Developmental Bayesian Optimization of Black-Box with Visual Similarity-Based Transfer Learning

Amaury Depierre, Maxime Petit, Xiaofang Wang, Emmanuel Dellandréa, Liming Chen

Developmental Bayesian Optimization of Black-Box with Visual Similarity-Based Transfer Learning

1st Maxime Petit
LIRIS, CNRS UMR 5205
Ecole Centrale de Lyon
Ecully, France
maxime.petit@ec-lyon.fr

2nd Amaury Depierre
LIRIS, CNRS UMR 5205
Ecole Centrale de Lyon
Ecully, France
amaury.depierre@ec-lyon.fr

3rd Xiaofang Wang
LIRIS, CNRS UMR 5205
Ecole Centrale de Lyon
Ecully, France
xiaofang.wang@ec-lyon.fr

4th Emmanuel Dellandrea
LIRIS, CNRS UMR 5205
Ecole Centrale de Lyon
Ecully, France
emmanuel.dellandrea@ec-lyon.fr

5th Liming Chen
LIRIS, CNRS UMR 5205
Ecole Centrale de Lyon
Ecully, France
liming.chen@ec-lyon.fr

***Abstract*—We present a developmental framework based on a long-term memory and reasoning mechanisms (Vision Similarity and Bayesian Optimisation). This architecture allows a robot to optimize autonomously hyper-parameters that need to be tuned from any action and/or vision module, treated as a black-box. The learning can take advantage of past experiences (stored in the episodic and procedural memories) in order to warm-start the exploration using a set of hyper-parameters previously optimized from objects similar to the new unknown one (stored in a semantic memory). As example, the system has been used to optimized 9 continuous hyper-parameters of a professional software (Kamido) both in simulation and with a real robot (industrial robotic arm Fanuc) with a total of 13 different objects. The robot is able to find a good object-specific optimization in 68 (simulation) or 40 (real) trials. In simulation, we demonstrate the benefit of the transfer learning based on visual similarity, as opposed to an amnesic learning (*i.e.* learning from scratch all the time). Moreover, with the real robot, we show that the method consistently outperforms the manual optimization from an expert with less than 2 hours of training time to achieve more than 88% of success.**

***Index Terms*—developmental robotics, long-term memory, learning from experience, Bayesian Optimisation, transfer learning, automatic hyper-parameters configuration**

I. Introduction

Many algorithms and frameworks in the field of robotics require optimal parameter settings to yield strong performance (*e.g.* Deep Neural Networks [1], Reinforcement Learning [2]), whether they are used to move from one place to another or to grasp objects. These parameters can be manually optimized by a human expert, but this task is tedious and error-prone. Moreover, this solution is not viable in practice for situations where the hyper-parameters have to be defined frequently, such as for each subset or task (*e.g.* for each object to be grasped). To overcome these challenges, techniques have been developed to automatized high-level parameters search such as Bayesian Optimization [3], especially suited where the evaluation of the algorithm, treated as a blackbox function, is very expensive and noisy (which is the case for real world robotic grasping application). These automated configuration techniques are however commonly used before the deployment of the solution on a system, or launched manually when needed, separated from the autonomous "life circle" (*i.e.* offline) of the robotics platform from instance. Hence, the optimizations are starting from scratch each time (this is called a *cold-start*) without taking advantage of the previous experience of the system (*warm-start*) [4], as opposed to developmental framework that mights benefit from transfer learning.

Our contribution consist of a developmental cognitive architecture (composed of a long term memory and reasoning modules) allowing a robot to optimize by experience the parameters of a manipulation and/or vision algorithm (treated as black-box) where fine-tuning according to objects is needed. The learning procedure efficiency is increased by taking advantage of previous experiences (*i.e.* past optimization of similar objects). The framework will be tested in both simulation and with real robot.

This work was in part supported by the EU FEDER funding through the FUI PIKAFLEX project and in part by the French National Research Agency, l'Agence Nationale de Recherche (ANR), through the ARES labcom project under grant ANR 16-LCV2-0012-01.

II. Related Work

Bayesian Optimization (BO) [5], [6] has been used in the robotic field, especially for automatic gait optimization (*e.g.* [7]–[9]). Among others, Cully *et al.* developed a walking robot that can quickly adapt its gait after been damaged using an intelligent trial and error algorithm [10]. The robot is taking advantage of previous simulated experiences where legs were damaged, with the best walking strategies stored in a 6-dimensional behavioural space (discretized at five values for each dimension, representing the portion of time each leg is in contact with the ground). For our application, the behavioral space will have to represent the object similarity to optimize the search. However, such behavioural space cannot be objectively and numerically obtained because the concept of

