Density Descent for Diversity Optimization

David H. Lee
dhlee@usc.edu
University of Southern California
Los Angeles, California, USA

Anishalakshmi V. Palaparthi
palapart@usc.edu
University of Southern California
Los Angeles, California, USA

Matthew C. Fontaine
mfontain@usc.edu
University of Southern California
Los Angeles, California, USA

Bryon Tjanaka
tjanaka@usc.edu
University of Southern California
Los Angeles, California, USA

Stefanos Nikolaidis
nikolaid@usc.edu
University of Southern California
Los Angeles, California, USA

ABSTRACT

Diversity optimization seeks to discover a set of solutions that elicit diverse features. Prior work has proposed Novelty Search (NS), which, given a current set of solutions, seeks to expand the set by finding points in areas of low density in the feature space. However, to estimate density, NS relies on a heuristic that considers the k-nearest neighbors of the search point in the feature space, which yields a weaker stability guarantee. We propose Density Descent Search (DDS), an algorithm that explores the feature space via CMA-ES on a continuous density estimate of the feature space that also provides a stronger stability guarantee. We experiment with DDS and two density estimation methods: kernel density estimation (KDE) and continuous normalizing flow (CNF). On several standard diversity optimization benchmarks, DDS outperforms NS, the recently proposed MAP-Annealing algorithm, and other state-of-the-art baselines. Additionally, we prove that DDS with KDE provides stronger stability guarantees than NS, making it more suitable for adaptive optimizers. Furthermore, we prove that NS is a special case of DDS that descends a KDE of the feature space.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**; • **Mathematics of computing** → *Density estimation.*

KEYWORDS

quality diversity, kernel density estimation, novelty search

ACM Reference Format:
David H. Lee, Anishalakshmi V. Palaparthi, Matthew C. Fontaine, Bryon Tjanaka, and Stefanos Nikolaidis. 2024. Density Descent for Diversity Optimization. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3638529.3654001

1 INTRODUCTION

We study how stable, continuous approximations of density can accelerate the search for diverse solutions to optimization problems.

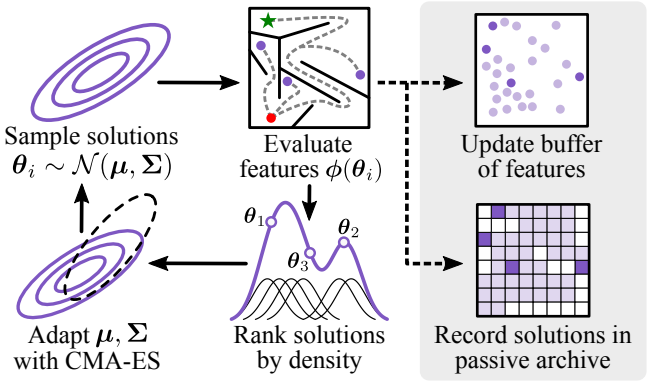


Figure 1: We propose density descent search (DDS) for solving diversity optimization (DO) problems. DDS first draws solutions from a Gaussian $\mathcal{N}(\mu, \Sigma)$. After computing the solution features (in this case, the final position of the robot in a maze), DDS ranks solutions by density. This density ranking is passed to CMA-ES, which updates the search distribution to sample solutions with lower density on the next iteration. Concurrently, solutions are stored in a buffer that forms the basis for density estimates, and in a passive archive that tracks all discovered solutions.

There is a range of applications where searching directly for behavioral diversity results in finding suboptimal solutions that act as "stepping stones," mitigating convergence to local optima [16]. A classic example is the problem of training an agent to reach a target position in a *deceptive maze* [29]. There, directly minimizing the distance between the agent's final position and a target goal causes the agent to get stuck. On the other hand, the problem can be solved by ignoring the objective of directly reaching the target and instead attempting to find a diverse range of agents, each of which reaches a different region of the maze.

