Quality Diversity under Sparse Interaction and Sparse Reward: Application to Grasping in Robotics

Johann Huber, François Hélénon, Miranda Coninx, Faïz Benamar, Stephane Doncieux

Quality Diversity under Sparse Interaction and Sparse Reward: Application to Grasping in Robotics

Johann Huber johann.huber@isir.upmc.fr
Sorbonne Université, CNRS, ISIR*, Paris, 75005, France

François Helenon francois.helenon@isir.upmc.fr
Sorbonne Université, CNRS, ISIR*, Paris, 75005, France

Miranda Coninx miranda.coninx@isir.upmc.fr
Sorbonne Université, CNRS, ISIR*, Paris, 75005, France

Faïz Ben Amar faiz.ben-amar@isir.upmc.fr
Sorbonne Université, CNRS, ISIR*, Paris, 75005, France

Stéphane Doncieux stephane.doncieux@isir.upmc.fr
Sorbonne Université, CNRS, ISIR*, Paris, 75005, France

Abstract

Quality-Diversity (QD) methods are algorithms that aim to generate a set of diverse and high-performing solutions to a given problem. Originally developed for evolutionary robotics, most QD studies are conducted on a limited set of domains – mainly applied to locomotion, where the fitness and the behavior signal are dense. Grasping is a crucial task for manipulation in robotics. Despite the efforts of many research communities, this task is yet to be solved. Grasping cumulates unprecedented challenges in QD literature: it suffers from reward sparsity, behavioral sparsity, and behavior space misalignment. The present work studies how QD can address grasping. Experiments have been conducted on 15 different methods on 10 grasping domains, corresponding to 2 different robot-gripper setups and 5 standard objects. An evaluation framework that distinguishes the evaluation of an algorithm from its internal components has also been proposed for a fair comparison. The obtained results show that MAP-Elites variants that select successful solutions in priority outperform all the compared methods on the studied metrics by a large margin. We also found experimental evidence that sparse interaction can lead to deceptive novelty. To our knowledge, the ability to efficiently produce examples of grasping trajectories demonstrated in this work has no precedent in the literature.

Keywords

Quality diversity, Sparse reward, Sparse behavior, Grasping, Evolutionary robotics.

1 Introduction

Quality-Diversity (QD) methods are evolutionary algorithms that optimize both diversity and quality to generate large repertoires of high-performing solutions to a given problem (Pugh et al. (2016), Cully and Demiris (2017)). This field produced significant results in evolutionary robotics, including recovery from injury (Cully et al. (2015)), generation of adversarial objects for robotic grasping (Morrison et al. (2020)), or morphological evolution (Zardini et al. (2021)). The recent rise of interest led to novel interactions between fields, with notable combinations

* Institut des Systèmes Intelligents et de Robotique

of QD and Reinforcement-Learning (Sigaud (2022)) or with supervised learning (Macé et al. (2023)). Numerous ideas are explored to further carry the field regarding the method (Fontaine and Nikolaidis (2021), Faldor et al. (2023)) or trying to address more complex tasks (Anne and Mouret (2023), Flageat and Cully (2023)).

Interestingly, **most QD domains for evolutionary robotics are tasks in which the fitness and the behavioral functions deliver non-constant signals, making them exploitable**. Those domains usually involve navigation, where it is easy to design (Faldor et al. (2023)) or to automatically learn (Paolo et al. (2020)) a behavior space that expresses the agent's displacement. Even if the fitness function is usually orthogonal to the targeted task – e.g. energy minimization while trying to generate locomotion policies – the function is always defined such that the algorithm can continuously optimize both the diversity and the quality of the processed solutions.

Recently, Paolo et al. (2021) studied tasks submitted to sparse fitness, proposing a QD algorithm that optimizes quality despite the limited reward signal. However, all the considered tasks involve the navigation of an entity within the environment (the agent itself or a ball to push somewhere): a behavioral space defined as the key entity's last position gives the targeted task's complete information. **Such a behavioral characterization can be said *complete*, as it provides the guarantee that the optimal solution will eventually be found** (NS assumption of uniform exploration (Doncieux et al. (2019))) **and that the algorithm can always rely on an exploitable behavioral signal throughout the evolutionary process** (see section 3.3.1). This work argues that **some challenging tasks like robotic manipulation ones cannot be addressed under such a reliable behavioral characterization**.

Grasping refers to making an agent pick an object by applying forces and torques on its surface. Considered a prerequisite for many manipulation tasks (Hodson (2018)), the sparsity of grasping's reward makes data-oriented approaches struggle in these domains. Despite efforts from many research communities, grasping is still only partially solved (Zhang et al. (2022)). An essential matter for solving grasping with learning methods is the ability to generate demonstrations that can bootstrap learning (Wang et al. (2021), De Coninck et al. (2020)). **The present work shows that QD methods can be reliably leveraged to generate a large set of diverse high-performing solutions that fulfill this need for high-quality data**.

While defining the proper behavioral characterization is not trivial, we here consider the first Cartesian position of the end effector when touching the object for the first time. By randomly initializing robotic arm policies, a significant part of the trajectories do not even touch the object. Therefore, those evaluations do not provide any exploitable behavioral information. Grasping is therefore not only submitted to sparse reward but also to *sparse interactions*. Plus, a behavioral characterization that guarantees that successful solutions will eventually be found is hardly designable. To our knowledge, **this work is the first that demonstrates QD algorithms capabilities to solve sparse fitness and sparse interaction problems without relying on an aligned behavior space.**

This paper aims to **study how the QD literature can scale up to sparse reward and sparse interaction tasks**, by comprehensively studying **how QD can be applied to the yet unsolved task of grasping**. This investigation raises many questions on QD, including the role of the behavioral characterization, the taxonomy of QD methods, and the evaluation procedure. It also leads to key insights on how QD methods perform on tasks submitted to sparse rewards and interaction, and bring to light the critical algorithmic components that make a QD method work in this context. Finally, this study led to insights into how QD methods can do efficient exploration in sparse interaction problems, showing that contrary to tasks where the behavior function is dense, novelty-driven approaches have poor exploration capabilities on sparse interaction problems.

Our contributions are the following:

- We propose a taxonomy to avoid ambiguities when talking about QD methods, especially regarding NS-related methods;
- We discuss the role of the behavioral characterization through the notions of *behavioral alignment*, *density of the behavioral function*, *behavioral completeness* and *driving/describing behavior space*;
- We introduce a simple framework that distinguishes the evaluation of a QD method from its internal components;
- We show that a simple variant of MAP-Elites consistently dominates state-of-the-art QD methods on the considered metrics, demonstrating capabilities to generate a large set of diverse and high-performing solutions on grasping – despite this task challenges: sparse fitness, sparse interaction, and behavior misalignment;
- We investigate the impact of the behavioral sparsity on QD methods performances, obtaining empirical evidence that sparse interaction can lead to deceptive novelty.

The code is available on Github[1]. We believe these results will open the way to apply QD on more complex tasks related to robotic manipulation, eventually solving problems that cannot easily be tackled with learning methods from other fields. The experimental results demonstrated here show that **QD methods can efficiently be leveraged to generate grasping trajectories of different fitnesses**. Such data could be used to bootstrap learning strategies of any kind. Generating grasping demonstrations is a **key matter to solve this task** (Wang et al. (2021), De Coninck et al. (2020)). To our knowledge, **no method in the literature is able to easily produce examples of grasping trajectories on different robots and objects as shown in this work**.

2 Related works

2.1 Quality diversity

While standard optimization approaches search for the extremum solution to a single-objective solution, Quality-Diversity (QD) algorithms aim to generate a set of diverse and high-performing solutions. Those methods lead to application in many fields, including image generation (Fontaine and Nikolaidis (2023)), discovery of drugs (Verhellen and Van den Abeele (2020)), or engineering optimization (Gaier et al. (2018)). QD methods rely on a behavioral characterization to compare the evaluated solutions for a given task, allowing to maintain diversity along with the optimization of a quality criterion.

QD methods emerged through two seminal works, NSLC and MAP-Elites. NSLC (Lehman and Stanley (2011b)) is a population-based evolutionary method that adds pressure toward the most novel and high-performing individuals through a Pareto-front selection. It has first been introduced as an extension of Novelty Search (NS) (Lehman and Stanley (2011a)) – an approach that replaces the quality-guided optimization process with a novelty-guided one. MAP-Elites (ME) (Mouret and Clune (2015)) is the second seminal QD method. It relies on a structured container that keeps the best previously generated solutions for different behavioral niches. Almost all QD algorithms derivate from those two pioneer methods. However, MAP-Elites-based algorithms seem to be the most popular ones: most of the current state-of-the-art methods for rapid illumination of a behavior space are more or less complex variants of ME (Fontaine and Nikolaidis (2023), Macé et al. (2023), Faldor et al. (2023)).

[1] https://github.com/Johann-Huber/qd_grasp

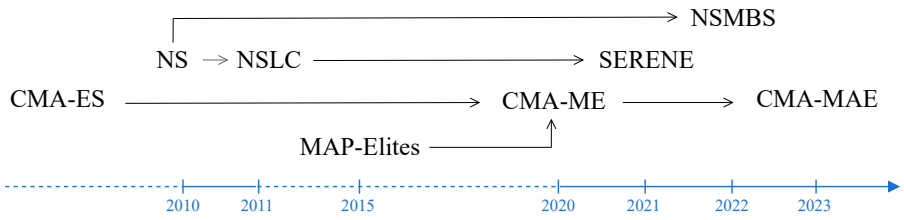

Figure 1: **Timeline of compared methods.** The literature is split into NS-derivated methods and MAP-Elites-derivated methods. In this work, we compare the 3 most promising methods for rapid illumination of a behavior space in sparse reward and behavior (SERENE, NSMBS, and CMA-MAE) with 12 other methods mainly derived from NS and ME. To fairly compare them, we proposed an evaluation framework in which the collected metrics are not dependent on the algorithms' components.

Most of the QD works in robotics are actually applied to locomotion (Lehman and Stanley (2011b), Macé et al. (2023), Faldor et al. (2023), Zardini et al. (2021)). In these domains, it is easy to design a behavioral characterization that expresses the agent's displacement, such that the exploration of this behavior space will eventually lead to the optimal solution (Lehman and Stanley (2011a), Paolo et al. (2021)). The present work aims to show that QD can efficiently be applied to more complex tasks like grasping, despite the involved challenges: sparse fitness, sparse interaction, and misaligned behavior space (see section 3.3.1).

The present work calls *complete* a behavioral characterization that allows an easy exploration by guaranteeing that a successful solution will eventually be found through the use of a non-constant behavioral signal (see section 3.3.1.4). This paper shows that **QD methods can efficiently be applied on more complex tasks like robot manipulation, in which the algorithm cannot rely on a complete behavioral characterization.**

2.2 QD for hard exploration problems

2.2.1 NS and ME divergence

Novelty Search has been introduced as a promising approach to address sparse or deceptive reward problems (Lehman and Stanley (2011a). This seminal paper led to the emergence of Quality Diversity, as NSLC, the first QD method, derivates from NS. Interestingly, the QD field has led to two research branches: **NS-based** and **ME-based works** (Figure 1). NS and ME-based methods are rarely compared in works involving one or the other family (Fontaine and Nikolaidis (2023), Kim et al. (2021)). All those methods involve similar properties and mechanisms; we argue that **comparing them might lead QD practitioners to insightful results**.

2.2.2 QD for sparse reward domains

Recent works have studied how QD methods could address domains submitted to sparse rewards: SERENE (Paolo et al. (2021)) proposes a new approach to optimize a real-valued fitness function in the sparse context; NSMBS (Morel et al. (2022)) explores multiple behavior spaces to generate grasping trajectories; CMA-MAE (Fontaine and Nikolaidis (2023)) claims to fix the limitations of CMA-ME (Fontaine et al. (2020)) on flat fitness landscapes, and reported state-of-the-art results on challenging tasks. Those methods can be considered as the most promising algorithms for addressing the task of grasping, which is submitted to sparse reward: SERENE is the only QD method that explicitly does Rapid Illumination of a Behavior Space (RIBS) in

sparse reward, NSMBS is the only QD method that demonstrated results on grasping, and CMA-MAE is a state-of-the-art method for doing RIBS, getting specific algorithmic mechanism to be more robust to sparse reward domains. **The present works compared these 3 methods with 12 other algorithms to identify the best-performing approach for doing RIBS on grasping.**

2.3 Grasping

Grasping refers to making an agent solidarize its end effector with an object by applying forces and torques on its surface. This task is of great interest to robotics and artificial intelligence research communities, as it is considered a prerequisite for many manipulation tasks (Hodson (2018)). After the early ages of analytical-based methods (Nguyen (1988)), data-driven approaches have dominated the literature on grasping in robotics since the beginning of the 21st century (Zhang et al. (2022)). Despite the involved research efforts, **grasping is still partially solved**: the reward sparsity of grasping makes it very challenging for learning methods to generate data to bootstrap learning from. To increase the chances of success of random movements, most of the approaches constrain the operational space to top-down movements (Yang et al. (2023)) or are limited to parallel grippers (Fang et al. (2020)). As the self-supervised acquisition of data is very expensive (Levine et al. (2018)), most of the recent works on grasping rely on human-provided demonstrations (Wang et al. (2021), Mosbach and Behnke (2023)). These promising results are at the cost of human-provided demonstrations, which is time expensive. The resulting grasping policies' adaptation capabilities are thus limited by the provided examples.

The acquisition of demonstrations that can bootstrap learning is thus a key matter for solving grasping. Ideally, those demonstrations should be acquired in simulation only (to avoid the issues raised by long-term self-supervised learning on real robots (Levine et al. (2018))), generated with limited human intervention, suited to many grasping scenes (robot, end effector, and objects), and diverse enough to foster generalization capabilities of the learned policies. In this work, **we leverage QD algorithms to generate large sets of diverse grasping trajectories**. The presented results show that a QD method can successfully generate grasping datasets for **different end effectors and objects**. Plus, the optimized fitness functions **associate to each generated grasp a quality label** that can be straightforwardly used for training.

3 Problem

3.1 Notations

This section introduces the notations used throughout this paper. The following subsections describe the QD notations background (section 3.1.1), the notion of fitness sparsity (section 3.1.2), behavioral sparsity (section 3.1.3), and the notations related to policy learning applied to robotics (section 3.1.4). Figure 2 overviews the notations.