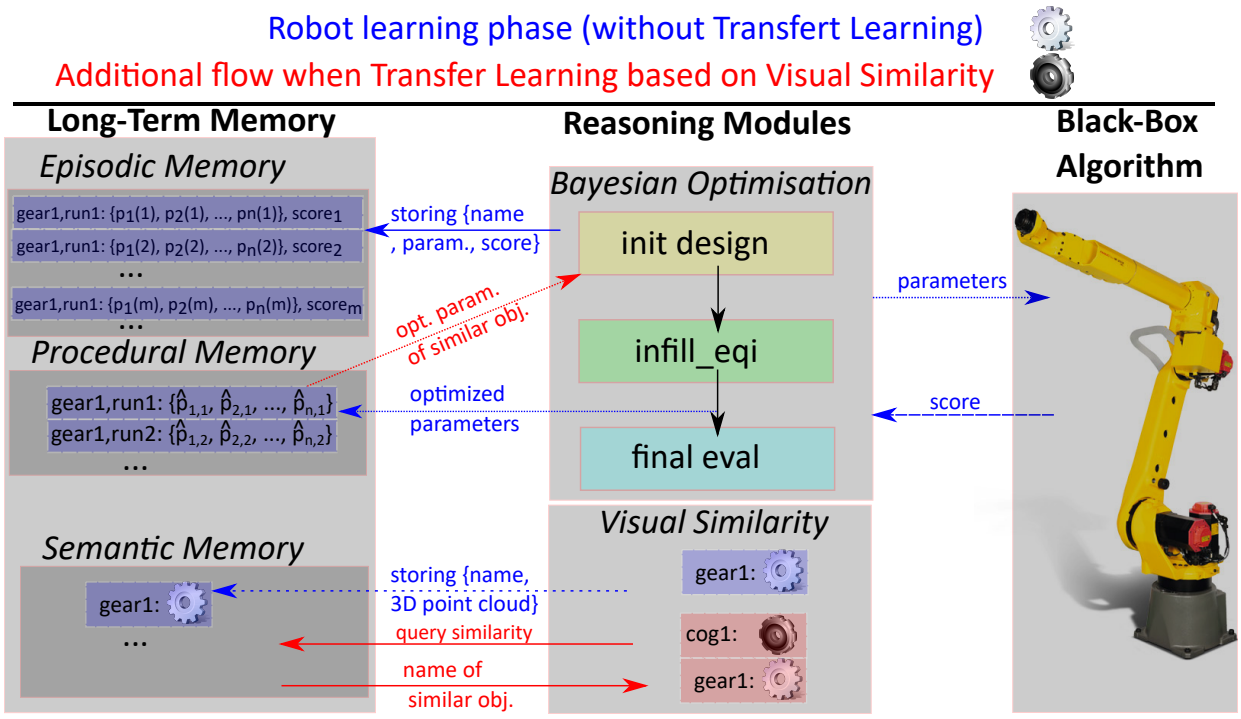


Fig. 1. Architecture of the cognitive developmental framework, based on Long-Term Memory (with episodic, procedural and semantic memories) and Reasoning Modules (Bayesian Optimisation and Visual Similarity) allowing a robot to learn how to grasp objects. This learning consists of guiding an efficient hyper-parameters optimization of black-box algorithm controlling the robot. The blue arrows represents the data flows during a learning phase without transfer learning (*i.e.* without taking advantage of the Long-Term memory, just storing the experiences). The red arrows shows the additional queries and exchanges of information during a learning phase with transfer learning, based on the visual similarity between objects the robot knows how to grasp, and a new one.

similarity is not easily interpretable into discrete dimensions. Regarding robot grasping tasks, the recent work of Nogueira *et al.* uses BO allowing a humanoid robot iCub to not only learn how to perform grasping, but also doing it safely (*i.e.* not sensitive to small errors during grasp execution) [11]. However, this method only concerns single object grasping (*i.e.* isolated objects) where they can be grasped in a collision-free environment from every direction, which is not the case for our context.

Transfer Learning is commonly used to reduce the time needed for a real robot to acquire new skills, usually involving reinforcement learning domain (see Taylor *et al.* for a review [12]). In particular, the most recent work is from Breyer *et al.* with a sim-to-real transfer based reinforcement learning for a grasping robot [13] which also has the restriction that the objects needs to be isolated. The idea of transfer learning has also been applied for Bayesian Optimisation techniques: Feurer *et. al.* [14] used meta-learning consisting of speeding up the BO by starting from promising optimisation results that performed well on similar datasets. We are following here the same strategies but applied on real robotics using the similarity between objects.

Among the work on **Long-Term Memory** for robots (see review from Wood *et al.* [15]), some studies were using the memory to improve the learning. Recently, a lifelong autobiographical memory has been proposed for the humanoid robot iCub [16] allowing reasoning modules to stores and collect multi-modal data. Initally focusing on the declarative (episodic and semantic) memory, the framework has been extended with a procedural memory [17] (where skills are stored) of pointing actions. A framework integrating declarative episodic and procedural memory systems for combining experiential knowledge with skillful know-how has also been proposed in [18], based on joint perceptuo-motor representations. However, the procedural memory consist in a simple repository of pre-defined elementary actions (reach, push, grasp, locomote and wait) instead of growing flexible and adaptable skills.

III. Methodology

A. Architecture Overview

The architecture of the cognitive system (see Fig. 1) relies on an Long-Term Memory that stores information in 3 different sub-memories (as described by Tulving [19]): 1) personally experiences events, that can be localized precisely in time, in the *episodic memory*, 2) motor skills or action strategies in the *procedural memory* and 3) facts and knowledge about concepts and objects in the *semantic memory*. The information stored in such memories is built and accessed by reasoning modules (a Bayesian Optimisation, see Sec.III-B, and a Visual Similarity component, see Sec.III-C). This allows the robot to take advantage of the growing knowledge acquires from experiences. Indeed, We consider as a single experience the performance score given by the robot (*i.e.* a percentage of successful grasps) using a specific set of hyper-parameters that was chosen by the Bayesian Optimisation module to be explored. The *episodic memory* is thus composed by tuples with the label of the object, the set of hyper-parameters used and the obtained performance. The best set of hyper-