We refer to the problem of finding a range of solutions that are diverse with respect to prespecified features as diversity optimization (DO). We characterize DO as a special instance of quality diversity optimization (QD) [37]. Like DO, QD seeks a diverse set of solutions, but QD also considers an objective function that the solutions must optimize. For instance, in deceptive maze, one could add an energy consumption objective so that the goal is to find a set of agents that minimize energy consumption while moving to diverse regions in

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0494-9/24/07.
https://doi.org/10.1145/3638529.3654001

the maze. Thus, DO is an instance of QD where the objective value of all solutions is constant.

In DO and QD, discovering solutions with new features is challenging because the mapping from solutions to features is complex and non-invertible [5]. For example, in maze exploration, it is not known *a priori* how to produce an agent that travels to a given (x, y) position.

A classic method applied to DO is Novelty Search (NS) [29]. NS retains an *archive* of previously found individual solutions. It aims to expand the archive by finding solutions that are far in the feature space from existing solutions in the archive. Specifically, NS optimizes for solutions with a high *novelty score*, where novelty score is the average distance in feature space from a solution to its k-nearest neighbors in the archive.

While the novelty score is framed in terms of distance, we study its interpretation as an approximation of *density* [5], where density is directly proportional to the number of solutions in a region of feature space. A high novelty score indicates that a solution's features are far from those of its k-nearest neighbors in the archive, i.e., it is located in an area of the feature space with low density.

An alternative approach to DO is to apply general-purpose QD algorithms and assume that all solutions in the search space have identical quality, i.e., equal to some constant. Covariance Matrix Adaptation MAP-Annealing (CMA-MAE) [12] is a state-of-the-art QD algorithm that optimizes for archive improvement with the CMA-ES [23] optimizer. CMA-MAE retains a discrete archive of solutions in feature space. When applied to solve a DO problem by ignoring the solution quality, this archive naturally becomes a histogram that represents the distribution of solutions in feature space, with lower values of the histogram indicating lower density. CMA-MAE then performs *density descent* on this histogram, where it continually seeks to fill the areas of low density.

We emphasize that, to efficiently explore feature space, both NS and CMA-MAE leverage density estimates of solutions in feature space — NS is guided by novelty score, while CMA-MAE reads from its histogram. However, both of these density estimates have drawbacks. On one hand, the novelty score in NS is *continuous* but provides a *weaker* stability guarantee, meaning that its value can change arbitrarily when new features are discovered. On the other hand, the histogram in CMA-MAE provides a *stronger* stability guarantee but utilizes a *discrete* approximation that gives flat gradient signals on its bins.

Our key insight is to overcome the drawbacks of current density estimation methods in DO by introducing continuous, stable approximations of the solution density in feature space. This insight results in the following contributions: (1) We propose density descent search (DDS; Fig. 1), an algorithm that queries continuous density estimates to guide an optimizer, in our case CMA-ES, to search for solutions that are diverse in the feature space (Sec. 4). We propose two variants of DDS: DDS-KDE leverages non-parametric Kernel Density Estimation (KDE) [4], while DDS-CNF learns the underlying density function with continuous normalizing flow [31]. (2) We show that, when combined with a ranking-based optimizer like CMA-ES, NS reduces to a special case of DDS-KDE (Sec. 5). (3) We prove that KDE provides stronger algorithmic stability guarantees than novelty score (Sec. 5). (4) We demonstrate that our DDS algorithms outperform prior work on 3 out of 4 domains (Sec. 6). (5) We show that our algorithms perform well on multi-dimensional feature spaces, which currently present significant challenges to QD and DO algorithms.

2 PROBLEM DEFINITION

Quality Diversity Optimization (QD). For solution $\boldsymbol{\theta} \in \mathbb{R}^n$, QD assumes an *objective* function $f : \mathbb{R}^n \to \mathbb{R}$ and m *feature* functions, jointly represented as a vector-valued function $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^m$. We refer to the image of $\boldsymbol{\phi}$ as the *feature space* S. The goal of QD is to find, for each $\boldsymbol{s} \in S$, a solution $\boldsymbol{\theta}$ where $f(\boldsymbol{\theta})$ is maximized and $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{s}$.