3.1.1 Background

This work relies on QD standard notations (Cully et al. (2022)). Let $\Theta \subseteq \mathbb{R}^{n_\theta}$ be the *parameters space*, $\theta \in \Theta$ an *individual* (also referred as a *genome* or a *solution*). Let $\mathcal{B} \subseteq \mathbb{R}^{n_b}$ be the *behavior space*, We note $\phi_{\mathcal{B}} : \Theta \to \mathcal{B}$ the *behavior function*, such that $b_\theta = \phi_{\mathcal{B}}(\theta)$ is the *behavior descriptor* of θ. Let $f : \Theta \to \mathbb{R}$ be the *fitness function*, $d_{\mathcal{B}} : \mathcal{B}^2 \to \mathbb{R}$ a distance function within $\mathcal{B}$. We aim to generate an *archive* A defined as:

$$\begin{cases} \forall b \in \mathcal{B}_{reach},\ \exists \theta \in A,\ d_{\mathcal{B}}(\phi_{\mathcal{B}}(\theta), b) < \epsilon \\ \quad \forall \theta' \in A,\ \theta' = \text{argmax}_{\theta \in N(b_{\theta'})} f(\theta) \end{cases} \tag{1}$$

where $\mathcal{B}_{reach} \subseteq \mathcal{B}$ is the *space of reachable behaviors*, $\epsilon \in \mathbb{R}^{+*}$ is a small value that defines the density of $\mathcal{B}_{reach}$ paving, and $N(b_{\theta'}) = \{\theta \mid neighbor_{d_{\mathcal{B}}}(b_\theta, b_{\theta'})\}$ is the set of solutions for

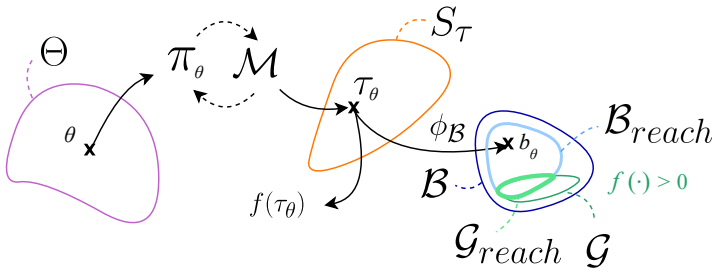

Figure 2: **Notations overview.** Each *individual* $\theta \in \Theta$ define a *policy* π_θ. This policy allows an agent to interact with an *environment* corresponding to a Markov Decision Process $\mathcal{M}$. The sequence of states and actions obtained after an *evaluation* on T steps is called the resulting *trajectory* τ_θ. This high dimensional vector is then projected in a behavior space $\mathcal{B}$ through a *behavior function* $\phi_\mathcal{B}$, obtaining a *behavior descriptor* b_θ in low dimensions. τ_θ also allow the computation of the fitness $f(\tau_\theta)$. The QD algorithm conducts its selection-mutation process using the resulting $f(\tau_\theta)$ and b_θ. The goal space $\mathcal{G}$ is the subset of $\mathcal{B}$ in which the corresponding trajectories have led to a non-null fitness.

which the projection in $\mathcal{B}$ are close to each others. $\phi_\mathcal{B}$ is supposed deterministic. The function $neighbor_{d_\mathcal{B}}$ usually corresponds to a k-nearest neighbors algorithm (Lehman and Stanley (2011a), Lehman and Stanley (2011b)) that relies on $d_\mathcal{B}$.

As the QD objective is usually to illuminate a behavioral space (i.e. fill A with high-performing solutions), we can see those problems as looking for a way to explore $\mathcal{B}$ while sampling parameters from Θ, considering that $\phi_\mathcal{B}$ is unknown and usually highly non-linear (Doncieux et al. (2019)). The exploration of $\mathcal{B}$ – and the concurrent optimization of f – is carried through the evaluations of a *domain* (or *environment*). This step is described in subsection 3.1.4 through the prism of policy learning in robotics.

A large variety of QD methods exists in the literature (section 2.1), for which the archive A can be of different nature. A can be structured (Mouret and Clune (2015)), unstructured (Lehman and Stanley (2011b)), composed of several sub-spaces (Morel et al. (2022)), or with depth (Flageat and Cully (2020)). There can also be variants without any archive (Salehi et al. (2021)). As we want to consider several kinds of QD algorithms, we here distinguish the running archive A from the *outcome archive* A_o. A is used by the algorithm during the evolutionary proccess, while A_o is an external archive used to analyse the outcome of the algorithm. Section 4.3.1 describes how A and A_o are distinguished in practice.

3.1.2 Fitness sparsity

This work focuses on the sparse fitness context. In grasping, most of the evaluated θ_i result in $f(\theta_i) = 0$. Let $f_c : \Theta \to \{0, 1\}$ be the sparse *success criterion*, such that $f_c(\theta) = \mathbb{1}_{f(\theta)>0}$. In this context, the actual output of QD is a *success archive* A_s defined as $A_s = \{\theta \in A_o \mid f_c(\theta) = 1\}$. Thus, the space to explore is the *goal space* $\mathcal{G}$ defined as $\mathcal{G} = \{b_\theta \in \mathcal{B} \mid f(\theta) > 0\}$, and more accurately the *space of reachable goals* $\mathcal{G}_{reach} = \{b_\theta \in \mathcal{B}_{reach} \mid f(\theta) > 0\}$.

3.1.3 Behavior sparsity and eligibility

This paper argues that applying QD to grasping requires facing a sparse interaction problem: $\exists \theta \in \Theta$ such that $\phi_\mathcal{B}(\theta)$ is not defined. We say here that θ results in a *non-eligible*

behavior descriptor[2]. The set of *eligible* behavior descriptors $\mathcal{B}_{elig}$ are defined as $\mathcal{B}_{elig} = \{b \mid \exists \theta \in \Theta, b = \phi_{\mathcal{B}}(\theta)\}$. In this study, $\mathcal{B}_{elig} = \mathcal{B}_{reach}$ (see section 3.3.2).

3.1.4 Policy learning in Robotics

We evaluate a policy π_θ on the Markov Decision Process $\mathcal{M}$, corresponding to the task's environment. We then obtain a trajectory $\tau_\theta \in S_\tau$, defined as a sequence of states and actions for each evaluation step along the *episode*. S_τ is usually in high dimension. We then project τ_θ from S_τ to a space $\mathcal{B}$ s.t. $dim(\mathcal{B}) << dim(S_\tau)$, and uses N to compare resulting behaviors. We here consider a fixed initial state s_0 and a deterministic $\mathcal{M}$.

In practice, the behavior function $\phi_{\mathcal{B}}$ is the result of the interaction of π_θ with $\mathcal{M}$, and the projection in $\mathcal{B}$ of the resulting trajectory τ_θ. To better match the conditions of present work's experiments, we will consider that the evaluation of a policy π_θ lead to a point τ_θ in the trajectory space S_τ, and the above-defined fitness functions project elements of S_τ to their respective space: $\phi_{\mathcal{B}} : S_\tau \to \mathcal{B}$ and $f : S_\tau \to \mathbb{R}$.

3.2 Taxonomy of QD methods

Quality Diversity (QD) and Novelty Search (NS) are concepts that describe specific methods and imply the emergence of specific addressable problems. From an algorithmic point of view, there is a clear difference between those two families of methods: NS does not rely on a fitness function – while QD does.

Nevertheless, those notions might easily be mixed up. Many works include vanilla NS among tested methods that derivate from NSLC and MAP-Elites (Pugh et al. (2016), Paolo et al. (2021)), while some others introduce as NS-related some methods that match the above-mentioned definition of QD (Kim et al. (2021), Morel et al. (2022)). The commonly shared definition of QD – an algorithm that maintains diversity and optimizes a local quality criterion – is itself not restrictive enough to prevent NS from being considered as a specific instance of QD algorithm[3].

We believe that some key insights can result from an exhaustive comparison of QD methods of different nature. Such an approach requires to avoid the mentioned ambiguities. To make the analysis easier, **we propose a taxonomy that allows us to focus on each of the similarities and differences between those methods**. Actually, the ambiguities come from the usage of *NS-based* and *QD-based* methods for different purposes. Here is a list of the main reasons for the ambiguous usage of those notions and the alternative we propose to clarify them:

- **Root algorithm.** The most well-spread misuse of language on that matter is to distinguish algorithms derivated from MAP-Elites (Mouret and Clune (2015)) and those derivated from Novelty Search (Lehman and Stanley (2011a)) by respectively calling them NS-based methods and QD-based methods. The implicit idea behind this shortcut is that those derivated algorithms share common properties of the root algorithm. The problem is that we cannot state if the shared property is the nature of the archive, the way the population is generated, or even the overall goal. In practice, many algorithms share common properties with NS (Lehman and Stanley (2011b), Kim et al. (2021), Morel et al. (2022)) or with MAP-Elites (Bruneton et al. (2019), Nilsson and Cully (2021), Macé et al. (2023)). We will thus refer to those families of methods as *NS-derivated* and *ME-derivated*, using any of the below-proposed distinctions as needed to avoid confusion.

[2]In practice, many QD algorithms require each individual to have a defined behavior descriptor. To avoid this issue, we set non-eligible descriptors to a vector of 0. For that reason, we can refer to the search space region in which the resulting behavior is eligible as the *support* of $\phi_{\mathcal{B}}$.

[3]NS novelty can be actually be considered as a dynamic quality criterion the algorithm is optimizing by maintaining a diversity of solutions throughout the evolutionary process.

- **Nature of the archive.** One might talk about NS-based methods for unstructured-archive-based ones, and QD-based methods for structured-archive based ones (Cully et al. (2022)). The structured archive is here expressing the container based on a grid (Mouret and Clune (2015)) from those in which the novelty is computed with a nearest-neighbors approach (Lehman and Stanley (2011a)). Note that the introduction of CVT-MAP-Elites (Vassiliades et al. (2017)) and the notion of minimal novelty to add an individual into an archive (Lehman and Stanley (2010)) makes those containers work similarly. To clarify this matter, we will always use the notions of *UA-based* (Unstructured-Archive) and *SA-based* (structured-archive) methods to distinguish them.

- **Population.** Considering that UA-based methods result from NSLC (Lehman and Stanley (2011b)) and SA-based from MAP-Elites (Mouret and Clune (2015)), one might use the notion of NS-based methods for population-based one – in which a living population is maintained concurrently with a container – and QD-based methods for non-population-based one – in which offspring are generated from individuals directly sampled from the container. But nothing prevents NS-derivated methods from relying on the container to sample parents. An explicit distinction should thus be made between *PP-based* methods (population-based) and *NPP-based* methods (non-population-based).

- **Run goal.** Another point of ambiguity is the purpose of the algorithm execution itself. While the ultimate goal of QD methods is clearly established (i.e. generate a set of diverse and high-performing solutions – here referred to as *Rapid Illumination of a Behavior Space (RIBS)*), NS-derivated methods aim to uniformly cover a behavior space (referred here as *Behavior Space Coverage, BSC*) (Wiegand (2020), Doncieux et al. (2019)) and by doing so, find an optimal solution to a given problem (*single objective optimization, SOO*) (Lehman and Stanley (2011a), Shorten and Nitschke (2014)).

Finally, it is worth noting that the notion of Quality-Diversity becomes the most well-spread terminology when referring to the overall field. We will thus use *QD* and *QD-based* methods to talk about the field in general and distinguish it from other research perspectives on policy learning.

This work considers state-of-the-art QD methods to generate a diverse set of high-performing grasping trajectories. To do so, **we compare methods with any of the abovementioned mechanisms**, considering that **some could have been designed to address another primary purpose** but **might lead to promising results or properties on the considered task.**

3.3 Behavioral characterization

Grasping is a challenging task (see section 2.3) in which the definition of the behavior space $\mathcal{B}$ is not trivial. This section motivates the behavioral characterization used in this work experiment. At first are introduced the concepts and notions that can help to define a behavior space $\mathcal{B}$ for a new problem to address (section 3.3.1). These concepts are then applied to grasping, in which some facilitating behavioral hypotheses cannot be verified (section 3.3.2).

3.3.1 Defining $\mathcal{B}$ for a QD problem

This subsection provides definitions of key behavioral concepts. These notions allow to define a suited $\mathcal{B}$ for a given problem but also to stress the challenges caused by its behavioral characterization. At first are defined the notions of *driving* and *describing* $\mathcal{B}$ (section 3.3.1.1), followed by the notion of *task alignment* (section 3.3.1.2), then the *density* of a behavioral function (section 3.3.1.3) and finally the notion of *behavioral completeness* (section 3.3.1.4).

3.3.1.1 Driving $\mathcal{B}$ vs Describing $\mathcal{B}$.

In section 2.2, we presented the history of NS-QD methods and how the two families of methods resulted in different paradigms. Depending on the targeted overall goal, we can distinguish two main usages of the behavior space. When doing SOO or BSC, the behavior space guides the evolutionary process toward the optimal solutions or the exploration of an outcome space. In those cases, $\mathcal{B}$ plays a *driving* role. Introduced in the seminal Novelty-Search paper (Lehman and Stanley (2011a)), this idea is still critical in recent works on NS-derivated methods (Paolo et al. (2021)). When doing RIBS, the behavior space depicts how diverse is a solution θ compared to θ' for the targeted task – through the comparison of b_θ and $b_{\theta'}$, therefore playing a *describing* role. This idea can first be found in NSLC (Lehman and Stanley (2011b)) and MAP-Elites (Mouret and Clune (2015)) papers, and is critical for all recent works on ME-derivated methods (Fontaine and Nikolaidis (2023), Anne and Mouret (2023)). In brief, **a driving $\mathcal{B}$ helps to discover solutions, while a describing $\mathcal{B}$ allows to distinguish them.** In any case, $\mathcal{B}$ is always driving and describing, but algorithms almost always focus on one or the other usage of $\mathcal{B}$.

It is worth noting that some recent works proposed to explore multiple behavior spaces (*multiBD*) to leverage the two usages of $\mathcal{B}$: Kim et al. (2021) explores both the last position of the ball thrown by the robot (driving $\mathcal{B}$) and the orientation of the end effector at the middle of the episode to generate diverse ways to throw the ball to a given position (describing $\mathcal{B}$). Similarly, NSMBS (Morel et al. (2022)) explores several behavior spaces to generate diverse grasps (describing $\mathcal{B}$) while also exploring the position of the object at the end of the episode to force the generation of successful grasp (driving $\mathcal{B}$). The multiBD paradigm raises many questions: How to explore several behavioral spaces efficiently? How to design or learn the most relevant driving or describing $\mathcal{B}$? More importantly, how to compare QD methods on the obtained results, and how to interpret the output of the algorithm?

This work studies sparse reward and interaction through the application of grasping in robotics. As the ultimate objective is to do RIBS, **a good describing $\mathcal{B}$ is required**. As grasping is a sparse reward task, **a good driving $\mathcal{B}$ is also required**. To let the abovementioned multiBD questions for future work, **the problem must be addressed through a single $\mathcal{B}$ that is both driving and describing**.