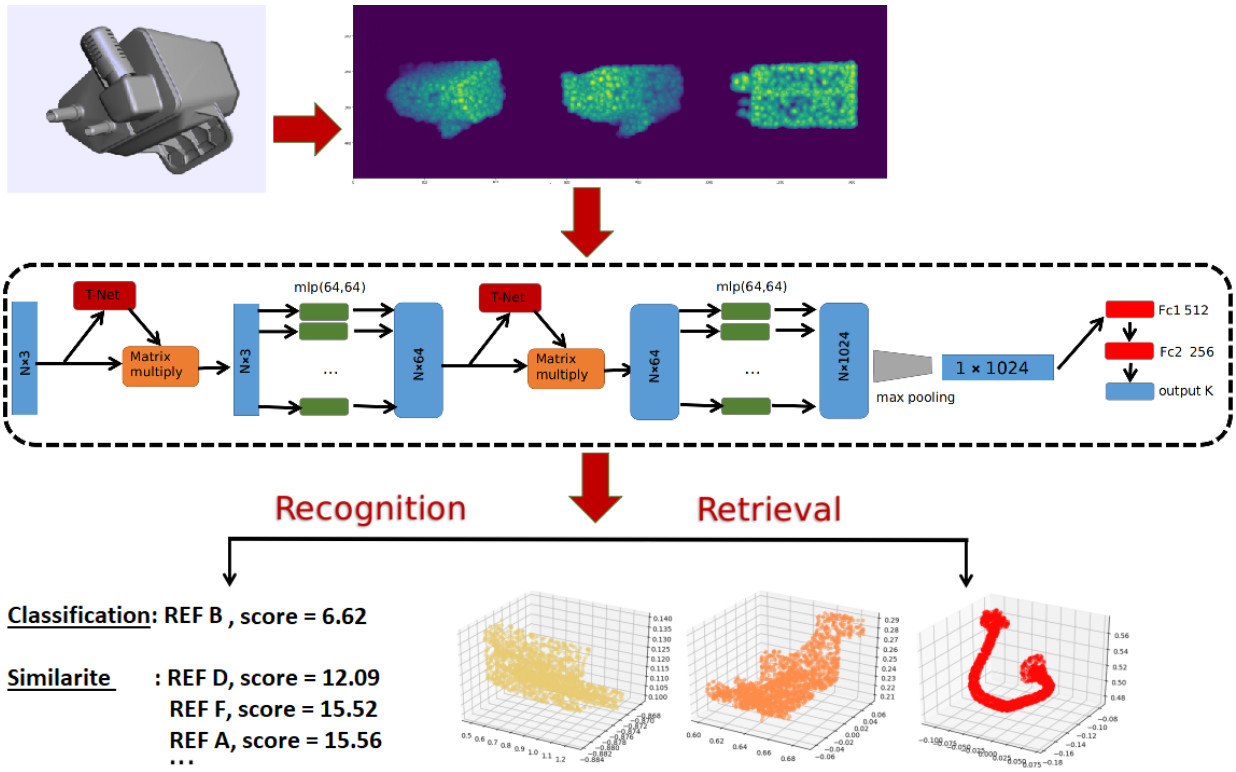


Fig. 2. Main architecture of the Visual Similarity module. A 3D CAD model is first sampled randomly and normalized into set of points (x, y, z), then it is fed into a deep neural network based on PointNet, which learns a global geometry shape by aggregating results of all points into a 1024 feature vector.

parameters at the end of a run (*i.e.* an full session of an Bayesian Optimisation) are stored in the *procedural memory*, along their performance and the corresponding object. Each set represent a strategy for analysing the scene and grasping the object that has been defined by and is thus adapted to the robot used. The *semantic memory* focuses here on the visual component where the 3D points clouds of objects are stored with their names by the Visual Similarity module. This module is also able to provide the more similar objects known by the robot against another one. The Bayesian Optimisation module queries and push data from and to the episodic and procedural memory of the robot. Indeed, the transfer learning will be employed for manipulation tasks on objects (*e.g.* grasping) where similar domains for the transfer implies similar objects to be manipulated. The combination of both these reasoning modules gives to the robot the capability of Transfer Learning: when confronted to a new object, the robot will first use the Visual Similarity component to extract the labels of known objects with similar shape. It is then able to access the sets of optimized hyper-parameters for these objects in order to force the Bayesian Optimisation module to explore such strategy during the initial design.

B. Bayesian Optimisation module

For the BO module, we are using the R package *mlrMBO*[1] [20] using Gaussian Processes (GP), also known as the Kriging model [21] as surrogate model. It is the most often used surrogate model to estimate both the fitness (used to exploit) and the uncertainty (used to explore) [22]. A BO run will provide a single optimized set of parameters and can be decomposed in 3 main parts:

- initial design (*"init_design"*): select points independently to draw a first estimation of the objective function.
- Bayesian search mechanism (*"infill_eqi"*), balancing exploitation and exploration, where the next point is extracted from the acquisition function (constructed from the posterior distribution over the objective function) with the Expected Quantile Improvement (EQI) criteria from Pichney *et al.* [23]. It is an extension of the Expected Improvement (EI) criteria for heterogeneously noisy functions, where the improvement is measured in the model rather than on the noisy data.
- final evaluation (*"final_eval"*), where the best predicted set of hyperparameters (prediction of the surrogate, which reflects the mean and is less affected by the noise) is used several times in order to provide a distribution of the optimization results.

It has to be noted that the BO is dealing with n hyper-parameters $p_1, p_2, ..., p_n \in [0:1]$, which are transformed using the boundaries of the hyper-parameters before sending them to the robot. The BO module then launches the robot in order to achieve a task (*e.g.* try to grasp X times an object) which provides, in return, a performance score $s_i \in [0:100]$ (*i.e.* the percentage of successful grasps among K attempts), where i is the iteration number.