Diversity Optimization (DO). DO is a special case of QD where the objective is constant, i.e., $f(\boldsymbol{\theta}) = C$. The goal of DO is to find, for each $\boldsymbol{s} \in S$, a solution $\boldsymbol{\theta}$ where $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{s}$.

3 BACKGROUND

3.1 QD and DO Algorithms

Various methods address the QD and DO problems by relaxing them to the problem of finding an *archive* (i.e., a set) $\mathcal{A}$ of representative solutions. The structure of the archive defines two major families of algorithms: those based on Novelty Search and those based on MAP-Elites.

Novelty Search (NS). The key insight of NS [29] is to discover diverse solutions in feature space by optimizing for solutions that are "novel" with respect to a current set of solutions. Given a set of features $\mathcal{X} \subseteq \mathbb{R}^m$, the novelty of a solution $\boldsymbol{\theta} \in \mathbb{R}^n$ and its features $\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{\theta})$ relative to $\mathcal{X}$ is encapsulated in its *novelty score*, denoted $\rho(\boldsymbol{y}; \mathcal{X})$:

$$\rho(\boldsymbol{y}; \mathcal{X}) = \frac{1}{k} \sum_{\boldsymbol{y}' \in N_k(\boldsymbol{y}; \mathcal{X})} \text{dist}(\boldsymbol{y}, \boldsymbol{y}') \tag{1}$$

where $N_k(\boldsymbol{y}; \mathcal{X})$ is the set of k-nearest neighbors to $\boldsymbol{y}$ in $\mathcal{X}$, and dist is a distance function — henceforth, we consider dist to be some norm $\|\cdot\|$.

NS maintains an archive $\mathcal{A}$ of unbounded size and generates solutions with an underlying optimizer, traditionally a genetic algorithm. For each solution $\boldsymbol{\theta}$ produced by the optimizer, NS computes the novelty score with respect to the current archive, i.e., $\rho(\boldsymbol{y}; \mathcal{A})$. If $\rho(\boldsymbol{y}; \mathcal{A})$ exceeds an *acceptance threshold*, then $(\boldsymbol{\theta}, \boldsymbol{y})$ is added to the archive. In this manner, NS gradually adds novel solutions to the archive and explores the feature space.

Importantly, the novelty score is non-stationary in that the novelty of a solution $\boldsymbol{\theta}$ changes as the archive $\mathcal{A}$ is updated. We prove in Sec. 5 that the degree of non-stationarity in novelty score is unbounded for general feature spaces, resulting in significant optimization challenges for adaptive optimizers such as CMA-ES.

While we consider general optimization domains in this work, we note that prior works [6] have specialized NS for reinforcement learning domains, particularly the case when no feature function is assumed [8, 36].

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites). While NS was developed for DO, MAP-Elites [33] was developed for the general QD setting. Compared to the unstructured archive of NS, MAP-Elites divides the feature space into a predefined tessellation $T : \mathbb{R}^m \to \{1, \ldots, l\}$, where $e \in \{1, \ldots, l\}$ is a *cell* in the tessellation and l is the total number of cells. Given a cell e, the

MAP-Elites archive associates a solution $\boldsymbol{\theta}$ with cell e if and only if the solution's features are contained in e, i.e., $T(\boldsymbol{\phi}(\boldsymbol{\theta})) = e$. Moreover, MAP-Elites stores at most one solution for every cell in the tessellation.

During a QD search, MAP-Elites first draws solutions from a predefined distribution, e.g., a Gaussian, and inserts the solutions into the archive. Subsequently, MAP-Elites generates and inserts new solutions by mutating existing archive solutions with a genetic operator, such as the Iso+LineDD operator [47]. Importantly, when solutions inserted into the archive land in the same tessellation cell, MAP-Elites only retains the solution with the greatest objective value. Thus, MAP-Elites gradually collects high-performing solutions that have diverse features. We apply MAP-Elites to DO by setting a constant objective $f(\cdot) = C$ over the solution space.