3.3.1.2 Task alignment

In QD methods, the behavior space $\mathcal{B}$ supports the exploration – either to push the solutions toward some part of the outcome space (driving $\mathcal{B}$) or to distinguish them (describing $\mathcal{B}$). Several works discussed the importance of **having a good driving $\mathcal{B}$ for making those methods successful**. In particular, Pugh et al. (2015) shows that the success of QD methods requires that $\mathcal{B}$ must be "aligned with the notion of quality".

QD methods succeed on hard exploration problems because the exploration of $\mathcal{B}$ guarantees to eventually find a successful solution. This matter has been discussed through the hypothesis of uniform sampling of $\mathcal{B}_{reach}$ by Doncieux et al. (2019). We propose to merge the idea of alignment with the hypothesis of uniform sampling through the following definition:

Definition 1 *A behavior space $\mathcal{B}$ is aligned with a task submitted to a fitness function f, a success threshold f_s, and a goal space $\mathcal{G} = \{b_\theta \in \mathcal{B} \mid f(\tau_\theta) > f_s\}$ if the probability p_{θ^s} to find a solution θ^s such that $b_{\theta^s} \in \mathcal{G}$ verifies $\lim_{n_e \to \infty} p_{\theta^s} = 1$, where n_e is the number of domain evaluations.*

Let us illustrate this idea with the experimental example given by Doncieux et al. (2019): the two wheels navigation robot that has to reach a specific point of a given maze at the end of the episode. Let us assume the hypothesis of NS uniform coverage proposed in the paper.

By taking $\phi_{\mathcal{B}}(\tau) = (x_T, y_T)$, we guarantee to eventually find a solution that is close enough to the goal point $g \in \mathcal{B}$ to verify the success criterion. However, let $\phi_{\mathcal{B}}(\tau) = \alpha_T$, α_i being the agent's orientation at time step i. In that case, we cannot guarantee that the agent will ever find a successful solution: we can generate diversity by rotating the robot at its initial position without moving from it.

Now, there are two major limitations to the above reasoning. Firstly, this hypothesis assumes that $\mathcal{G} \subset \mathcal{B}_{reach}$, which is highly dependent on both the controller and the domain; Secondly, Doncieux et al. hypothesis is about pure NS: no later study has extended this work to the overall QD paradigm, where quality optimization is usually orthogonal to pressure toward novelty. We here are interested in illuminating a behavior space. **Now that we consider sparse reward tasks, we want to make sure that the chosen $\mathcal{B}$ can be illuminated in practice.**

The purpose of the present work is not to dig into those theoretical problems. What matters here is to stress the importance of the choice of $\mathcal{B}$: **by mostly working on navigation tasks, the QD literature in evolutionary robotics assumes that $\mathcal{G} \subseteq \mathcal{B}_{reach}$ and make sure it is true in practice.** Note also that the Euclidean distance is a distance function that provides meaningful information on the exploration process: if the task is to reach a specific point at the end of the episode, the Euclidean distance on the last position allows accurate comparison of rollouts with respect to the considered task.

3.3.1.3 Behavioral density

Similarly to fitness, some behavior functions do not always provide information the working algorithm can exploit. To address grasping, Morel et al. (2022) proposed several behavioral characterization – including the orientation of the robot's end effector when touching the object for the first time; if the object is not touched, the behavior descriptor is not defined. A behavioral characterization that always provides exploitable information can be described as follow:

Definition 2 *A behavioral function $\phi_{\mathcal{B}}$ is dense if its support is equal to its domain (here if* $\operatorname{supp} \phi_{\mathcal{B}} = \Theta$*).*

Note that the navigation tasks usually addressed in QD all involve dense $\phi_{\mathcal{B}}$ (Lehman and Stanley (2011b), Macé et al. (2023), Faldor et al. (2023), Zardini et al. (2021)). On the contrary, a behavioral function that does not always provide exploitable information can be defined as follow:

Definition 3 *A behavioral function that is not dense is called sparse.*

In robotics, this setup can be referred to as a *sparse interaction* problem, as sparse behavioral function corresponds to trajectories in which the robot did not interact with the entities of interest. Of course, **all sparse behavioral functions do not result in similar task difficulty**. To estimate the sparsity associated with $\phi_{\mathcal{B}}$, a ratio of evaluations that result in a behavioral signal can be computed (see *outcome ratio* in section 4.3.2).

3.3.1.4 Behavioral completeness

Definition 4 *A behavioral characterization is complete if the behavior space $\mathcal{B}$ is aligned with the targeted task and the behavioral function $\phi_{\mathcal{B}}$ is dense.*

All QD works that address sparse reward problems involve a complete behavioral characterization (Lehman and Stanley (2011a), Paolo et al. (2021), Fontaine and Nikolaidis (2023)). In other words, the defined $\mathcal{B}$ ensures that **the algorithm can rely on an exploitable behavioral signal** throughout the evolutionary process, and that the exploration of $\mathcal{B}$ will **eventually result into the discovery of the solutions of interests** (e.g. the optimal solution, or all the solutions that validates a success criterion).

The present work argues that **a complete behavioral characterization cannot be trivially defined for many interesting tasks**. The next section elaborates on **why grasping is one of them**.

3.3.2 Defining $\mathcal{B}$ for grasping:

The definition for $\mathcal{B}$ is crucial for making QD algorithms work well (Pugh et al. (2015)). The ideal $\mathcal{B}$ would be: 1) in **low dimension** (Cully et al. (2022)), 2) **unique** (to avoid the complexity of multiBD), 3) **aligned with the task** (to efficiently drive the exploration), 4) **meaningful from the task perspective** (to generate a diversity of solution we are interested in), 5) and **easily interpretable**. This section elaborates on why misalignment cannot be avoided in grasping, and propose a behavior space that satisfies all the other mentioned criteria.

3.3.2.1 Why misalignment cannot be avoided

To address hard exploration problems, QD works on NS-derivated methods that rely on a behavior space $\mathcal{B}$ aligned with the targeted task. In these works, $\mathcal{B}$ is defined such that it can be inferred from the expression of the success criteria f_c. In Paolo et al. (2021), all the success criteria depend on the last position of the object of interest (e.g. the end effector, a ball). $\mathcal{B}$ is thus aligned with the task: the exhaustive exploration of $\mathcal{B}$ defined as the agent's last position will eventually result in the discovery of the best-performing solutions.

A similar approach might be applied for grasping: infer from f the trajectory's components that must be explored in order to find the best solutions. This work considers a grasping trajectory to be successful if a validation condition $\xi : \mathbb{R}^6 \times \mathbb{R}^6 \rightarrow \{0, 1\}$ is verified for N_g steps:

$$\left(\sum_{i=1}^{N_g} \mathbb{1}_{\xi(X_{obj}^{T-N_g+i}, X_a^{T-N_g+i})} \right) = N_g$$

where X_{obj}^i and X_{ee}^i are respectively the state of the object and the agent's end effector at step i. Both states are expressed as a 6 degrees-of-freedom pose, that is, the concatenation of a Cartesian position and its orientation in Euler angles. The grasping validation condition ξ is verified if the end effector and the object are in contacts[4], and if $z_{obj}^i > z_{obj}^0$, z_{obj}^i being the z component of the object's Cartesian position at step i. Inferring $\mathcal{B}$ from the above success criterion lead to:

$$dim(\mathcal{B}) = N_g(dim(X_a) + dim(X_{obj}))$$

By setting $N_g = 10$ steps, as used in this work experiments:

$$dim(\mathcal{B}) = 10 \times (6 + 6) = 120$$

The obtained value is way larger than low dimensional behavior spaces usually considered in QD, where $dim(\mathcal{B}) < 10$. QD methods are designed to operate on low dimensional spaces (Cully et al. (2022)). Considering such a high dimensional $dim(\mathcal{B})$ raises many questions: Can QD methods correctly explore such a large space? Should the obtained behavioral vector be encoded in a lower dimension – falling into the problems of representation-learning for QD (Cully (2019), Paolo et al. (2020))? Note that this $\mathcal{B}$ is designed to drive the exploration. Should a describing $\mathcal{B}$ also be defined – resulting in a multiBD problem?

[4] Assuming the 3D models of the object and the robot are known, the contacts can be detected with the 6 degrees-of-freedom poses only.

3.3.2.2 Proposed choice

The present work aims to study QD for grasping in a straightforward and interpretable manner. Therefore, **we have decided to set $\boldsymbol{\phi_{\mathcal{B}}(\tau) = X_a^{touch}}$, where $\boldsymbol{X_a^{touch}}$ is the position of the agent's end effector when touching the object for the first time in the episode**. This choice gives to $\mathcal{B}$ many of the expected properties: 1) it **keeps $\mathcal{B}$ in low dimensions** ($dim(\mathcal{B}) = 3$, which is the same order of magnitude as standard QD problems); 2) it consists of a **single behavior space**; 3) it is a **good driving behavior**, as touching the object is a prerequisite for grasping – discarding from a behavioral perspective any trajectory that does not interact with it; 4) it is a **well-describing descriptor**, as the goal is to distinguish grasps from the position in which the end effector interacts with the object – allowing us to control the granularity of the generated diversity at a physically meaningful scale (e.g. $1cm^3$); and 5) it makes the **outcome easily interpretable and visualizable**.

3.3.2.3 Consequences

A major drawback of the proposed behavior space is that for any given individual θ, the resulting descriptor b_θ is not defined if the object is not touched throughout a trajectory τ_θ. But what makes the QD methods efficient on hard exploration problems is replacing a fitness signal with a behavioral one, in cases where the fitness landscape is deceptive or flat (Lehman and Stanley (2011a)). In cases where $\mathcal{B}$ is perfectly aligned with the task, the optimization process can be guided with novelty to explore a behavioral landscape that always provides exploitable information. **Taking $\boldsymbol{\phi_{\mathcal{B}}(\tau) = X_a^{touch}}$ makes the problem fall into a new kind of QD problem where the domain is submitted to a sparse behavior function (or sparse *interaction*)**. Another drawback is that the chosen **$\mathcal{B}$ is misaligned with grasping**, as exhaustively exploring the space of first touching points on the object does not guarantee to find a successful grasp.

Figure 3 gives an overview of the challenges to tackle when addressing grasping with QD. While standard benchmarks imply leveraging a complete behavioral characterization for generating diversity of solutions, or for solving sparse or deceptive reward tasks, **grasping involves sparse $\boldsymbol{\phi_{\mathcal{B}}}$ and misaligned $\boldsymbol{\mathcal{B}}$**. Several questions arise: **can successful solutions be found with a misaligned $\boldsymbol{\mathcal{B}}$?** And **how to do rapid illumination of a behavior space if $\boldsymbol{\phi_{\mathcal{B}}}$ is sparse?** The next section describes the experimental protocol proposed to get empirical answers to those questions by studying QD methods performances on the task of grasping.

4 Experiments

This section describes the conducted experiments for evaluating how QD methods can address grasping. At first are described the grasping environments (section 4.1), then the compared methods (section 4.2), and finally the evaluation process (section 4.3).

4.1 Environments

The evaluated domains are grasping simulated scenes that rely on *pybullet* (Coumans and Bai (2016)) (see Figure 4). All scenes share a similar grasping scenario: a robotic arm is positioned close to a table with an object to grasp. Our study involves two *kuka iiwa*, a 7-DoF robotic manipulator, with different end effectors: the first one is a parallel 2-fingers gripper (1-DoF); the other one is an Allegro hand, a 4-fingers dexterous robotics hand with 16-DoF (4-DoF per finger). Experiments have been made on 5 objects from the YCB-dataset (Calli et al. (2015)): *chips can*, *power drill*, *mug*, *bowl* and *cracker box*. The first environment is initialized so the parallel gripper is far from the object (*kuka_wsg50_far*). The dexterous hand is initialized closer to the object in the second environment (*kuka_allegro_close*).

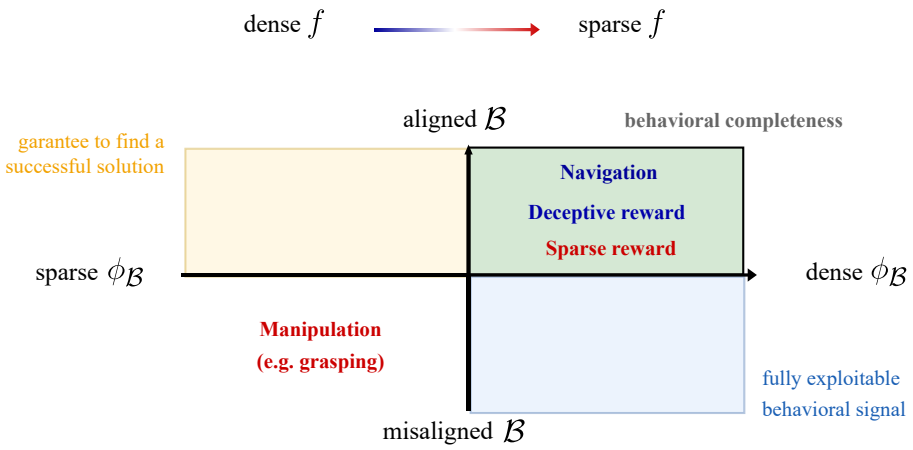

Figure 3: **Challenges to address when using QD for grasping.** In standard QD benchmarks for evolutionary robotics, the behavioral characterization is complete: $\phi_{\mathcal{B}}$ is dense, and $\mathcal{B}$ is aligned with the task. This is true for rapid illumination of $\mathcal{B}$ in locomotion tasks (Cully et al. (2022)), deceptive reward problems (Lehman and Stanley (2011a)), or sparse reward QD (Paolo et al. (2021)). By studying grasping, the present work argues that addressing QD with manipulation tasks can lead to sparse $\phi_{\mathcal{B}}$ and misaligned $\mathcal{B}$.

The grasping controller consists of an open-loop trajectory guided by 3 waypoints. The gripper position is initialized to open position. When the end effector first touches the object, the gripper is closed with constant force. This mechanism is inspired by the *Palmar Grasp Reflex*, which makes newborn infants close their hands when pressure and touch are applied to the palm (Futagi et al. (2012)). **The controller is described in detail in the section A of the supplementary materials.** Finally, the fitness function consists of a normalized mixture of two sub-fitnesses that aim to minimize energy consumption and to minimize the variance of contact points between the end effector and the objects. Note that its value is set to 0 if the object is not grasped. **The fitness function computation is detailed in section B of the supplementary materials.**

4.2 Methods

Table 1 provides an overview of each of the studied methods with respect to the taxonomy proposed in section 3.2. This table shows why NS and ME-derivated methods are usually called NS and QD-based methods, as most of those algorithms share common properties. However, we believe this matter is a chicken-and-egg problem: the lack of accurate distinction results in implicit design choices that do not question previously established algorithmic paradigms.