C. Visual Similarity module

The Visual Similarity module (see Fig. 2) is retrieving the most similar objects from semantic memory, *i.e.* a retrieval

[1]https://github.com/berndbischl/mlrMBO

and classification system learned from a reference database, where each reference's parameters has been optimized. To train such semantic memory, we apply a deep learning neural network, such as PointNet [24] on 3D model. PointNet is a deep Learning method for 3D Classification and Segmentation, designed to learn point clouds geometrical shape in 3D. It takes the coordinates of N points as input, that are transformed with an affine transformation matrix by a mini-network called T-Net. Then, each point is learned by convolutional kernel (size of 1), and finally aggregated by symmetric operations (*i.e.* max-pooling), into a global feature of dimension 1024, followed that, three fully connected layers are stacked on to learn the object classes.

To learn the semantic memory, our reference 3D CAD models are represented first with point clouds. Then the last layer in PointNet is modified into our reference database (change number of classes), and we further fine-tune the PointNet by freezing earlier layers of *conv5*, based on pre-trained model from ModelNet40 [25], which turns out a satisfactory accuracy and efficiency.

To retrieve the most similar models from a new given 3D model, we first sample the new object into point clouds (N points of (x, y, z)), then we extract its global feature with 1024 dimensions, and finally we calculate its pair of distance between each reference model, with the most similar reference corresponding to the minimal distance.

D. Memory

The memories are stored in a PostgreSQL[2] database similarly to other implementation of long-term memory system [16], [26]. The episodic memory stores for each iteration i of each run r the label of the object, the set of n hyper-parameters $\{p_1(i), p_2(i), ..., p_n(i)\}$ tested and the score s_i of the performance. The procedural memory is only composed by the best set of hyper-parameters for each run r, $\{\hat{p}_{1,r}, \hat{p}_{2,r}, ..., \hat{p}_{n,r}\}$, along the object name learned in the run. The semantic memory contains the visual information (stored as points clouds) about the objects in order to recognize them.

E. Transfer Learning

The transfer learning process is implementing a human-like strategy, as proposed by Feurer *et. al.* [14] and called meta-learning, which consists of warm-starting the BO by recovering previously optimized set of hyper-parameters for similar tasks (*i.e.* similar datasets for [14]) and try them before the bayesian search mechanism. While similarity was defined by the human in their work, we will use our Visual Similarity module to provide the most similar known object (compared to a new one) which allows the robot to query its procedural memory for such object, and X different sets of optimized hyper-parameters (obtained from X different runs) are transmitted to the BO module. They will be tested at the end of the the "init_design", where the other previous iterations are still proposed by the maximimLHS function. It has to be noted that in order to estimate the effect of the transfer learning, the number of total points during the "init_design" will be fixed.

[2]http://www.postgresql.org/

IV. Experiments

The task is to grasp 15 times from an homogeneous cluttered bulk composed of the same object instance (simulated or real, see Fig. 3). We are using a professional software called Kamido[3] from the company Sileane as a black-box controlling a parallel-jaw gripper robot that need to get parametrized for each object to manipulate. We will confront the framework with the optimization of 9 continuous hyper-parameters, used by Kamido to 1) analyse the images from the scene and 2) define a proper grasping target.

For the experiment in simulation, we authorize a finite budget (to be able to compare each run with the same learning conditions) of 80 iterations for the BO process, with a decomposition in 18/50/12 iterations for the 3 steps (9 parameters). The *init_design* points are selecting using a Maximin Latin Hypercube function [27] to maximize the minimum distance between points in order to cover space as much as possible (initially forcing the exploration). The GP's kernel is a Matern 3/2 classically used in machine learning (as explained in [28], p85). The EQI criterion is set with a quantile level $\beta = 0.7$. It has been experimentally obtained, and allows the algorithm to increase the optimization at the *infill_eqi* step after reaching a limit during the initial design step, as shown in Fig. 4, when considering 5 different objects and optimized multiple times (between 3 to 12 runs). To optimize the infill criterion, we are using a Covariance Matrix Adapting Evolutionary Strategy (CMA-ES) [29], [30] from the package *cmaes*[4]. It is a stochastic derivative-free numerical optimization algorithm for difficult (non-convex, ill-conditioned, multi-modal, rugged, noisy) optimization problems in continuous search spaces.

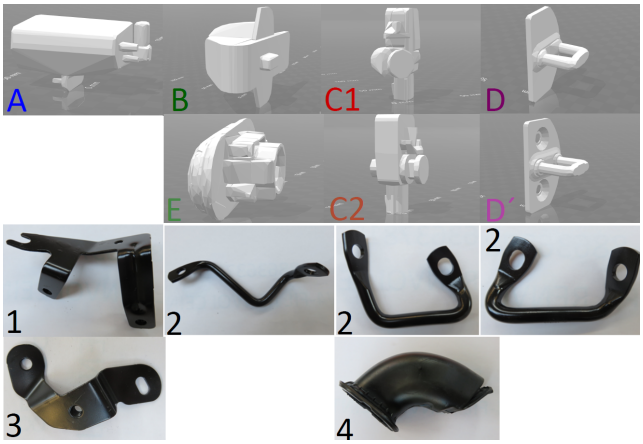


Fig. 3. Reduced CAD models of the objects used in simulation (top) and pictures of the objects used in experiments with a real robot (bottom).

V. Results

A. From simulated Robot

The simulator is based on the real-time physics pyBullet. Objects are loaded in the environment from their CAD model,

[3]http://www.sileane.com/en/solution/gamme-kamido
[4]https://cran.r-project.org/package=cmaes

on which we apply a volumetric hierarchical approximate convex decomposition [31] in order to reduce the complexity of the collisions.