The choice of tessellation in MAP-Elites can significantly impact scalability. The most common tessellation is a grid tessellation [33], which divides the feature space into equally-sized hyperrectangles. In high-dimensional feature spaces, grid tessellations require exponentially more memory due to the curse of dimensionality. Thus, a common alternative is the centroidal Voronoi tessellation (CVT) [46], which divides the feature space into l evenly-sized polytopes.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES). One recent line of QD algorithms has combined CMA-ES [23] with MAP-Elites. CMA-ES is a state-of-the-art single-objective optimizer that represents a population of solutions with a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Each iteration, CMA-ES draws λ solutions from the Gaussian. Based on the *rankings* (rather than the raw objectives) of the solutions, CMA-ES adapts the covariance matrix Σ to regions of higher-performing solutions. While CMA-ES is a derivative-free optimizer, it has been shown to approximate a natural gradient [1].

Covariance Matrix Adaptation MAP-Elites (CMA-ME). The first work to integrate CMA-ES with MAP-Elites was CMA-ME [14]. The key idea of CMA-ME is to optimize for archive improvement with CMA-ES. Namely, in addition to a MAP-Elites-style archive, CMA-ME maintains an instance of CMA-ES. The CMA-ES instance samples solutions from a Gaussian, and the solutions are ranked based on how much they improve the archive, e.g., solutions that found new cells in the archive are ranked high, while those that were not added at all are ranked low. With this *improvement ranking*, CMA-ES adapts the Gaussian to sample solutions that will further improve the archive. Note that in DO settings, since the objective is a constant value, the improvement ranking places solutions that filled a new cell ahead of solutions that did not add a new cell.

Covariance Matrix Adaptation MAP-Annealing (CMA-MAE). One limitation of CMA-ME is that it focuses too much on exploring for new cells in feature space rather than optimizing the objective [45]. CMA-MAE [12] addresses this problem by introducing an *archive learning rate* α. When $\alpha = 1$, CMA-MAE maintains the same exploration behavior as CMA-ME, but when $\alpha = 0$, CMA-MAE focuses solely on optimizing the objective, like CMA-ES. For $0 < \alpha < 1$, CMA-MAE blends between these two extremes, enabling it to both explore feature space and optimize the objective.

In DO settings, where the objective is constant ($f(\cdot) = C$), CMA-MAE naturally performs *density descent* [12]. Intuitively, the archive becomes a histogram that represents how many solutions have been found in each region of feature space. Lower values of the histogram indicate lower *density* of solutions, i.e., if a histogram bin has a low value, few solutions have been discovered in that region of feature space. CMA-MAE then seeks to *descend* the histogram by searching for solutions that fill the low-valued histogram bins.

3.2 Density Estimation

We consider two density estimation methods: Kernel Density Estimation and Continuous Normalizing Flows.

Kernel Density Estimation (KDE). KDE is a non-parametric density estimation method that does not make any assumptions on the underlying probability density distribution from which samples are drawn [4]. Given a set of features $\mathcal{X} \subseteq \mathbb{R}^m$, bandwidth parameter h, and kernel function $K(\cdot)$, KDE computes the density function $\hat{D} : \mathbb{R}^m \to \mathbb{R}$ for a given feature $\boldsymbol{y} \in \mathbb{R}^m$ as:

$$\hat{D}_h(\boldsymbol{y}; \mathcal{X}) = \frac{1}{|\mathcal{X}|h} \sum_{\boldsymbol{y}' \in \mathcal{X}} K\left(\frac{\|\boldsymbol{y} - \boldsymbol{y}'\|}{h}\right) \tag{2}$$

Prior work has thoroughly studied the problem of selecting bandwidth that accurately estimate the underlying density function [3, 40, 42]. While accurate density estimation is important for KDE, we show in Theorem 5.2 that larger bandwidth results in more stable density estimates (albeit at the cost of accuracy), which is beneficial to the proposed algorithms in Sec. 4.