This section provides details on the compared methods. All methods rely on the same behavioral characterization ($\phi_{\mathcal{B}}(\tau) = X_a^{touch}$). When not explicitly stated, the mutation is a Gaussian perturbation. We decided not to use the crossover to avoid adding too many complexities and hyperparameters that might affect the results. Some works in the field suggest that this could improve the performances (Vassiliades and Mouret (2018)); crossover is thus considered for future work. For comparison fairness, all NPP methods that can sample a variable number of individuals from the container are implemented such that this number matches the population size of PP methods.

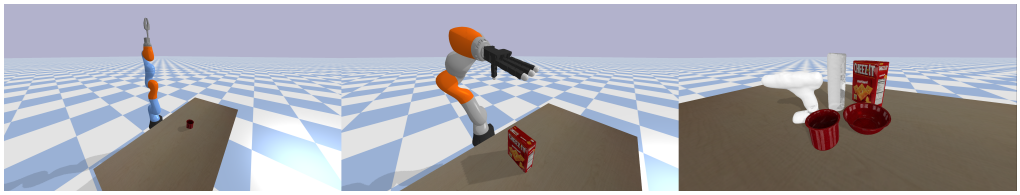

Figure 4: **Studied domains.** (Left) The *kuka_wsg50_far* environment consists of a parallel 1DoF gripper mounted on a kuka iiwa robotic manipulator. At the initial position, the end effector is far from the object. (Center) The *kuka_allegro_close* robot environment consists of an Allegro 4-fingers dexterous hand mounted on a kuka iiwa. This end effector is initialized close to the object. (Right) The 3D models of the YCB objects (Calli et al. (2015)) to grasp: power drill, chips can, cracker box, mug, and bowl.

Random. Simple PP baseline that initializes its population randomly, evaluates it, sample offspring, evaluate them, and then randomly generate a new population for the next generation.

NS. Similar to standard NS method (Lehman and Stanley (2011a)).

Fit. Another common PP baseline that selects its population based on individuals' fitnesses (descending order).

ME-rand. Standard MAP-Elites algorithm (Mouret and Clune (2015)), that randomly samples individuals from the archive of elites.

ME-scs. ME-rand does not seem suited to the targeted task, as the hard exploration nature of the problem will make the probability of sampling the best-performing individuals very low. To explore the potential of ME on grasping, a simple variant that selects in priority individuals θ_i such that $f(\theta_i) > 0$ would be more suited to the targeted task. In practice, it randomly samples individuals from the successful ones in the container, and if there are not enough solutions to fill the "population" (set of individuals used to generate the offspring), it fills it with randomly sampled solutions – just like a standard ME-rand. This simple ME variant seems way more adapted to grasping, as the research will be strongly biased toward local regions around already successful solutions.

NSLC. Similar to standard NSLC (Lehman and Stanley (2011b)). Surprisingly, this seminal QD method has been left behind in favor of MAP-Elites (Cully et al. (2022)). To our knowledge, no reference paper provides theoretical or experimental results that would justify dropping it when addressing a new QD problem. There are many algorithmic differences with ME: Pareto front selection over novelty and local quality, the way local quality is computed (how many neighbors have lower fitness than the evaluated individual), and its UA and PP backbone. This goal is not to compare those two approaches comprehensively but instead consider both methods as candidates for addressing grasping with QD.

NSMBS. NSMBS is the only QD method in the literature that is explicitly designed to address grasping (Morel et al. (2022)). It consists of an NS-derivated algorithm – PP and UA – that introduces two innovations: exploring multiple behavior spaces, and a specific selection operator that sequentially selects a behavior space and then makes a tournament-based novelty-guided selection. Note that an exhaustive comparison with QD methods is yet to be done. To get a fair comparison and easily interpretable results, NSMBS has been applied on a single behavior space ($\phi_{\mathcal{B}}(\tau) = X_a^{touch}$). NSMBS is thus similar to NS with a tournament-based selection. A comprehensive study on multiBD for grasping is considered for future work.

SERENE. NS-derivated method specifically designed to address sparse reward problems

methods	**root**			**container**			**pop**		**goal**		
	NS	ME	ø	UA	SA	ø	PP	NPP	RIBS	cvg($\mathcal{B}$)	θ^*
Random			•			•	•				
NS	•			•			•			•	•
Fit			•	•			•				•
NSLC	•			•			•		•		
NSMBS	•			•			•			•	
SERENE	•			•			•		•		
ME-rand		•			•			•	•		
ME-scs		•			•			•	•		
CMA-ES			•			•	•				•
CMA-ME		•			•			•	•		
CMA-MAE		•						•	•		

Table 1: **Taxonomy of the compared methods.** Each column corresponds to a family of QD method (see section 3.2): the root algorithm (Novelty Search (NS), MAP-Elites (ME), or none of those two (ø)), the nature of the container (unstructured archive (UA), structured archive (SA), or no container (ø)), the population mechanism (population-based (PP) or directly sampled from the archive (NPP)), and the overall objective of the algorithm (rapid illumination of a behavior space (RIBS), dense coverage of a behavior space (cvg($\mathcal{B}$)), or finding the optimal solution (θ^*).

(Paolo et al. (2021)). The method can be decomposed into two phases: an exploration phase consisting of a standard NS algorithm; and an exploitation phase, exploiting solutions found with non-null reward to initialize CMA-ES emitters to refine the solutions. To our knowledge, this is the only work specifically focusing on sparse reward context for QD.

CMA-ES. Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is one of the best derivate-free optimization algorithms for continuous domains (Hansen (2016)). It models the sampling distribution of the population as a multivariate normal distribution, estimated from the previous generation's best-performing solutions. Even though this method is single objective-oriented, its impressive results on many problems made us use it as a baseline to estimate how good other methods are to explore, or to generate high-performing solutions. Note that CMA-ES is not a QD algorithm.

CMA-ME. Covariance Matrix Adaptation MAP-Elites (CMA-ME) combines self-adaptation techniques of CMA-ES with diversity-maintaining techniques of MAP-Elites (Fontaine et al. (2020)). Despite the great results shown in the paper, recent work shows that this method struggle in sparse reward domains (Paolo et al. (2021)). This method is kept as a baseline to verify this weakness in new domains and emphasize key properties of other algorithms.

CMA-MAE. Improved version of CMA-ME to address three of its limitations: quick abandonment of difficult-to-optimize objectives, inability to efficiently explore flat objective functions, and inefficiency on low-resolution archives. The results presented in this recently published paper (Fontaine and Nikolaidis (2023)) makes CMA-MAE a state-of-the-art QD method. The fact that it has been explicitly designed to address flat fitness landscape scenarios makes it a serious candidate to tackle grasping.

Given the good results of MAP-Elites, a study has also been conducted on the following variants:

ME-fit. Standard MAP-Elites that select solutions with respect to their fitness, sorted in descending order.

ME-nov. Standard MAP-Elites that select solutions with respect to their novelty, sorted in

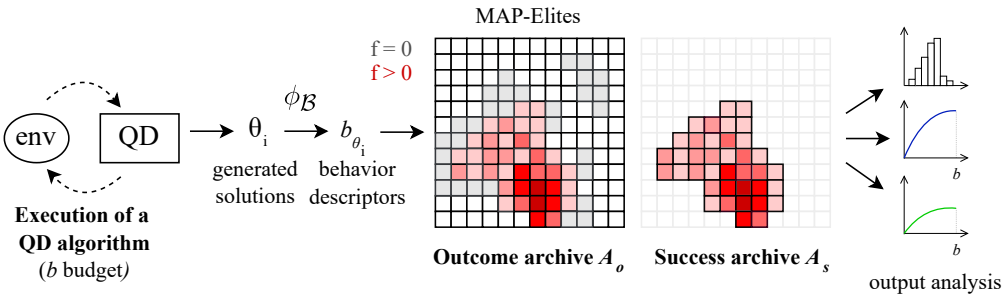

Figure 5: **Algorithm evaluation framework.** To distinguish the algorithmic components from the evaluation container, all the generated solutions are considered as candidates to be added into an archive of elites (Mouret and Clune (2015)) here called the *outcome archive*, after having projected them into the behavior space $\mathcal{B}$. Standard QD metrics can then be computed on this external archive. Grasping is a sparse reward task; what matters in this study is the set of successful solutions within A_o, called the *success archive* A_s. This last archive is the ultimate output of the evaluated methods in the present work.

descending order.

ME-nov-fit. Standard MAP-Elites that sample solutions through a pareto-front non-dominated selection, using both novelty and fitness.

ME-nov-scs. Standard MAP-Elites that select solutions with respect to their novelty, sorted in descending order – selecting successful solutions in priority (similarly to ME-scs).

As discussed in section 3.3.2, $\phi_{\mathcal{B}}(\tau) = X_a^{touch}$ is defined for any sampled trajectory. The following design choice has thus been made: if a method requires a defined behavioral descriptor for an evaluated individual, its value is set to an arbitrary fixed value ($b_\theta = (0, 0, 0)$)). Otherwise, its value is left undefined, as it will be discarded through the upcoming behavioral instructions.

All methods have been implemented from scratch, except for the following ones: the official SERENE implementation has been used (Paolo et al. (2021)), as well as pyribs (Tjanaka et al. (2023)) to get the official implementation of CMA-ME and CMA-MAE. Note that pyribs has also been leveraged for standard CMA-ES, relying on the theoretical equivalence of CMA-MAE with CMA-ES for $\alpha = 0$ (Fontaine and Nikolaidis (2023)). Details can be found in the publicly shared code.

To limit the energy consumption of this study, the experiments have been carried out on two evaluation budgets: 100k evaluations (long run) and 400k evaluations (very long run). The 100k evaluations experiments include the following methods: Random, NS, Fit, ME-rand, ME-scs, NSLC, NSMBS, SERENE, CMA-ES, CMA-ME, CMA-MAE. Methods tested on 400k evaluations depend on the best-performing methods obtained on 100k evaluations: ME-rand, ME-scs, ME-fit, ME-nov, ME-nov-fit, ME-nov-scs, CMA-ME, CMA-MAE.

4.3 Evaluation

This section describes the evaluation protocol used in the experiments. At first is presented the framework proposed to distinguish the evaluation of QD algorithms from their internal components (section 4.3.1), then the computed metrics (section 4.3.2), and finally the chosen hyperparameters (section 4.3.3).

4.3.1 Algorithm output

As most QD works focus on either NS-derivated or ME-derivated methods, the running output is usually the container used by algorithms. RIBS-related works measure the structured archive coverage and qd-score (Fontaine and Nikolaidis (2023), Macé et al. (2023)), while all works that focus on NS measure performances on the resulting unstructured archive or extracting from the evolutionary process the best-performing individual (Paolo et al. (2021), Conti et al. (2018)).

Comparing QD methods of different natures raises many issues: is the qd-score still a relevant metric when comparing SA and UA-based algorithms? How to compute $cvg(\mathcal{B})$? More generally, can the evaluation of performances be independent from the container used during the evolutionary process – discarding the constraint of operating in a fixed behavior space or even the requirement of using one? Recent works in the QD community stressed the rising need for distinguishing the output of the algorithms from the algorithmic modules themselves (Fontaine and Nikolaidis (2023)).

In this work, **we introduce an evaluation procedure that distinguishes the algorithms' output from its internal components.** The evaluation framework is described in Figure 5, and basically consist of an external MAP-Elites that does not generate any new solution. The interaction between the environment and the QD algorithm is considered as a black box that generates solutions θ_i for a given budget b. All the solutions that have ever been generated during a given run are submitted to this evaluation procedure. Each obtained trajectory τ_{θ_i} is then then projected into a behavior space $\mathcal{B}$, obtaining a behavior descriptor b_{θ_i}. This descriptor is then considered for being added into an *outcome archive* A_o, similarly to a standard MAP-Elites. The *success archive* A_s is then defined as : $A_s = \{b_{\theta_i} \in A_o \mid f(\tau_{\theta_i}) > 0\}$. Note that in the case of non-sparse reward domains, $A_s = A_o$.

4.3.2 Metrics

Considering the above-proposed evaluation framework, the analysis is conducted on the following metrics:

Coverage of the Outcome Archive. As the behavior space is here $\phi_{\mathcal{B}}(\tau) = X_a^{touch}$, computing the coverage of A_o results here to answer the question: how diverse are the entry point for all the grasping attempts? The higher $cvg(A_o)$, the more first touching points have been discovered by the agent.

Coverage of the Success Archive. Computing $cvg(A_s)$ answers the following question: How many diverse successful grasps have been found? Interestingly, this metric appeared to be the most important one, as the qd-score for the chosen fitness function is aligned with this metric (see supplementary materials C).

Top-N fitnesses. To get an idea of how efficient the generated solutions are, the top-N fitnesses ever produced by each algorithm are also compared. This metric is an alternative to the qd-score to estimate the quality of the diverse generated solutions. Qd-score might be dominated by the number of found solutions, especially in difficult exploration tasks like grasping. It informs us on the number of successful grasps found, but not on their performances.

Environment difficulty. To evaluate the challenges associated with each studied task, the environment difficulty is evaluated through two metrics: the *outcome ratio* η_o and the *success ratio* η_s. To compute those metrics, N_{ed} random individuals θ_i are generated by sampling from a uniform distribution within the genotype space Θ. The obtained θ_i are then evaluated on the environment $\mathcal{M}$, getting the resulting number of individuals that successfully touched the object (n_o) and the number that grasped it (n_s). The ratios are then computed as follows:

$$\eta_o = \frac{n_o}{N_{ed}} \qquad \eta_s = \frac{n_s}{N_{ed}}$$

Those metrics stress whether an environment is submitted to sparse behavior function

($\eta_o \rightarrow 0$) or not ($\eta_o \rightarrow 1$). Similarly, η_s expresses how sparse the problem is: the closer to 0 the sparser, the closer to 1 the denser.