Simulation results will be shown using curves and boxplots. For each run we calculate the third quartile (Q3) of the already explored points at each iteration and compute their maximum up to now for each run (instead of the maximum directly because the evaluation is very noisy), before eventually calculating the mean among the repetitions (*i.e.* same objects or With/Without transfer learning depending on the figures). The curves corresponds to a smoothing among the obtained points, using the non-parametric LOcally weighted RegrESSion (*i.e.* "loess") method [32]. We also reinitialized at the beginning of each new step (*i.e.* initial design, Bayesian optimisation using EQI criteria, final evaluation) to have independent points (*e.g. final_eval* curves are not influence by the previous step).The boxplot are representing the mean of all the final performances of the optimized parameters for each run, grouped by their corresponding object.

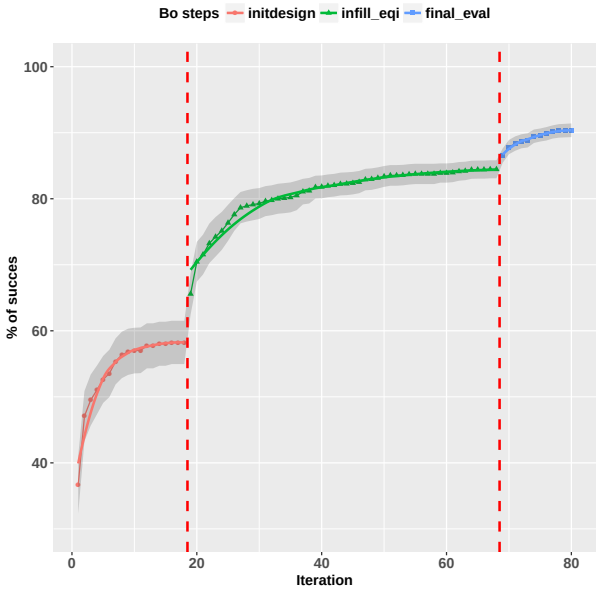


Fig. 4. Mean of the max(Q3) of the % of success for all objects and runs combined (without transfer learning), and for each BO steps, with EQI's $\beta = 0.7$

1) Experiment 1, without Transfer Learning: We show here how well the robot can develop its procedural memory to manipulate objects by optimizing its grasping hyper-parameters without using any prior information (*i.e.* a "cold-start", without transfer learning) We have tested 5 different objects in simulation: A, B, C1, C2 (*i.e.* very similar to C1) and D (see Fig. 3). We have several repetitions for each objects, in order to check the robustness of the system, respectively 6, 6, 12, 3 and 12.

Fig. 4 shows an overview of the BO performance during the iterations, split among the 3 parts (*i.e. init_design*, *infill_eqi*, *final_eval*), with objects and runs without transfer learning combined. It demonstrates that the robot benefit from the BO *infill_eqi* part with an increase of the performance thanks to the smart exploration and exploitation trade-off of the method.

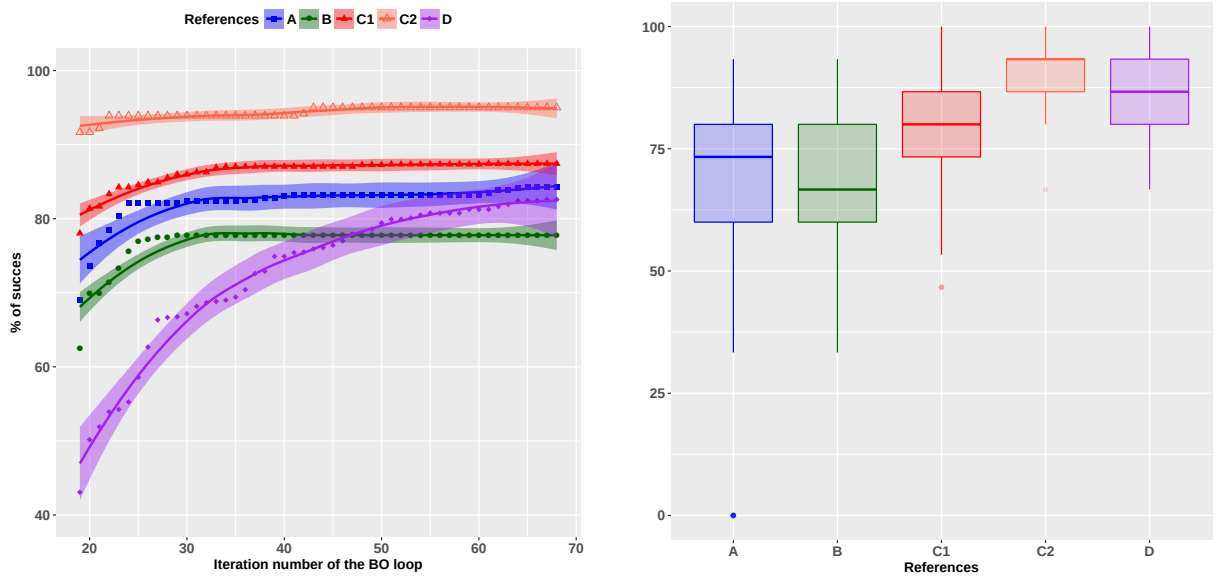
Fig. 5 provides a detailed look at the results, by 1) focusing on the *infill_eqi* process and on the final evaluations, and 2) by splitting the results among the different object. In Fig. 5(a) we show the consistent increasing of performance during the 50 iterations of the BO *infill_eqi* part. The object D is the hardest to optimize and need the full 50 iterations to reach a maximum, where for the other objects the method could find one in roughly 20 iterations only. Fig. 5(b) shows the final results of the obtained optimized sets of parameters for each run, grouped by objects, and completed by numerical results from Table II. Object B is more difficult than the others but still achieve a decent optimization with a median score of 66.67% (all runs combined) and 73.34% (best run). The optimization of the objects A, C1, C2 and D are better, with median score of respectively 73.33%, 80%, 93.33% and 86.67% (all runs) and 83.33%, 86.67%, 93.33% and 90% (best run).