When optimizing a density estimate (e.g., with gradient descent), KDEs have several advantages over histograms. First, the shape of a histogram depends on its bin size [48]. Second, the binning procedure leads to a discontinuous optimization landscape and flat gradient signals inside each bin [48]. In contrast, KDEs produce a smooth, continuous density function based on the location of the samples in the underlying distribution [4].

Continuous Normalizing Flow (CNF). CNF [31] is a generative modeling method that constructs a diffusion path between a simple distribution (e.g., a Gaussian distribution) and an unknown distribution. The diffusion path describes a mapping from a point on the simple distribution to a corresponding point on the unknown distribution such that their probability densities are roughly equal. CNFs enable sampling from probability distributions where direct sampling is difficult (e.g., distributions of images). This is accomplished by sampling from the simple distribution and transforming the sample via the learned diffusion path of the CNF. However, we utilize CNF to estimate the density of feature space for our proposed algorithms in Sec. 4.

4 DENSITY DESCENT SEARCH

We present Density Descent Search (DDS), a DO algorithm that efficiently explores the feature space by leveraging continuous density estimates. DDS maintains a buffer of features $\mathcal{B}$ that represents the distribution of features discovered so far. Based on this buffer, DDS models a density estimation of the feature distribution, and by querying this density estimate, DDS guides an adaptive optimizer to discover solutions in less-dense areas of feature space. As it searches for such solutions, DDS expands the discovered set of features. We provide an overview of DDS in Fig. 1 and describe the components of DDS below.

Density Estimation. The core of DDS is its representation of feature space density. We propose two variants of DDS that differ in

their density representation. The first variant, DDS-KDE, represents density with KDE. With the KDE, we compute density via Eq. 2, with $\mathcal{X}$ set to the buffer of features $\mathcal{B}$.

The second variant, DDS-CNF, estimates the density in feature space by learning a CNF between the standard normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and the observed feature space distribution. Similar to DDS-KDE, the feature space distribution is represented by the buffer $\mathcal{B}$. We compute a density estimate at any location in the feature space by integrating an ordinary differential equation [31]. Applying techniques from prior work [31], we represent the CNF with a neural network, and on each iteration, we finetune the network on the new distribution of features contained in the buffer.

KDE and CNF differ in how easily their smoothness can be controlled. On one hand, the shape of the KDE can be controlled with the bandwidth hyperparameter. Higher bandwidth leads to a smoother density estimate but conceals modes of the true density function, as illustrated in Fig. 2a. On the other hand, while CNF does not require selecting a bandwidth hyperparameter, the smoothness of the density estimation cannot be easily controlled, which undermines the performance of the algorithm (Sec. 6.3). We discuss the impacts that the smoothness of the density estimate has on DDS in Sec. 5.

Feature Buffer. To provide the basis for density estimates in feature space, DDS maintains a buffer $\mathcal{B}$ that stores the features of sampled solutions (line 4). This buffer represents the *observed* distribution of the feature space and is updated every time new solutions and features are discovered. In theory, the buffer can have infinite capacity, storing every feature ever encountered. In practice, the buffer can only retain a finite number of features due to computation and memory limitations. To decide which features to retain in the buffer, we manage the buffer with an optimal reservoir sampling algorithm [30]. This algorithm updates the buffer with online samples that accurately represent the distribution of features discovered so far by DDS (line 6).

Optimizer. DDS guides an adaptive optimizer to discover solutions in low-density regions of the feature space. Examples of such optimizers include xNES [17] and Adam [26]. We select CMA-ES as the underlying optimizer due to its reputation as a state-of-the-art optimizer [23] and its high performance in existing QD algorithms [11, 12, 14].

Passive Archive. Following prior work [38], to track how much of the feature space has been explored, DDS inserts all discovered solutions and features into a passive MAP-Elites-style (Sec. 3) archive $\mathcal{A}$. While DDS itself only uses the archive to record solutions and features, the archive is useful when computing metrics in experiments (Sec. 6.1).