4.3.3 Hyperparameters

QD methods. Notations for hyperparameters are the following: μ is the population size, λ the number of offspring, n_A the number of individuals added to the archive at each generation, k the number of neighbors considered for novelty computation, N_{rt} the maximum number of rollouts. We set: $\mu = \lambda = 100$, $n_A = 40$, $k = 15$. All offspring are mutated with a probability $ind_{pb} = 0.3$ to modify each gene. For a fair comparison, all ME-derivated methods sample $\mu = \lambda$ individuals for offspring generation at each iteration. The mutation operator applied by default to all the methods is a Gaussian perturbation of 0 mean and 0.5 standard deviation[5]. The archive size is unbounded for UA-based methods. For NSMBS, the tournament size for selection is set to 15. For NSLC, we use 50 neighbors to estimate local quality due to the sparsity of the task. The same parameters as SERENE paper have been used (Paolo et al. (2021)): the chunk size is set to 1000, the emitter population len to 6, and the same k as other methods for estimating novelty. 5 individuals are added to the archive at each iteration. For CMA_* variants, we used the same parameters as in the papers (Fontaine et al. (2020), Fontaine and Nikolaidis (2023)): The emitter batch size is set to 36, and the number of emitters to 15. For CMA_MAE, $f_{min} = -1$ and $\alpha = 0.01$.

Grasping domains. The boundaries of the structured archive match the operational space. To get a precision of 1cm³ for contact points, the number of bins per dimension is (n_bins_x, n_bins_y, n_bins_z) = (24, 25, 25). Parameters used for normalizing energy consumption fitness for each robot are provided in the shared code on Github. The operational space is set as a box of (dx, dy, dz) = (1, 0.7, 0.5) meters on the top of the table. The object is initialized at the center. Robots are controlled in position such that their cartesian target cannot be set out of a virtual box within the operational space: (dx, dy, dz) = (0.8, 0.4, 0.4) meters for Allegro, (dx, dy, dz) = (1, 0.7, 0.5) for the 2-fingers grip. For the dexterous allegro hand, 6 grasping primitives have been defined: 1) index and thumb closure; 2) middle finger and thumb closure; 3) thumb and last finger closure; 4) thumb, index, and mid; 5) thumb; mid and last; and 6) all fingers closure. The episode length is $T = 2000$ for *kuka_wsg50_far* and $T = 1500$ for *kuka_allegro_close*.

Environment difficulty. 400k randomly sampled trajectories have been deployed on all the considered domains. The controller and parameter space are the same as those used for the QD evaluation methods.

Outcome archive. The archive sampling matches the one used on ME variants described above.

5 Results

This section presents the obtained results. Here is what can be expected regarding the literature:

- NSMBS, CMA-MAE, and SERENE are the most promising methods to do RIBS on grasping; they should dominate other methods on the evaluated metrics;

- NS should perform poorly on $cvg(A_s)$, as NSMBS paper shows that NS struggles to generate many successful solutions on grasping (Morel et al. (2022)). However, it should lead to a high value of $cvg(A_o)$, as NS is directly optimizing the exploration of $\mathcal{B}$ (Paolo et al. (2021)).

[5]This standard variation has been chosen after doing a grid search on tested methods, selecting the value that maximizes $cvg(A_s)$ on the maximum number of methods avec 20k evaluations. It is worth noting that a too-small std prevents exploration, while a too-large std constrains the waypoints at the edge of the operational space.

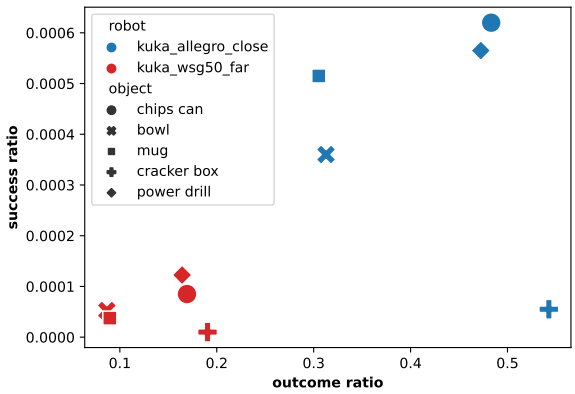

Figure 6: **How challenging are the studied tasks.** We indeed are in a sparse reward context, as a few of the sampled trajectories lead to a non-null fitness. Plus, many deployed trajectories do not provide any behavioral information to exploit, making the exploration even more difficult. The higher the outcome ratio, the higher the success (Pearson correlation $r = 0.61$ with $p < 0.06$). Overall, *kuka_allegro_close* domain is easier than *kuka_wsg50_far* – except for the Allegro on the cracker box, in which the object is easy to touch but hard to grasp.

The section 5.1 presents the result on the difficulties associated with each domain. The results obtained on 100k evaluations are provided in section 5.2. The results obtained on 400k evaluations are provided in section 5.3.

5.1 Environment difficulty

Figure 6 gives an overview of the challenges associated with each domain. The outcome ratio on *kuka_allegro_close* is significantly higher than those obtained on the *kuka_wsg50_far*: between 31% and 54% of the randomly sampled trajectories touches the object on the first domain, while between 9% and 19% on the other one. In both cases, **many trajectories do not provide a behavioral signal that the algorithm can exploit to guide the exploration toward high-performing solutions**. As expected, the smaller objects (bowl and mug) lead to lower η_o than the larger ones (power drill, cracker box, and chips can). **All domains are submitted to sparse rewards**, as the higher value of η_s is close to 0.06% of success, meaning that in a pure random search context, about 99.94% of trajectories result in a null fitness.

Overall, the domains in which the object is easier to touch are those in which the object is easier to grasp (Pearson correlation between η_o and η_s: $r = 0.61$ with $p < 0.06$). The Allegro hand on the cracker box is a remarkable outlier, though: it is easier to touch it ($\eta_o = 0.54$) than any other objects on that gripper, but it is about 10 times more difficult to grasp it than on any other tested objects. The easier domains are (*kuka_allegro_close*, chips can) and (*kuka_allegro_close*, power drill), with respectively 48% and 47% of chance to randomly touch the object, and about 0.062% and 0.057% of chance to grasp it. (*kuka_allegro_close*, bowl) can be seen as a intermediate-difficulty environment ($\eta_o = 0.31$ and $\eta_s = 0.00036$). The most challenging domains are the *kuka_wsg50_far*-based setups, especially on the cracker box on which the success ratio reaches its smallest value ($\eta_s = 0.00001$). The most challenging environment regarding the behavioral sparsity are (*kuka_wsg50_far*, mug) and (*kuka_wsg50_far*, bowl). It is also worth noting that a similar difficulty pattern can be seen on both robots for the

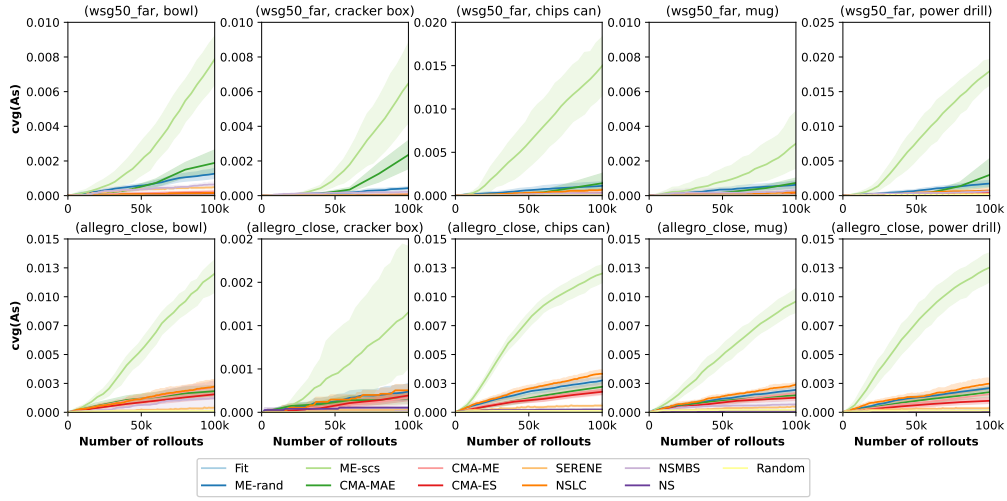

Figure 7: **Coverage of the success archive throughout the evolutionary process.** Over 10 seeds. ME-scs dominate all the compared methods by a large margin, even the state-of-the-art QD methods designed to address sparse reward domains (CMA-MAE, SERENE, NSMBS).

studied object: the chips can and the power drill are the easier objects to address; the cracker box and the mug are more challenging tasks regarding both behavioral and fitness sparsity; and the cracker box is the most difficult to grasp object – despite of its high probability to touch it. The ratios obtained on the cracker box can be explained by the fact that there is a high probability of making the object fall, resulting in an impossible-to-grasp state for those grippers – while the chips can and the power drill are less challenging as they can still be grasped after having made them fell on the table.

Those metrics provide insight into key properties of the studied environments: **grasping is indeed submitted to sparse reward**. Plus, **the chosen behavior space** (see section 3.3.2) **results in a flat behavioral landscape** too – as the object is not always touched. It actually mirrors the nature of the task and its inherent challenges, as the reach-and-grasp sequence is way more complex than adults think it is: it takes about 4 months for a baby to be able to reliably deploy this skill (Needham and Nelson (2023)).

The next section presents the result obtained after 100k evaluations on the compared methods.

5.2 Comparison of state-of-the-art QD methods

Generation of diverse successful solutions. In Figure 7 are the obtained coverage of the success archive for each method throughout the evolutionary process. The most striking result is that **ME-scs is dominating all the compared methods on all the evaluated domains by a large margin**. CMA-MAE outperforms all other methods except for ME-rand, which reaches similar coverage for several environments. SERENE is way below, and so is NSMBS. Another interesting point is that NSLC, the NS-derivated RIBS method, does not succeed as well as the ME-derivated variant. We attribute this to its selection process and its PP nature. On such a hard exploration problem, it is difficult to reach non-null fitness – such that the local quality score (Lehman and Stanley (2011b)) is likely to either be at the maximal value (i.e. the individual is

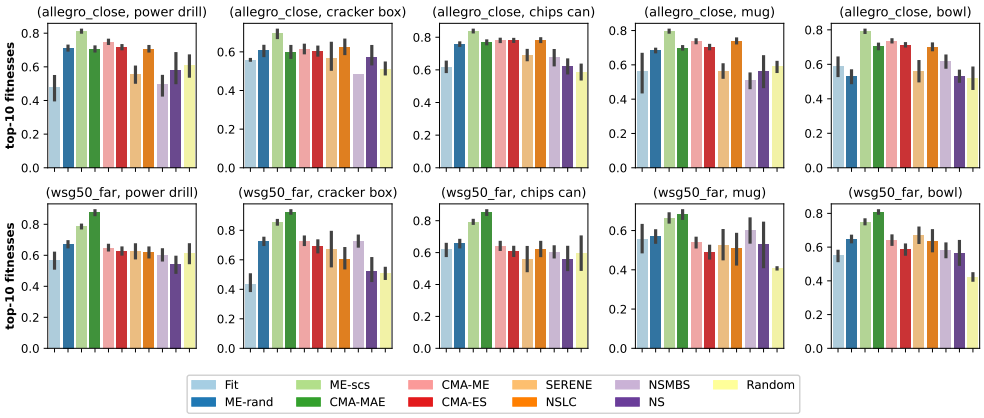

Figure 8: **Fitnesses of the top-10 best performing individuals obtained after 100k evaluations.** ME-scs and CMA-MAE outperforms other methods after this number of evaluation, obtaining high-performing solutions with comparable fitnesses.

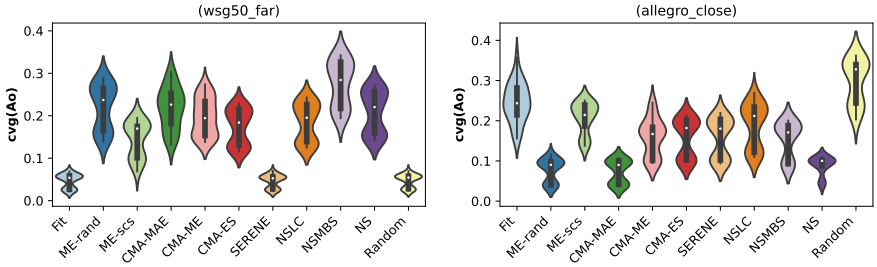

Figure 9: **How do QD methods explore the objects surface.** Coverage of the output archive after 100k evaluations, over 10 seeds. Even though ME-scs crushes other methods on the generation of successful grasps, its exploration of $\mathcal{B}$ is not significantly higher than other methods – meaning that ME-scs prioritize parents that are the more likely to mutate into successful individuals.

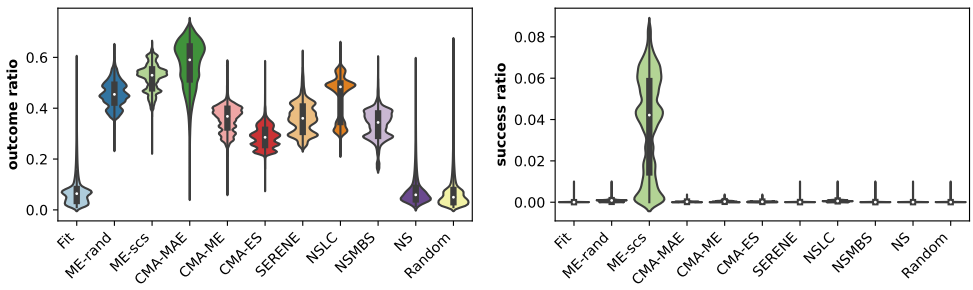

Figure 10: **Distribution of success ratios and outcome ratios for each method on *kuka_allegro_close*.** Aggregated over all objects throughout 100k evaluations over 10 seeds. By prioritizing successful solutions to generate offspring, ME-scs favors having a high probability of success. NS pushes its population close to the non-eligible regions of the search space. Quality-oriented emitters constrain the exploration in promising regions. CMA-MAE explores efficiently $\mathcal{B}$ but does not generate as many successes as ME-scs on those domains.

dominating all its neighbors) or to the lower value (i.e. the individual has a fitness lower or equal to all its neighbors). Consequently, the selection is always guided by novelty, making NSLC acts similarly to a standard NS. Note that the PP nature of both algorithms is prone to forgetting, as lower quality solutions that are not likely to generate new successes might be preferred to solutions with a higher quality and/or that are more likely to produce new successes because of the local quality issue. Lastly, NSLC optimizes the novelty and the quality of its current population. In contrast, ME directly optimizes a container with the same structure as the outcome archive to illuminate.