2) Experiment 2, with Transfer Learning: In these experiments, we demonstrate the effectiveness of the Visual Similarity component allowing the robot to take advantage of its acquired procedural memory to optimize faster and better with a Transfer Learning strategy when confronted to new objects. We define the number of configuration coming from such method to 3, with the parameter sets proposed by the maximinLHS method reduced to 15 in order to keep an "init_design" phase with the same number of explored strategies.

We exemplify the concept of visual similarity and transfer learning on object D: as all other objects, the simulated model has been obtained from the volumetric hierarchical approximate convex decomposition of the original CAD model in order to reduce the complexity of collisions. For the transfer learning, we use the optimized sets of hyper-parameters obtained from experiments with the unmodified version of the object, called D', in order to shows the effect of transfer learning when using experience from very similar object. The objects A, B and C1 will be also optimized using the procedural memory of previously learned objects that are the most similar to them, respectively C2, C2 and E (corresponding scores are shown in Table I). It has to be noted that whereas the pairs D/D' and C1/C2 are highly similar, the couples A/C2 and B/E are more loosely related.

Fig. 6 shows the performance during the BO iteration where the runs are grouped between cold-start (*i.e.* without transfer learning, object A, B, C and D) and with the transfer learning based on similarity (*i.e.* A_TL_C2, B_TL_E, C1_TL_C2 and D_TL_D'). At the beginning of the "infil_eqi" part, the robot is performing better with transfer learning, and is reaching higher optima. These results are confirmed in Fig. 7 where the runs are grouped among their respective objects (including the differentiation between cold-start and with transfer learning, in pastel). The optimization is consistently faster to converge and with a higher optima when using transfer learning compared to without: the robot explores in a more efficient manner.

As shown in Table II the optimization is better overall (*i.e.* all runs for each object grouped together) with transfer learning as opposed to without for every tested object. This



(a) Mean among the runs for each object of the Max(Q3) of the % of success, when considering the *infill_eqi* step.

(b) Boxplot showing the %of success of the optimized parameters, during the *final_eval* step.

Fig. 5. Detailed results of the cold-start experiences from simulation, with an overview of the different performances for each object. Best viewed in colors.

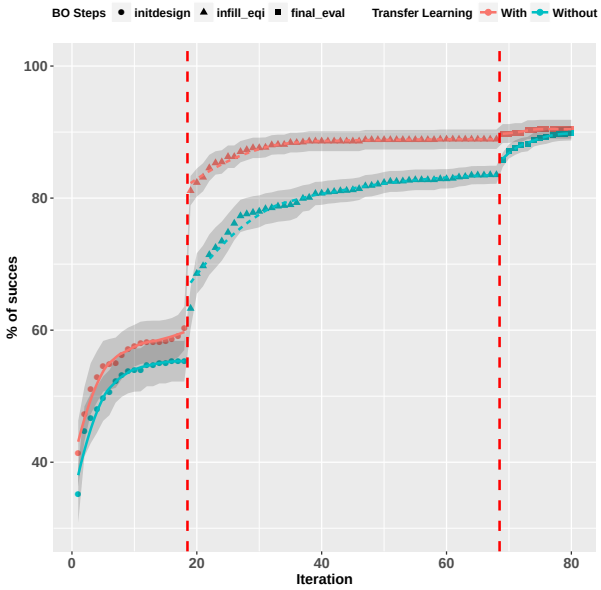


Fig. 6. Comparison With/Without Transfer Learning: Mean of the max(Q3) of the % of success for all objects & runs combined, and for each BO steps. Best viewed in colors.

benefit is confirmed even if we consider only the best run per objects: when conditions are equals (6 runs for the object A and B, with/without transfer learning) the best run from transfer learning achieves a better performance than the best run from without transfer learning. On the other hand, with transfer learning, the best optimized set of hyper-parameters achieve similar performance than the corresponding counterpart without transfer learning, despite having twice less independent runs.

TABLE I
MOST SIMILAR OBJECT KNOWN TO THE ROBOT COMPARED TO A REQUESTED OBJECT, WITH THE CORRESPONDING SIMILARITY

Object Requested	Most Similar Object	Similarity Score
A	C2	16.35
B	E	12.09
C1	C2	11.63