Summary. Algorithm 1 shows how the components of DDS come together. DDS begins by initializing its various components (line 2-4). During the main loop (line 5), DDS samples solutions with the underlying optimizer (line 8). Since our optimizer is CMA-ES, sampling consists of drawing from a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Subsequently, DDS computes the features of the solutions (line 9) and estimates their feature space density (line 10). After sampling solutions, DDS updates its various components. For instance, in DDS-KDE, the density update (line 12) consists of replacing the feature buffer, and in DDS-CNF, the update involves fine-tuning the neural network on the feature buffer. Furthermore, solutions with lower density are ranked first on line 13, causing CMA-ES to adapt (line 14) towards less-dense regions of the feature space.

Algorithm 1: Density Descent Search (DDS)

1 **DDS** $(\boldsymbol{\phi}(\cdot), \mathit{density}, b, N, \lambda, \boldsymbol{\mu}_0, \sigma)$
  Input: Feature function $\boldsymbol{\phi}(\cdot)$, density estimator object $\mathit{density}$, buffer size b, number of iterations N, batch size λ, initial solution $\boldsymbol{\mu}_0$, and initial step size σ
  Result: Generates $N \cdot \lambda$ solutions, storing a representative subset of them in a passive archive $\mathcal{A}$.
2 Initialize CMA-ES search point $\boldsymbol{\mu} := \boldsymbol{\mu}_0$, search direction $\Sigma = \sigma \boldsymbol{I}$, and internal parameters $\boldsymbol{p}$
3 Initialize empty archive $\mathcal{A}$
4 Initialize empty buffer $\mathcal{B}$ of size b
5 **for** $\mathit{iter} \leftarrow 1$ **to** N **do**
6 Update buffer $\mathcal{B}$ with $\boldsymbol{\phi}_{1..\lambda}$ via reservoir sampling
7 **for** $i \leftarrow 1$ **to** λ **do**
8 $\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$
9 $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}(\boldsymbol{\theta}_i)$
10 $D_i \leftarrow \mathit{density}(\boldsymbol{\phi}_i)$
11 Add $(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i)$ to archive $\mathcal{A}$
12 Update $\mathit{density}$ with the new buffer $\mathcal{B}$
13 Rank $\boldsymbol{\theta}_i$ in *ascending* order by D_i
14 Adapt CMA-ES parameters $\boldsymbol{\mu}, \Sigma, \boldsymbol{p}$ based on density ranking D_i

5 CONNECTION BETWEEN NOVELTY SEARCH AND KERNEL DENSITY ESTIMATES

We provide theoretical insight into the connection between NS and DDS-KDE and delineate the advantage of KDE over novelty score. KDE and novelty score are non-stationary since they change as more solutions are discovered by their respective optimizers. However, the magnitude of non-stationarity is potentially different in KDE and novelty score (Theorem 5.2, 5.3). Furthermore, when $k \geq |\mathcal{B}|$ in the k-nearest neighbor calculation for novelty score, novelty search becomes a special case of DDS-KDE (Theorem 5.4). Finally, we conjecture that any meaningful density estimator utilizing a point's distance to its k-nearest neighbors will incur similar weak stability guarantees as novelty score (Conjecture 5.5). We include all proofs in Appendix E.

Stability Under Non-stationarity. DO algorithms such as NS, DDS, and CMA-MAE gradually learn a non-stationary density representation (e.g., KDE or histogram) as they explore the feature space. However, drastic changes in the representation present significant challenges for adaptive optimizers, such as Adam [26] and CMA-ES. If the density representation changes drastically every iteration, it would be impossible for adaptive optimizers to properly adapt its parameters to optimize the density function.