Generation of high-performing solutions. Figure 8 provides the distribution of top-10 fitnesses obtained with the evaluated methods. Again, **ME-scs outperforms all other methods**. Even CMA-MAE generated less performing solutions, showing that **the ME-scs ability to produce a large success archive lead to the emergence of high performing solutions**. The simple fitness-guided evolutionary algorithm (Fit) did not perform better than a random process, which is expected in hard exploration domains (Lehman and Stanley (2011a)). What is more unexpected is that NS did not perform any better than those two, while this method is meant to find high-quality solutions in hard exploration problems. We attribute this performance to the choice of $\mathcal{B}$. **As $\mathcal{B}$ is here not aligned with the task** – there exist solutions that touch the object from any entry point without resulting into a grasp – **there is no guarantee that NS will eventually find the best-performing solutions.**

Another interesting result is the poor performance of NSMBS. It is worth noting that NSMBS slightly outperforms NS in all the domains. The unique difference between those two methods here is the selection operator: the tournament selection of NSMBS by-pass the issue of non-eligible solutions – that must have a valid behavior descriptor for NS to compute each individual's novelty. The arbitrarily set value for non-eligible solutions creates a strong bias of novelty within the archive, pushing the exploration away from those values. On the contrary, NSMBS discards the non-eligible solutions from selection, as the tournament's candidate individuals are sampled from eligible individuals only (Morel et al. (2022)). It forces the exploration process toward eligible solutions without biasing the estimation of novelty within $\mathcal{B}$. Nevertheless, NSMBS performances are far from the best-performing ones. Those results dis-

card NSMBS from the most promising QD methods to address grasping, stressing what should be its main interest: the multiBD context (see discussion in section 6.1.4).

SERENE reached slightly better performance than NS and got similar results to NSMBS. It matches what has already been discussed in the original paper: SERENE struggles to refine solutions in hard exploration domains (Paolo et al. (2021)). It initially behaves just like a standard NS. As soon as non-null fitness solutions have been found, they are evaluated as candidates for budget assignment to do local optimization. The problem is that in such a hard exploration domain as grasping, a successful solution might not lead to any successful one in a few 1-mutation trials. We attribute the obtained results to this phenomenon: in most cases where NS found successes, the candidates are evaluated as non-promising solutions for local optimization, making the method fall back to a standard NS. It sometimes allows local optimization, which explains the slight performance gain compared to NS on all the domains.

Exploration of the object surface. Figure 9 shows the coverage of the output archive after 100k evaluations. Interestingly, ME-scs does not dominate other methods on that metric. It shows that **by selecting successful solutions from the archive, ME-scs focuses on the regions of the object surface which are more likely to result in successful grasps**.

Surprisingly, methods derivated from NS do not explore the object surface better than other methods. On *kuka_wsg50_far*, NS does not report a higher coverage than most quality-guided methods. Even more surprising is that **NS reports one of the lowest $cvg(A_o)$ on *kuka_allegro_close*.** Pure NS or NS variants that are not often submitted to a non-null quality signal are however optimizing $cvg(\mathcal{B})$ directly. NS-derivated methods which are also guided by quality have generated a significant number of successful solutions: SERENE, NSLC, and NSMBS dominate NS on $cvg(A_s)$ and on the top-10 fitnesses on many domains. They have therefore been confronted to non-null fitnesses, limiting their exploratory capabilities. But no straightforward answer can be brought to NS performances. Results given on the outcome ratio provide hints to explain those results.

Outcome and success ratios. In Figure 10 are displayed the outcome and success ratios measures for each algorithm throughout the evolutionary process on the *kuka_allegro_close* domains. The selection operator of ME-scs leads to a probability of generating success $\eta_s = 3.7\%$. ME-scs is above 60 times more sample efficient than random sampling in Θ on the easier domain and 50 times more sample efficient than the second higher success ratio on that domain (ME-rand). But there is no statistically significant difference between NS and Random on the outcome ratio. Figure 9 shows that NS does not behave as a Random search on this problem, the reported $cvg(A_o)$ is different for both methods. It means that **NS pressure for novelty has a deteriorating impact on $cvg(A_o)$ in this context** – leading to an exploration of $\mathcal{B}$ that is similar to quality-guided methods (*kuka_wsg50_far*) or significantly worse (*kuka_allegro_close*). Such a result goes to the opposite of the strong exploratory power of NS that has been demonstrated in literature so far (Paolo et al. (2021), Doncieux et al. (2019)). **We attribute the deteriorating impact of NS to its pressure for evolvability.** This point is discussed in section 6.2.

The resulting QD-scores are provided in supplementary materials (Figure 15). In this case, this metric is dominated by the number of found successful solutions, preventing us from relying on this measure to evaluate the ability of the tested methods to generate a set of diverse and high-performing solutions. Thus, the results only reflect the size of A_s, which is redundant with the coverage information.

The leading performances obtained with ME-scs invite us to investigate the role of the selection operator on the illumination of A_s through ME variants. This second experiment compares several ME variants on longer runs (400k evaluations). Are also included the second-best performing solution (CMA-MAE) and the anterior version of this method (CMA-ME) to get more clues on how those state-of-the-art approaches behave in those domains.

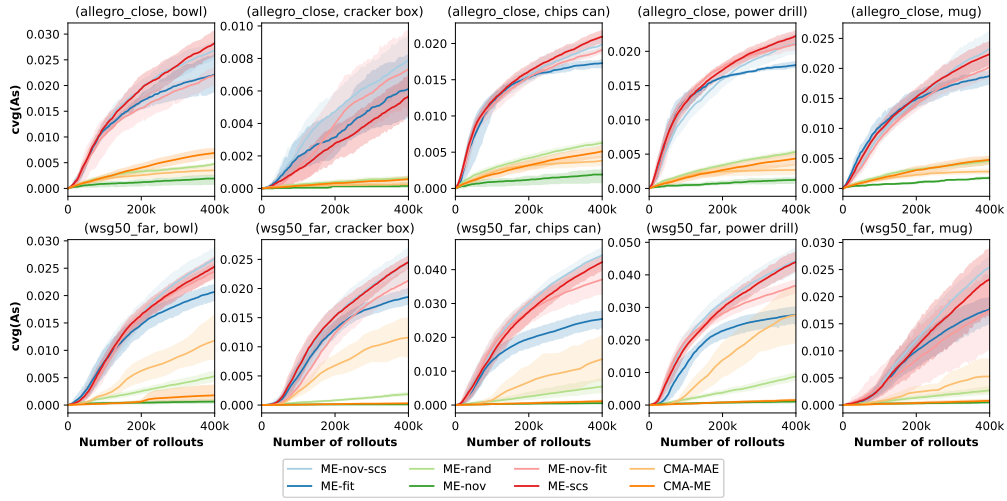

Figure 11: **Impact of the selection operator on ME's ability to generate diverse, successful solutions.** Over 10 seeds. All ME variants that select non-null fitness solution in priority outperforms other methods. Prioritizing fitness can stick the exploration into local minima while selecting successful solutions regardless of their performance increases the success archive's size continuously.

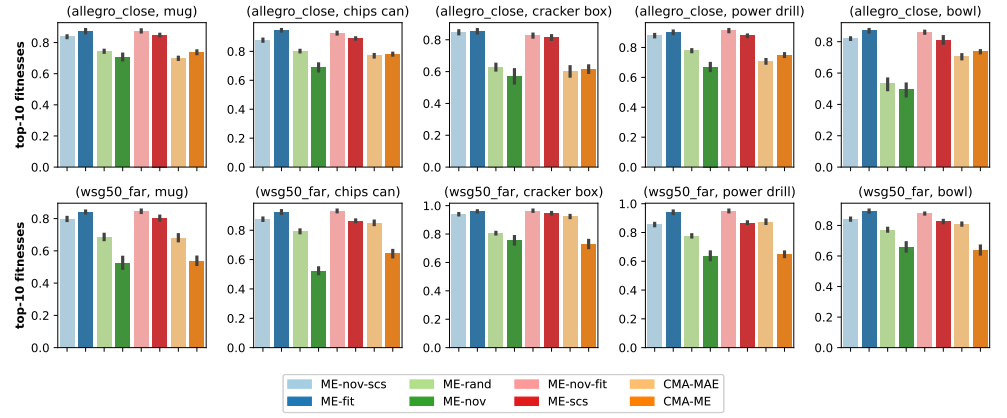

Figure 12: **Impact of the selection operator on high performing solutions fitnesses.** Over 10 seeds, after 400k evaluations. ME-scs outperforms CMA-MAE on longer runs. ME-fit-based variants generate slightly better solutions than ME-scs-based ones.

5.3 Impact of the selection operator

Generation of diverse successful solutions. Figure 11 shows the coverage of the success archive. **Variants that select in priority the successful solutions (ME-scs, ME-nov-scs) dominate all other methods.** Prioritizing the most novel solution among the successful ones does not lead to a statistically significant difference. Selecting individuals with respect to their fitness (ME-fit and ME-nov-fit) leads to better results than other variants, except for the success-based ones. By selecting the best-performing individuals among the already discovered ones, the evolutionary process can get stuck in local minima – preventing the generation of new successful solutions.

Generation of high-performing solutions. Figure 12 shows the obtained distribution of top-10 fitnesses. It can be seen that **ME variants that prioritize fitness lead to the generation of better-performing individuals than success-based selection.** The two ME-scs variants reached comparable results. There are no statistically significant differences between ME-fit variants too. This result is unsurprising: the pressure for novelty is applied to already successful solutions for those 4 variants[6].

The **poor performances of ME-nov on both $cvg(A_s)$ and top-10 fitnesses enforce the result obtained with NS**: pressure for novelty might lead to deteriorated exploration performances. Interestingly, a ME-derivated method shares similar properties with an NS-derivated one. There are differences between ME-nov and NS beyond the nature of the archive: individuals compete on fitness in their behavioral niche, while NS does not. Discussion on the impact of novelty-based selection under in flat behavior landscape is provided in section 6.2.

CMA-MAE is outperformed by all success or fitness-guided ME variants. It is worth noting that CMA-MAE consistently generates better high-performing solutions than CMA-ME on the most challenging robot (*kuka_wsg50_far*). However, the opposite can be seen on the other robot. The method that dominates this other for generating high-performing solutions is also the one that reached the higher coverage of the success archive (see Figure 11, or more explicitly in section D of supplementary materials). We attribute this observation to the behavior sparsity of the task. CMA-ME is well known for pushing the search away from previously found solutions (Fontaine and Nikolaidis (2023)). It limits CMA-ME from optimizing the performances of already-found solutions. CMA-MAE alleviates this issue with a rolling fitness threshold that conditions elites' insertion into the container. This constraint CMA-MAE's exploration capability, as it spends more budget in the same region of the search space. This mechanism allows CMA-MAE to better exploit previously found successes to find new ones on the most difficult task. On the allegro task, such a strategy might keep the evolutionary process to local minima. CMA-ME exploration capabilities help to escape those minima, while it can be detrimental in a more sparse behavioral domain.

The resulting QD-scores are provided in supplementary materials (section C). Similarly to the first set of experiments (section 5.2), this metric is dominated by the coverage of A_s.

6 Discussion

This section discusses the obtained results – with a spotlight on methods that were expected to get good performances on grasping. Section 6.1 analyses the obtained results on RIBS for grasping. Section 6.2 provides a hypothesis on the detrimental role of novelty in sparse interaction domains, and section 6.3 discusses the proposed evaluation framework.

[6] A multi-objective solution that balances the pressure on performing/successful individuals with the pressure for novelty on non-successful individuals could have been set. As the scope of this work is rapid illumination of a success archive for grasping, the results obtained with novelty-guided solutions are not prone to stimulate too much effort in that research direction.

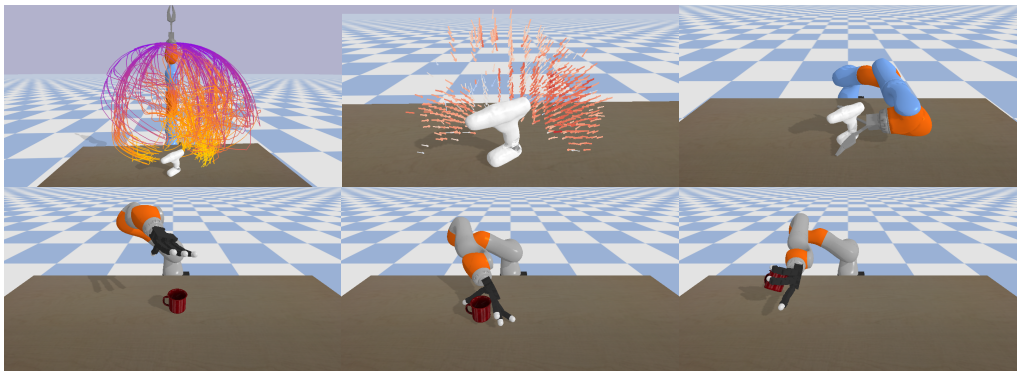

Figure 13: **Output visualizations.** (Top left) 200 trajectories randomly sampled from an A_s produced by ME-scs. Trajectories are displayed as a sequence of end effector positions. Color expresses temporality, from purple to yellow. The generated grasps are spread all over the operational space. (Top center) Visualization of the outcome descriptor associated with each solution from the same A_s. All the 724 solutions are displayed. The hotter the color, the higher the fitness. The regular space between the arrow's initial points expresses A_s predefined sampling. The best-performing solutions are the ones that minimize the energy cost while maximizing grasp stability. (Top right) Grasping position corresponding to the best-performing individual from the same A_s ($f = 0.95$). (Bottom) A sequence of a randomly sampled success from a generated A_s on the Allegro hand, from left to right.

6.1 Top performing methods vs promising methods

6.1.1 ME-scs

The **high performances of ME-scs variants on grasping is the most striking result** obtained in the above-presented experiments. Grasping rewards greedy exploitation of previously found successes; focusing on local search from successful solutions significantly increases the probability of finding new ones. We can distinguish success-greedy (ME-scs-based) from fitness-greedy (ME-fit-based) variants, respectively focusing on finding new successes or optimizing the quality of already found ones. The gap in performances obtained with those methods compared to QD state-of-the-art shows that **there is room for developing QD algorithms that are suited to address tasks with similar properties**.

6.1.2 SERENE

One can easily create a domain that would show the limits of a ME-scs-like approach, by making the exploration-exploitation tradeoff proposed by SERENE a requirement for success. Such a domain would have the following characteristics: 1) the goal space $\mathcal{G}$ must consist of several distinct regions of $\mathcal{B}_{reach}$ that would require several mutations to move from one sub-region of $\mathcal{G}$ to another one, and 2) making each sub-region of $\mathcal{G}$ large enough to saturate ME-scs's population with solutions from a single region after having discovered it. Paolo et al.'s redundant arms or curling might be a good example of domains that satisfy those criteria. In this case, SERENE should outperform ME-scs, as its exploration-exploitation mechanisms would eventually lead to the discovery and exploration of each sub-regions of the goal space – while ME-scs would be trapped into the first found sub-region. It would confirm the difference in properties of those two algorithms: **SERENE balances exploration with exploitation**, which might block

its execution into a standard NS algorithm if the rewarding regions are very hard to reach (as discussed in Paolo et al.), while **ME-scs explores until a reward zone is found**, and **then focuses on the exploitation of this first entry point** to generate as much diversity as possible while concurrently optimizing the reward of each rewarded solutions.