TABLE II
OPTIMIZATION RESULTS FROM SIMULATION WITH TRANSFER LEARNING

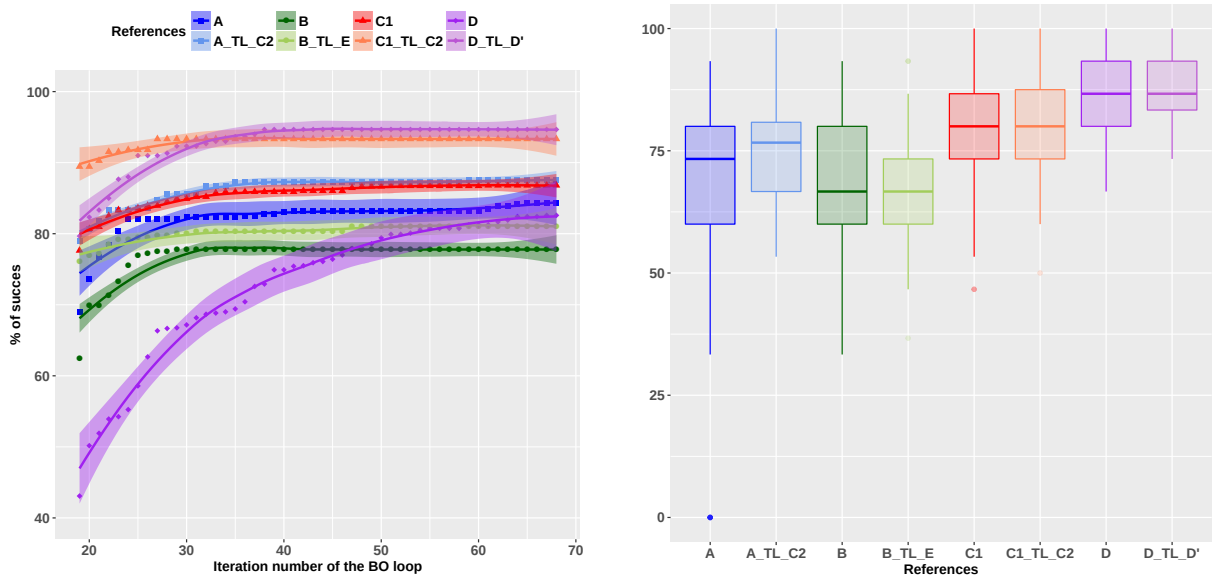
Ref.	Nb Runs	% success (all run, mean±sd, median)	% success (best run, mean± sd, median)
A	6	65.47±27.3, 73.33	78.89±11.31,83.33
A_TL_C2	6	**76.11±10.19, 76.67**	**82.78±9.93**, 83.33
B	6	68.01±12.89, 66.67	72.78±14.34, 73.34
B_TL_E	6	**68.11±12.17**, 66.67	**73.06±13.52, 76.67**
C1	12	78.95±10.87, 80	**83.89±7.63**, 86.67
C1_TL_C2	6	**81.30±11.04**, 80	82.5±11.82, **88.33**
D	12	86.92±9.45, 86.67	**91.11±8.21**, 90
D_TL_D'	6	**87.33±7.44**, 86.67	90.56±7.76, 90
C2	3	91.76±6.74, 93.33	95±4.14, 93.33

B. From real Robot

In this experiment, we will compare the results of the set of hyperparameters optimized from the Bayesian algorithm to the one manually defined by an expert after a day of tuning, without any transfer learning capabilities. We will use 4 different items (see Fig. 3).

We are using a 6-DOF industrial robotic arm FANUC M-20iA/12L[5] with parallel gripper. In order to keep the training time relatively short (*i.e.* less than 2 hours), we are reducing the BO to a total of 45 iterations (20, 20 and 5 iterations respectively for design, Bayesian loop and final evaluation). Table III shows that our optimization method outperforms the expert settings consistently on the 4 tested items, with at least

[5]http://www.fanuc.eu/fr/en/robots/robot-filter-page/m-20-series/m-20ia-12l



(a) Transfer Learning: Mean among the runs for each object of the Max(Q3) of the % of success, during the *infill_eqi* step.

(b) Transfer Learning: Boxplot showing the %of success of the optimized parameters, during the *final_eval* step.

Fig. 7. Detailed results of the comparison between cold-start and warm-start (*i.e.* without and with transfer learning) experiences from simulation, with an overview of the different performances for each object and learning. Best viewed in colors.

82% of successful grasp. Moreover, item 2 is a mix of different but similar objects put together in an heterogeneous bulk, and the robot is showing robustness, by still being able to learn a common strategy to grasp them all.

TABLE III
OPTIMIZATION FROM AN EXPERT VS ALGORITHM

Ref. Nb.	% success (expert)	% success (method)	Time per iter. (sec)	Tot. training time (min)
1	80.0	**88.0**	133.8±21.5	100.4
2	88.2	**88.6**	124.3±49.3	93.2
3	74.3	**82.2**	160.8±48.7	120.6
4	86.8	**91.7**	132.9±25.2	99.6

VI. CONCLUSION AND FUTURE WORK

We have shown how the robot can take advantage of its experience and its long-term memory to perform transfer learning when objects are similar, both in simulation and with a real robot. In simulation, with a fixed budget of 68 trials (over which the last 50 are defined by the evaluation function of a Bayesian optimization method), we are able to optimize for each 5 objects 9 continuous hyper-parameters of an industrial grasping algorithm and achieve good performance, despite the fact that the evaluation is noisy. Indeed, we achieve a mean percentage of grasping success between 73% and 95% in simulation (depending on the object) with a faster convergence with transfer learning. The method has also been tested in an experiment with a real robot, where the framework has been confronted to an expert. The autonomous method provides better optimization than the manually exploration from the expert, with between 82.2% and 91.7% of success over 4 objects (representing an increase of success between 0.4% and 8% compared to the expert's optimization) despite a smaller exploitation budget (20 iterations instead of 50) to keep the optimization process below 2 hours.