To characterize the extent of change in the density estimate, we appeal to the notion of *uniform stability* [24], defined as follows:

Definition 5.1. *Let a function $F(\boldsymbol{x};\mathcal{B}) : \mathbb{R}^d \to \mathbb{R}$ be parameterized by a set $\mathcal{B} \subseteq \mathbb{R}^d$. We say that F is ϵ-uniformly stable if for all $\mathcal{B}, \mathcal{B}' \subseteq \mathbb{R}^d$, where $\mathcal{B}$ and $\mathcal{B}'$ differ by at most one element, we have*

$$\sup_{\boldsymbol{x}\in\mathbb{R}^d} \left|F(\boldsymbol{x};\mathcal{B}) - F(\boldsymbol{x};\mathcal{B}')\right| \leq \epsilon \tag{3}$$

We note that uniform stability has close connections to influence functions in statistics, which measure how much an estimator deviates from the ground truth when given a subset of the data [7]. In this interpretation, uniform stability is an upper-bound of the (empirical) influence function.

We prove that, for KDE, the higher the size of its feature buffer and bandwidth, the stronger the uniform stability guarantee.

Theorem 5.2. *A kernel density estimate $\hat{D}_h(\boldsymbol{x};\mathcal{B})$ managed with reservoir sampling, such that features in the buffer are exchanged one at a time, is $\frac{1}{|\mathcal{B}|h}$-uniformly stable, where h is the bandwidth.*

Higher bandwidth makes the function more stationary and thus more suited for adaptive optimizers. However, higher bandwidth also leads to *over-smoothing* of the density estimate, which conceals modes of the true density function [4] (Fig. 2a). Thus, selecting an optimal bandwidth for DDS requires a fine balance between the accuracy and uniform stability of the KDE.

In contrast, novelty score becomes less uniformly stable as the diameter of the feature space W increases:

Theorem 5.3. *Novelty score $\rho(\boldsymbol{x};\mathcal{B})$ is $\frac{W}{k}$-uniformly stable, where k is the nearest neighbors parameter in novelty score and $W = \max_{\boldsymbol{s}_1,\boldsymbol{s}_2\in S}\|\boldsymbol{s}_1 - \boldsymbol{s}_2\|$ is the diameter of the feature space S.*

Therefore, for unbounded feature spaces, the uniform stability of novelty score is also unbounded, and for bounded feature spaces, novelty score has stronger uniform stability guarantees when k is larger.

When the feature space is bounded, as in our experiments, KDE has a stronger stability guarantee than novelty score for bandwidth $h \geq \frac{k}{|\mathcal{B}|}$. Following this insight, we select a bandwidth satisfying this inequality with the empirical experiments in Appendix B. Our theoretical results are corroborated by our experiments in Sec. 6, which demonstrate that DDS-KDE outperforms NS in all domains on all metrics.

Equivalence of NS and DDS-KDE. We observe that, when $h = 1$ and as $k \to |\mathcal{B}|$, the uniform stability upper-bound of novelty score approaches that of KDE. We prove in Theorem 5.4 that when all points are considered in the novelty score computation (i.e., let $k = \infty$), NS is a special case of DDS-KDE under ranking-based optimizers such as CMA-ES. Specifically, we demonstrate that under these conditions, the ranking of solutions based on their novelty score is identical to the ranking based on their kernel density estimate.

Theorem 5.4. *Let $\mathcal{B} \subseteq \mathbb{R}^m$ be a set of features. Consider the rankings π_{NS} and π_{KDE} on another set of features $\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m\} \subseteq \mathbb{R}^m$ where $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\boldsymbol{\theta}_i)$. We define the rankings as follows: $\pi_{\text{NS}}(i) \geq \pi_{\text{NS}}(j) \iff \rho(\boldsymbol{\phi}_i;\mathcal{B}) \geq \rho(\boldsymbol{\phi}_j;\mathcal{X})$ and $\pi_{\text{KDE}}(i) \geq \pi_{\text{KDE}}(j) \iff \hat{D}(\boldsymbol{\phi}_i;\mathcal{B}) \leq \hat{D}(\boldsymbol{\phi}_j;\mathcal{B})$. We show that $\pi_{NS} = \pi_{KDE}$, when NS has $k = \infty$ (or, equivalently, $k = |\mathcal{B}|$) and KDE has triangular kernel $K(u) = 1 - |u|$ with support over the entire feature space S.*

Stability of k-NN. We conjecture that only considering k-nearest neighbors in the density representation, for $k < |\mathcal{B}|$, will inevitably result in poor uniform stability bounds. Notice that the maximum distance between any two features is the diameter of the feature space, W. Thus, when a new feature replaces a former k-nearest neighbors of $\boldsymbol{x}$, the total distance between a feature $\boldsymbol{x}$ to its k-nearest neighbors can change by W in the worst case.