Beyond those theoretical questions, one might ask **what kind of non-toy problems can be addressed by each of those approaches**, considering the above-discussed properties. The present study considers the task of grasping, a key task for both concrete industrial cases (Mészáros et al. (2022)) and open-ended scenarios in robotics (Brohan et al. (2022)). The experimental results suggest that ME-scs's focus on exploitation is the most promising approach to generating a large repertoire of diverse high-performing grasping solutions. **Similar properties should arise in many other robotics manipulation tasks**, as most of them share some of the challenges studied in the present work. It includes behavioral sparsity, misalignment between the behavioral space and the targeted task, and hard-to-explore localized regions that concentrate the fitness function support.

6.1.3 CMA-MAE

CMA-MAE (Fontaine and Nikolaidis (2023)) is an extension of CMA-ME (Fontaine et al. (2020)) that alleviates some of its weaknesses – its inability to efficiently illuminate a behavior space when facing flat fitness landscapes. This CMA-ME limitation is visible here. While CMA-MAE outperforms most of the compared methods, **its performances are still way below ME-scs variants**. One might argue that **grasping rewards the greedy exploitation of previously found solutions**, as successful individuals lie in a limited part of the genotype space. CMA-MAE would therefore conduct a better exploration-exploitation compromise than a ME-scs method, resulting in lower grasping performances but better generalization capabilities on other tasks. This point can be nuanced by the fact that **grasping is itself submitted to discontinuities that require exploration to mutate one grasp to another successful trajectory that applies forces elsewhere on the object**. It might explain the difference in top-fitness performances for CMA-ME and CMA-MAE on the two tested robots. However, **we cannot state to what extent CMA-MAE struggles in tasks submitted to sparse behavior from those grasping experiments**. As the α parameter of CMA-MAE allows to control its tendency to explore or to behave similarly to a single-objective optimizer, it would be interesting to study how to make CMA-MAE more robust to sparse interaction problems. In particular, how to design QD algorithms that efficiently balance exploration and exploitation in those tasks, as looking for exploration might result in poor exploration.

6.1.4 NSMBS

NSMBS (Morel et al. (2022)) authors introduce their method as a way to address the challenges inherent to the task of grasping. The present work shows that NSMBS is not the only method that can efficiently address grasping. **Our results show that other methods – like ME-scs variants – perform way better on the task**. Indeed, NSMBS's ability to explore several behavior spaces during the evolutionary process has not been used. One might argue that: 1) making NSMBS evolve on multiple behavior spaces might lead to better results, and 2) that NSMBS's capacity to optimize several $\mathcal{B}_i$ allows producing an outcome repertoire with a high diversity on multiple components of the trajectory (e.g. how the end effector approaches the object, and how forces are applied on it). The visualization of individuals from a success archive produced by ME-scs (Figure 13) shows that **choosing a single behavior space ($\boldsymbol{\phi_{\mathcal{B}}(\tau) = X_a^{touch}}$) does not result in limited diversity of grasping trajectory**. It is worth noting that the trajectories cover the whole operational space. Exploring the space of object-gripper contact points results in diverse ways to approach the object, as some opposite points on the object's surface are not likely to be reached with a similar approach – especially if we also optimize the energy consumption through

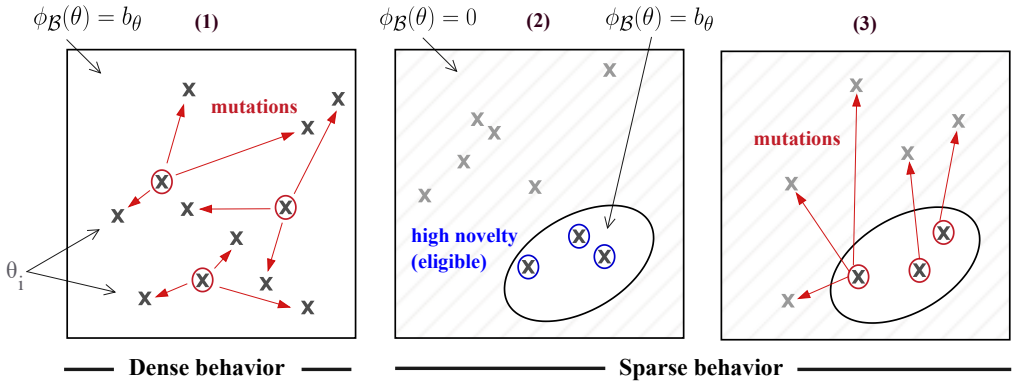

Figure 14: **How novelty can be deceptive in sparse behavior domains.** (1) In dense behavior domains, novelty-guided optimization selects highly evolvable individuals that efficiently explore the behavior space. (2) When the behavioral landscape is flat, eligibility pushes the population toward the support of $\phi_{\mathcal{B}}$ through novelty-based selection. (3) Selection of highly evolvable individuals in sparse behavior domains can push the population out of the region of interest due to their high instability in $\mathcal{B}$. The novelty is here deceptive.

fitness optimization. To avoid studying too many parameters simultaneously, we decided to let an in-depth analysis of multiBD on grasping for future work.

Identifying the best-performing method for doing RIBS on grasping is the main purpose of this paper. However, this study investigates a new kind of optimization problem – the sparse behavioral domains – in which some commonly admitted properties of QD algorithms do not hold. The most unexpected one is the detrimental role of novelty on the exploration capabilities of QD algorithms. This point is discussed in the following section.

6.2 On the detrimental role of novelty in sparse interaction domains

Novelty-guided algorithms are well known for efficiently exploring a targeted behavior space (Paolo et al. (2021)). **Our results show that this property does not hold in sparse interaction context**, as NS explores similarly – or significantly worse – than quality-guided methods or a simple random search. We attribute this statement to the role of evolvability in novelty-guided methods. By applying pressure on novelty for selection, novelty-guided methods favor highly evolvable individuals (Doncieux et al. (2020)). **The selected solutions are the most behaviorally unstable ones, which are the more likely to fall into the non-eligible regions of the search space** (Figure 14). This is why NS performs similarly to quality-guided methods on the problem submitted to strong behavioral sparsity (*wsg50_far*), while it reported the worst $cvg(A_o)$ performances on the less behavioral sparse problem (*kuka_close*).

The obtained results show that in sparse interaction context, **using methods that directly optimize $cvg(\mathcal{B})$ is not the best strategy to optimize it**. As discussed above, **the novelty is here *deceptive* regarding exploration**, as maximizing the evolvability is not the best strategy to explore a specific part of the targeted space in this context. Theoretical and experimental evidence regarding this specific phenomenon is left for future work.

Note that this point is a good example of why QD practitioners should include QD methods of different nature in their experiments, as the above-stressed property is shared by NS and ME derivated methods that are guided by novelty. These comparisons have been made possible

thanks to the proposed evaluation framework, which is discussed in the next section.

6.3 On the evaluation framework

This experiment shows that the proposed evaluation framework has many advantages for studying QD methods: it is **easy to implement, interpret and visualize**. By distinguishing the evaluation from the algorithmic internal components, **methods based on different kind of archives can fairly be compared**. Such an evaluation framework can similarly be used to compare methods with several behavioral spaces (Kim et al. (2021), Morel et al. (2022)) with QD state-of-the-art approaches. It also has the benefit of **making the outcome of QD algorithms explicit** on a given problem. We hope this approach will be exploited in future QD works to allow more flexibility in algorithm design without compromising the experimental studies' interpretability or exhaustiveness.

A limitation of this evaluation is that it might favor ME-derivated variants – as the sampling within the container matches the sampling of the outcome archive. An output structured archive has however many benefits: most of the QD works focus on SA-based methods, and it allows easy visualization, analysis, and exploitation. Practitioners who want to have fine control over granularity according to expert knowledge can set a specific sampling (like in this work), or can rely on a CVT sampling to fix the maximal size of the outcome archive without having to design the grid cell sampling.

7 Conclusion

This work investigates Quality-Diversity (QD) methods under sparse reward and sparse interactions problems through the application of grasping in robotics. What was arguably the most promising QD state-of-the-art methods for this domain (CMA-MAE, SERENE, and NSMBS) are significantly outperformed by variants of MAP-Elites that select non-null fitness solutions in priority. As a result, **an algorithm that is as simple as a standard MAP-Elites appeared to successfully generate a large set of diverse and high-performing solutions on multiple grasping domains**.

The best-performing methods can thus be used to **automatically generate datasets of grasping trajectories**. As the access to demonstrations is a **key matter to solve grasping**, we believe that such dataset generators can provide significant help in the resolution of this task.

Our results suggest that addressing a task submitted to behavioral sparsity can lead to counterintuitive results. In particular, **explicitly guiding an algorithm to optimize the exploration of an outcome space favors selecting the most evolvable and unstable solutions, pushing the offspring out of the behavioral function support**. One might parallel seminal works in Novelty Search and deceptive reward, in which directly optimizing fitness might lead to poor optimization. In this context, **behavior sparsity results in deceptive novelty**.

This paper opens many perspectives for the QD field regarding theoretical and practical matters: Do this work's observations hold in other robotic manipulation tasks? How to design more advanced algorithms that outperform ME-scs variants on grasping? How to avoid the sparse interaction setup through another behavioral characterization without compromising the algorithm's efficiency, as well as the output's interpretability and exploitability? More generally, we hope this work will incite the QD community on evolutionary robotics to tackle more complex problems, which are likely to result in breakthroughs in the field.

8 Acknowledgement

This work was supported by the Sorbonne Center for Artificial Intelligence, the German Ministry of Education and Research (BMBF) (01IS21080), and the French Agence Nationale de la Recherche (ANR) (ANR-21-FAI1-0004) - Learn2Grasp. It has received funding from the

European Commission's Horizon Europe Framework Programme under grant agreement No 101070381 and from the European Union's Horizon Europe Framework Programme under grant agreement No 101070596. This work was performed using HPC resources from GENCI-IDRIS (Grant 20XX-AD011014320). Many thanks Emily Clement for her writing advices, and Alessia Loi, Charly Pecqueux-Guezenec, and Olivier Serris for their help and feedback.

References

Anne, T. and Mouret, J.-B. (2023). Multi-task multi-behavior map-elites.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. (2022). Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Bruneton, J.-P., Cazenille, L., Douin, A., and Reverdy, V. (2019). Exploration and exploitation in symbolic regression using quality-diversity and evolutionary strategies algorithms. *arXiv preprint arXiv:1906.03959*.

Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., and Dollar, A. M. (2015). Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*.

Conti, E., Madhavan, V., Petroski Such, F., Lehman, J., Stanley, K., and Clune, J. (2018). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31.

Coumans, E. and Bai, Y. (2016). Pybullet, a python module for physics simulation for games, robotics and machine learning.

Cully, A. (2019). Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 81–89.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Cully, A. and Demiris, Y. (2017). Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259.

Cully, A., Mouret, J.-B., and Doncieux, S. (2022). Quality-diversity optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 864–889.

De Coninck, E., Verbelen, T., Van Molle, P., Simoens, P., and Dhoedt, B. (2020). Learning robots to grasp by demonstration. *Robotics and Autonomous Systems*, 127:103474.

Doncieux, S., Laflaquière, A., and Coninx, A. (2019). Novelty search: a theoretical perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 99–106.

Doncieux, S., Paolo, G., Laflaquière, A., and Coninx, A. (2020). Novelty search makes evolvability inevitable. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 85–93.

Faldor, M., Chalumeau, F., Flageat, M., and Cully, A. (2023). Map-elites with descriptor-conditioned gradients and archive distillation into a single policy. *arXiv preprint arXiv:2303.03832*.

Fang, H.-S., Wang, C., Gou, M., and Lu, C. (2020). Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453.

Flageat, M. and Cully, A. (2020). Fast and stable map-elites in noisy domains using deep grids. In *Artificial Life Conference Proceedings 32*, pages 273–282. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ....

Flageat, M. and Cully, A. (2023). Uncertain quality-diversity: Evaluation methodology and new methods for quality-diversity in uncertain domains. *IEEE Transactions on Evolutionary Computation*.

Fontaine, M. and Nikolaidis, S. (2021). Differentiable quality diversity. *Advances in Neural Information Processing Systems*, 34:10040–10052.

Fontaine, M. and Nikolaidis, S. (2023). Covariance matrix adaptation map-annealing. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 456–465.

Fontaine, M. C., Togelius, J., Nikolaidis, S., and Hoover, A. K. (2020). Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 94–102.

Futagi, Y., Toribe, Y., Suzuki, Y., et al. (2012). The grasp reflex and moro reflex in infants: hierarchy of primitive reflex responses. *International journal of pediatrics*, 2012.

Gaier, A., Asteroth, A., and Mouret, J.-B. (2018). Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation*, 26(3):381–410.

Hansen, N. (2016). The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.

Hodson, H. (2018). A gripping problem: designing machines that can grasp and manipulate objects with anything approaching human levels of dexterity is first on the to-do list for robotics. *Nature*.

Kim, S., Coninx, A., and Doncieux, S. (2021). From exploration to control: learning object manipulation skills through novelty search and local adaptation. *Robotics and Autonomous Systems*, 136:103710.

Lehman, J. and Stanley, K. O. (2010). Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 103–110.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evo. comp.*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436.

Lobbezoo, A., Qian, Y., and Kwon, H.-J. (2021). Reinforcement learning for pick and place operations in robotics: A survey. *Robotics*, 10(3):105.

Macé, V., Boige, R., Chalumeau, F., Pierrot, T., Richard, G., and Perrin-Gilbert, N. (2023). The quality-diversity transformer: Generating behavior-conditioned trajectories with decision transformers. *arXiv preprint arXiv:2303.16207*.

Mészáros, A., Franzese, G., and Kober, J. (2022). Learning to pick at non-zero-velocity from interactive demonstrations. *IEEE Robotics and Automation Letters*, 7(3):6052–6059.

Morel, A., Kunimoto, Y., Coninx, A., and Doncieux, S. (2022). Automatic acquisition of a repertoire of diverse grasping trajectories through behavior shaping and novelty search. *arXiv preprint arXiv:2205.08189*.

Morrison, D., Corke, P., and Leitner, J. (2020). Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3):4368–4375.

Mosbach, M. and Behnke, S. (2023). Learning generalizable tool use with non-rigid grasp-pose registration. *arXiv preprint arXiv:2307.16499*.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Needham, A. W. and Nelson, E. L. (2023). How babies use their hands to learn about objects: Exploration, reach-to-grasp, manipulation, and tool use. *Wiley Interdisciplinary Reviews: Cognitive Science*, page e1661.