In this work, we have implemented a transfer learning where a similar experience (*i.e.* grasping in simulation) of an object O was used for the learning of a similar object O', with the same conditions. Another type of transfer learning might be implemented in a future work: use the experience in simulation with an object O in order to warm-start the exploration of the same object but in reality, such as studied with the balancing a cart-pole [33] or grasping problems [13]. The current method is focusing on a noisy single objective optimization, where the score is solely based on the performance in terms of success. Future work might target to extend the system for optimizing multi-objective function, where for instance the speed of execution is an additional component to take into account. Pareto front of multiple criteria will then has to be considered using additional method provided by the *mlrMBO* R-package [34], [35]. Another possible benefit of our framework is that the exploration and storing of different optimized set of hyper-parameters can lead to embodied symbols or concepts emergence [36]. By providing human labels about the physical aspect of objects (*e.g.* "heavy", "flat"), co-occurences can be detected between these adjectives and a sub-set of optimized parameters, in a similar way that done to discover pronouns [37] or body-parts and basic motor skills [38]. In return, this might lead to another form of transfer learning, when some hyper-parameters will be directly extracted and fixed based on description labels of new objects provided by the human.

REFERENCES

[1] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information*

processing systems, 2012, pp. 2951–2959.
[2] T. Rückstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber, "Exploring parameter space in reinforcement learning," *Paladyn*, vol. 1, no. 1, pp. 14–24, 2010.
[3] J. Mockus, "Application of bayesian approach to numerical methods of global and stochastic optimization," *Journal of Global Optimization*, vol. 4, no. 4, pp. 347–365, 1994.
[4] D. Yogatama and G. Mann, "Efficient transfer learning method for automatic hyperparameter tuning," in *Artificial Intelligence and Statistics*, 2014, pp. 1077–1085.
[5] J. Mockus, "The bayesian approach to local optimization," in *Bayesian Approach to Global Optimization*. Springer, 1989, pp. 125–156.
[6] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.
[7] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans, "Automatic gait optimization with gaussian process regression." in *IJCAI*, vol. 7, 2007, pp. 944–949.
[8] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1-2, pp. 5–23, 2016.
[9] B. Yang, G. Wang, R. Calandra, D. Contreras, S. Levine, and K. Pister, "Learning flexible and reusable locomotion primitives for a microrobot," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1904–1911, 2018.
[10] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, p. 503, 2015.
[11] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, "Unscented bayesian optimization for safe robot grasping," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1967–1972.
[12] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633–1685, 2009.
[13] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Flexible robotic grasping with sim-to-real transfer based reinforcement learning," *arXiv preprint arXiv:1803.04996*, 2018.
[14] M. Feurer, J. T. Springenberg, and F. Hutter, "Initializing bayesian hyperparameter optimization via meta-learning." in *AAAI*, 2015, pp. 1128–1135.
[15] R. Wood, P. Baxter, and T. Belpaeme, "A review of long-term memory in natural and synthetic systems," *Adaptive Behavior*, vol. 20, no. 2, pp. 81–103, 2012.
[16] M. Petit, T. Fischer, and Y. Demiris, "Lifelong augmentation of multimodal streaming autobiographical memories," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 3, pp. 201–213, 2016.
[17] ——, "Towards the emergence of procedural memories from lifelong multi-modal streaming memories for cognitive robots," 2016.
[18] D. Vernon, M. Beetz, and G. Sandini, "Prospection in cognition: the case for joint episodic-procedural memory in cognitive robotics," *Frontiers in Robotics and AI*, vol. 2, p. 19, 2015.
[19] E. Tulving, "Memory and consciousness." *Canadian Psychology/Psychologie canadienne*, vol. 26, no. 1, p. 1, 1985.
[20] B. Bischl, J. Richter, J. Bossek, D. Horn, J. Thomas, and M. Lang, "mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions." [Online]. Available: http://arxiv.org/abs/1703.03373
[21] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science*, pp. 409–423, 1989.
[22] A. I. J. Forrester, A. J. Keane, and N. W. Bressloff, "Design and analysis of" noisy" computer experiments," *AIAA journal*, vol. 44, no. 10, pp. 2331–2339, 2006.
[23] V. Picheny, D. Ginsbourger, Y. Richet, V. Picheny, D. Ginsbourger, and Y. Richet, "Optimization of Noisy Computer Experiments with Tunable Precision," *Technometrics*, vol. 55, no. 1, pp. 2–13, 2013.
[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.
[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.
[26] G. Pointeau, M. Petit, and P. F. Dominey, "Successive developmental levels of autobiographical memory for learning through social interaction," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 3, pp. 200–212, 2014.
[27] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
[28] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.
[29] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 1996, pp. 312–317.
[30] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
[31] K. Mamou, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, E. Lengyel, Ed. A K Peters, 2016, pp. 141–158.
[32] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American statistical association*, vol. 83, no. 403, pp. 596–610, 1988.
[33] A. Marco, F. Berkenkamp, P. Hennig, A. P. Schoellig, A. Krause, S. Schaal, and S. Trimpe, "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1557–1563.
[34] "Multi-objective parameter configuration of machine learning algorithms using model-based optimization," *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2017.
[35] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, "Model-based multi-objective optimization: taxonomy, multi-point proposal, toolbox and benchmark," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 64–78.
[36] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, and H. Asoh, "Symbol emergence in robotics: a survey," *Advanced Robotics*, vol. 30, no. 11-12, pp. 706–728, 2016.
[37] G. Pointeau, M. Petit, G. Gibert, and P. F. Dominey, "Emergence of the use of pronouns and names in triadic human-robot spoken interaction," in *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*. IEEE, 2014, pp. 146–152.
[38] M. Petit and Y. Demiris, "Hierarchical action learning by instruction through interactive grounding of body parts and proto-actions," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3375–3382.