182/**330**	207/**522**	242/**557**	176/**302**	214/**416**
4	173/**303**	197/**475**	229/**509**	167/**273**	205/**381**
5	164/**280**	187/**435**	217/**467**	159/**248**	195/**349**
6	155/**259**	178/**401**	207/**430**	152/**227**	187/**322**
7	148/**241**	169/**371**	197/**398**	145/**209**	179/**298**
8	141/**225**	161/**344**	188/**369**	139/**194**	172/**277**
9	134/**210**	153/**321**	180/**344**	133/**181**	165/**259**
10	128/**198**	146/**301**	172/**322**	127/**169**	159/**243**

C.2 Interpretation of the gradients norm during adaptation

We here provide evidence to assess the hypothesis that the adaptation phase produces smaller gradients for the object keypoints than for the hand keypoints. In Sec. 4.1, we make this postulate to conclude that the hand keypoint regression task should not be severely impacted.

In Tab. 3 we show the norm of the gradients during the adaptation phase. For both experiments, on the same randomly sampled test objects, the gradient norms of the first 10 adaptation steps are averaged over batches of the same task. Both models were trained on the same 6 objects. Experiment 2 has consistently much larger gradients, meaning that jointly regressing the hand and object poses requires deviating further away from the initialisation during adaptation. This could reflect a poor initialisation of meta-parameters.



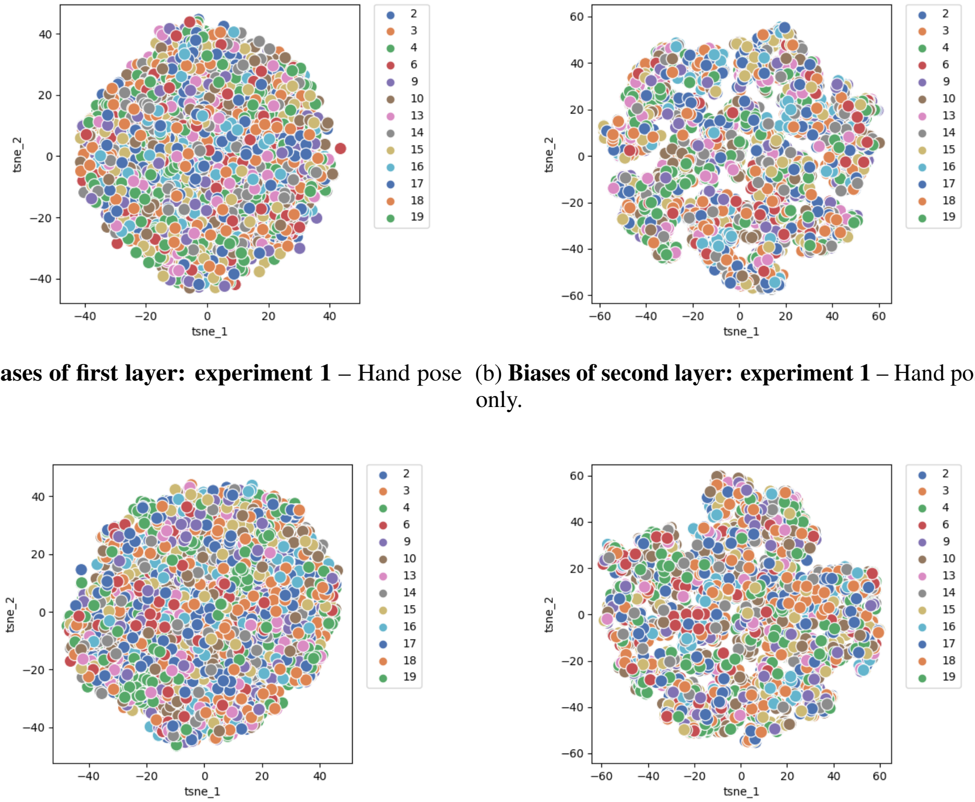
(a) **Weights of first layer: experiment 1** – Hand pose only.

(b) **Weights of second layer: experiment 1** – Hand pose only.

(c) **Weights of first layer: experiment 2** – Joint hand-object pose.

(d) **Weights of second layer: experiment 2** – Joint hand-object pose.

Figure 6: **Visualisation of weights per layer** – The t-SNE embeddings of the adapted weights of each layer, for both experiments, show no signs of separate clusters for the 13 different objects. This means that they mostly encode the same information regarding the hand or joint hand-object pose for all objects.



(a) **Biases of first layer: experiment 1** – Hand pose only.

(b) **Biases of second layer: experiment 1** – Hand pose only.

(c) **Biases of first layer: experiment 2** – Joint hand-object pose.

(d) **Biases of second layer: experiment 2** – Joint hand-object pose.

Figure 7: **Visualisation of biases per layer** – The t-SNE embeddings of the adapted biases of each layer, for both experiments, show that no structure is present in the first layer, while they appear to be clustered in some way for the last layer, the final 3D regression layer. However, those clusters do not coincide with the objects themselves, therefore the biases of the last layer may encode information specific to other aspects of the image features.