Nguyen, V.-D. (1988). Constructing force-closure grasps. *The International Journal of Robotics Research*, 7(3):3–16.

Nilsson, O. and Cully, A. (2021). Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 866–875.

Paolo, G., Coninx, A., Doncieux, S., and Laflaquière, A. (2021). Sparse reward exploration via novelty search and emitters. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 154–162.

Paolo, G., Laflaquiere, A., Coninx, A., and Doncieux, S. (2020). Unsupervised learning and exploration of reachable outcome space. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2379–2385. IEEE.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Pugh, J. K., Soros, L. B., Szerlip, P. A., and Stanley, K. O. (2015). Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 967–974.

Salehi, A., Coninx, A., and Doncieux, S. (2021). Br-ns: an archive-less approach to novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 172–179.

Salehi, A., Coninx, A., and Doncieux, S. (2022). Few-shot quality-diversity optimization. *IEEE Robotics and Automation Letters*, 7(2):4424–4431.

Shorten, D. and Nitschke, G. (2014). How evolvable is novelty search? In *2014 IEEE International Conference on Evolvable Systems*, pages 125–132. IEEE.

Sigaud, O. (2022). Combining evolution and deep reinforcement learning for policy search: a survey. *ACM Transactions on Evolutionary Learning*.

Tjanaka, B., Fontaine, M. C., Lee, D. H., Zhang, Y., Balam, N. R., Dennler, N., Garlanka, S. S., Klapsis, N. D., and Nikolaidis, S. (2023). pyribs: A bare-bones python library for quality diversity optimization. *arXiv preprint arXiv:2303.00191*.

Vassiliades, V., Chatzilygeroudis, K., and Mouret, J.-B. (2017). Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22(4):623–630.

Vassiliades, V. and Mouret, J.-B. (2018). Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 149–156.

Verhellen, J. and Van den Abeele, J. (2020). Illuminating elite patches of chemical space. *Chemical science*, 11(42):11485–11491.

Wang, P., Manhardt, F., Minciullo, L., Garattoni, L., Meier, S., Navab, N., and Busam, B. (2021). Demograsp: Few-shot learning for robotic grasping with human demonstration. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5733–5740. IEEE.

Wiegand, R. P. (2020). The objective of simple novelty search. In *The Thirty-Third International Flairs Conference*.

Yang, J., Tan, W., Jin, C., Liu, B., Fu, J., Song, R., and Wang, L. (2023). Pave the way to grasp anything: Transferring foundation models for universal pick-place robots. *arXiv preprint arXiv:2306.05716*.

Zardini, E., Zappetti, D., Zambrano, D., Iacca, G., and Floreano, D. (2021). Seeking quality diversity in evolutionary co-design of morphology and control of soft tensegrity modular robots. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 189–197.

Zhang, H., Tang, J., Sun, S., and Lan, X. (2022). Robotic grasping from classical to modern: A survey. *arXiv preprint arXiv:2202.03631*.

Supplementary Materials

A Architecture of policies for grasping

QD domains for robotics have led to different architectures of policies: open-loop controllers (Cully et al. (2015)), multiple layer perceptron (Doncieux et al. (2019)), or evolving neural networks (Pugh et al. (2016)). Our controller for grasping consists of an open-loop trajectory guided by 3 waypoints. We initialize the gripper position to open and close it during the episode. Each genome follows the bellow described pattern:

$$\theta = (X_1, \alpha_1, X_2, \alpha_2, X_3, \alpha_3)$$

Each $X_i = (x_i, y_i, z_i)$ coordinates define a waypoint in the cartesian space, and each $\alpha_i = (\alpha_i^p, \alpha_i^r, \alpha_i^y)$ values define the orientation the end effector must match at each waypoint in Euler angles with respect to the world basis. All those values are normalized to lie between -1 and 1, according to the predefined limits of the operational space. To evaluate an individual, we first project back the normalized coordinates into the Cartesian space and then apply a polynomial interpolation such that each of the 3 points should be respectively reached at $T/3$, $2T/3$ and T steps, where T is the episode length. Each robot is initialized at a fixed position.

When the end effector first touches the object, the gripper is closed with constant force. This mechanism is inspired by the *Palmar Grasp Reflex*, which makes newborn infants closes their hands when pressure and touch are applied to the palm (Futagi et al. (2012)). It is also well known in the robotics litterature that non-zero-velocity grasps make the problem significantly more challenging (Mészáros et al. (2022)).

While parallel grippers are controllable with a single value, dexterous hands provide more degrees of freedom. To exploit the dexterity of the Allegro hand without making the problem too complex, we have defined a set of grasp primitives that correspond to synergies that could be applied to a real Allegro hand. On the dexterous hand's domain, we have thus added a value l_{gp} to each genome that describes the label of the grasp primitive. Details of the n_{gp} designed primitives are provided in the hyperparameters section. The $[-1, 1]$ interval is uniformly sampled into n_{gp} parts, such that l_{gp} can be associated with a unique grasp primitive label. The corresponding grasp primitive is applied during the evaluation as soon as the object is first touched.

B Fitness

To bypass the fitness sparsity of grasping, many works in robotics or RL relies on *reward-shaping* to bootstrap the learning. It consists of manually designing a modular reward function to push the agent toward validating the sparse success criterion (Lobbezoo et al. (2021)). Many open-source environments for manipulation tasks in robotics provide a reward signal by default that makes the problem tractable with state-of-the-art methods. Consequently, the rare QD works on such domains deal with dense reward functions (Salehi et al. (2022)).

The main drawback of this approach is that it strongly biases how to solve the task. If we optimize a fitness function that rewards the agent when the object is grasped and when its end effector is getting closer to the object's center of mass, we do not learn "grasping" policies. Instead, we are addressing the derivated task: "grasp by applying forces around the object center of mass as fast as possible."

This is problematic for many reasons: firstly, it is unsatisfying from a learning point of view, as what we ultimately are interested in here is to make an agent learn to grasp; secondly, such a hand-designed fitness puts heavy constraints on the diversity of the generated solutions; finally, those works are not likely to result into interesting applications if the developed methods are not easily deployed on other sub-tasks one might want to address (e.g. grasping cautiously, grasping for a specific affordance), which is contradictory with the dependence to a certain way of solving the problem.

Instead, we here condition the obtention of reward on the validation of the binary sparse success criteria. Nevertheless, we are interested in optimizing a fitness signal, as we here want to study grasping from a RIBS perspective. We have therefore designed the following fitness function:

$$f(\tau_i) = \begin{cases} 0.5 \times f_{ec}(\tau_i) + 0.5 \times f_{gs}(\tau_i) & \text{if} \quad f_c(\tau_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $f_c : S_\tau \to \{0, 1\}$ is the grasping binary *success criterion*, $f_{ec} : S_\tau \to \mathbb{R}^+$ is the *energy consumption fitness*, $f_{gs} : S_\tau \to \mathbb{R}^+$ is the *grasp stability fitness*. $f_{ec}(\tau)$ and $f_{gs}(\tau)$ are normalized to lie in $[0, 0.5]$. In

practice, f_c returns 1 if the following conditions are verified for N_g steps consecutively: 1) this object does not touch the table; 2) the object does not touch the floor; 3) the robot does not touch the table; 4) the end effector of the robot is touching the object; and 5) there is no penetration between the 3D models of the robot and the object. The energy consumption fitness is the sum of the applied torques on each joint throughout the episode. The grasp stability fitness is measured as follows:

$$f_{gs}(\tau_i) = (-1) * \left(\frac{var(\{X_a^{touch_{i=1,...,T}}\}) + var(\{X_{obj}^{touch_{i=1,...,T}}\})}{n^{max\_touch} \times n^{cont\_touch}} + \zeta_d \right)$$

with $X_a^{touch_i}$ being the end effector-object contact point on the agent, $X_{obj}^{touch_i}$ being the end effector-object contact point on the object, n^{max_touch} being the number of iterations in which there has been contact between the agent and the object, n^{cont_touch} being the maximal number of iterations of continuous contact from the first touch between the agent and the object, and ζ_d being an additional cost that penalizes discontinuous interactions. The cost ζ_d is defined as the number of iterations from the first step at which the agent stops to touch the object until the end of the episode. The whole fitness is set as a negative function to maximize. The discontinuity cost is the dominating member of f_{gs} until continuous grasps are found. As f_{gs} cannot be lower than $1 - T$ (meaning that the object is touched during the first step only) and larger than 0, we rely on the assumption that $f_{gs} \in [-T, 0]$ to project it into the expected $[0, 0.5]$ interval. The normalization of f_{ec} relies on extrema values measured from executions of fitness-free methods (Random and NS).

C Qd-scores

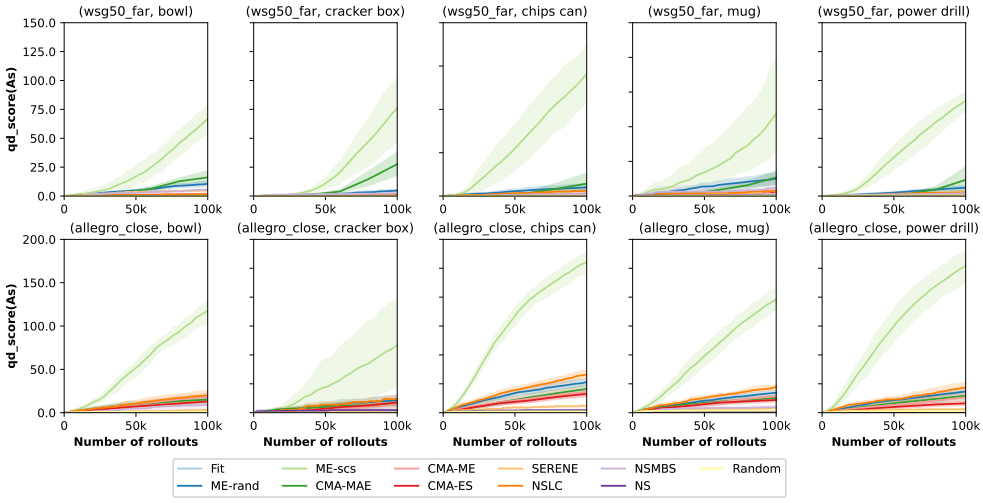

Figure 15: **Qd-scores measured throughout the evolutionary process for state-of-the-art QD algorithms**. The large differences in success archive size make the qd-score saturated by the number of solutions. This measure does not provide information on the quality of the generated successes.

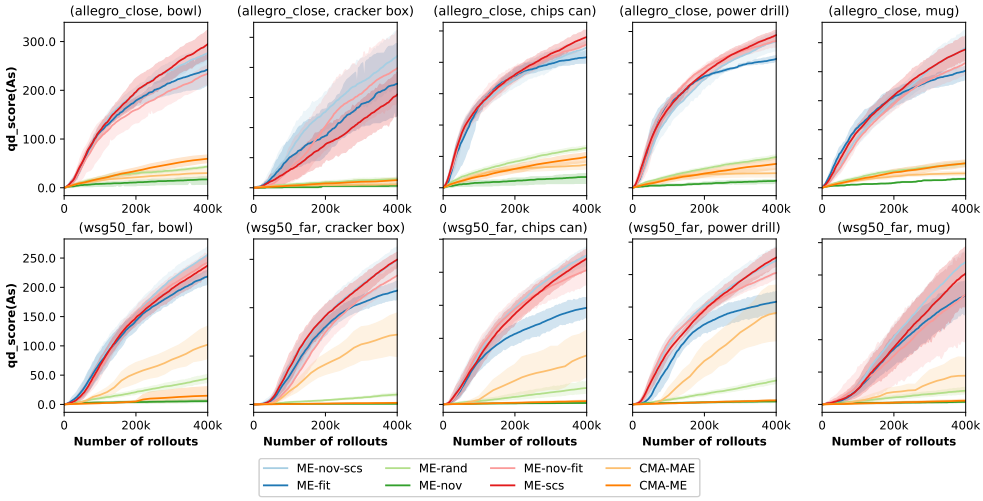

Figure 16: **Qd-scores measured throughout the evolutionary process for different variants of MAP-Elites.** Results are essentially the same as those delivered by the coverage of A_s: fitness-guided ME-* variants can get stuck into local minima.

D Success archive coverage of CMA-* variants

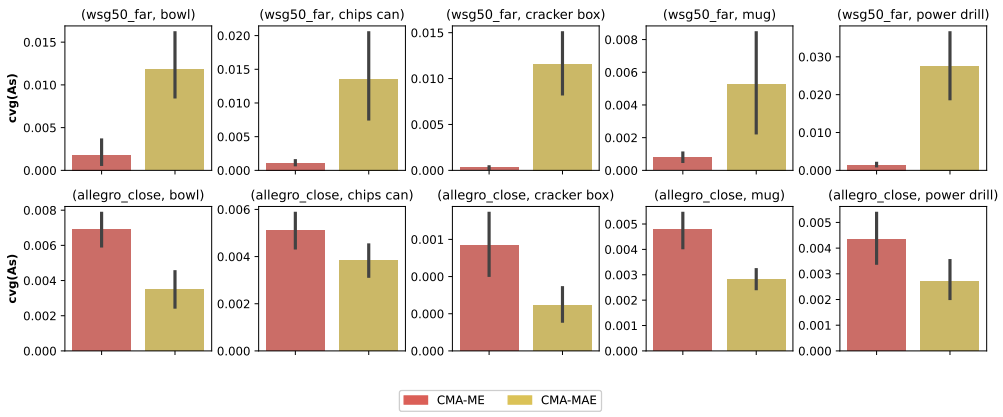

Figure 17: **CMA-ME vs CMA-MAE on the coverage of the success archive** Over 10 seeds, after 400k evaluations. While CMA-MAE explores $\mathcal{G}$ more efficiently on the most difficult task (*kuka_wsg50_far*), CMA-ME is better on the easier one (*kuka_allegro_close*. We attribute this observation to the behavior sparsity of the task. CMA-ME is well known for pushing the search away from previously found solutions (Fontaine and Nikolaidis (2023)). It limits CMA-ME from optimizing the performances of already-found solutions. CMA-MAE alleviates this issue with a rolling fitness threshold that conditions elites' insertion into the container. CMA-MAE spends more budget in the same region of the search space, limiting its exploration capabilities. This mechanism allows CMA-MAE to better exploit previously found successes to find new ones on the most difficult task. Such a strategy might keep the evolutionary process into local minima on the Allegro task. CMA-ME exploration capabilities help to escape those minima, while it can be detrimental under more sparse behavioral domains.