Conjecture 5.5. *Let $\mathcal{X} \subseteq \mathbb{R}^m$ and $D : \mathbb{R}^m \to \mathbb{R}$ be a density estimator. If $D(\boldsymbol{x};\mathcal{X})$ for $\boldsymbol{x} \in \mathbb{R}^m$ is computed based on the distance to its k-nearest neighbors of $\boldsymbol{x}$, then $D(\boldsymbol{x};\mathcal{X})$ is $O(W)$-uniformly stable for $k < |\mathcal{B}|$.*

A proof of this conjecture would imply that density estimators based on k-nearest neighbor metric have unbounded uniform stability in unbounded feature spaces, for $k < |\mathcal{B}|$. Thus, the potential for rapid changes in the optimization landscape makes such estimators incompatible with adaptive optimizers.

6 EXPERIMENTS

We compare the performance of DDS-KDE and DDS-CNF with the QD algorithms MAP-Elites (line), CMA-ME, and CMA-MAE, and with NS using CMA-ES as the underlying optimizer. MAP-Elites (line) refers to the implementation of MAP-Elites with the Iso+LineDD operator [47]. All algorithms are implemented with pyribs [44].

Our experiments include canonical benchmark domains from QD and DO: Linear Projection [14], Arm Repertoire [47], and Deceptive Maze [29]. As discussed in Sec. 2, we convert QD domains into DO domains by setting the objective to be constant, effectively removing the importance of solution quality. For Deceptive Maze, we use the implementation in Kheperax [21]. Furthermore, to address the challenges posed by high-dimensional feature spaces, we introduce a new domain, Multi-feature Linear Projection, that generalizes Linear Projection to high-dimensional feature spaces.

All experiments run on a 128-core workstation with an NVIDIA RTX A6000 GPU. While all algorithms are single-threaded, we use the GPU for training the CNF in DDS-CNF and for evaluating the Deceptive Maze domain.

6.1 Experimental Design

Independent Variable. In each domain, we conduct a between-groups study with the algorithm as the independent variable.

Dependent Variables. We compute the archive coverage as the number of occupied cells in the archive divided by the total number of cells. To compare the coverage of tessellation-based algorithms (CMA-MAE, CMA-ME, and MAP-Elites) with that of non-tessellation-based algorithms (DDS-KDE, DDS-CNF, and NS), we track the coverage of all algorithms on a passive archive $\mathcal{A}$ tessellated by a 100×100 grid. For the high-dimensional feature space experiment, we track coverage with a centroidal Voronoi tessellation with 10,000 cells [46].

Following prior work [18], we also assess the ability of each algorithm to uniformly explore the feature space. Thus, we measure the cross-entropy between a uniform distribution and the distribution of sampled features. Let N_e be the total times throughout the entire search that solutions were discovered in cell $e \in \{1, \ldots, l\}$ in the

passive archive, and $N_{\text{total}} = \sum_{e=1}^{l} N_e$. The cross-entropy score is defined as:

$$\text{CE} = -\sum_{e=1}^{l} \frac{1}{l} \log\left(\frac{N_e}{N_{\text{total}}}\right) \tag{4}$$

CE achieves its minimum value when N_e is uniformly distributed across all cells $e \in \{1, \ldots, l\}$. For all passive archives in our experiments, this minimum is 9.21 for $l = 10{,}000$ cells.

6.2 Domain Details

Linear Projection (LP). LP is a QD domain where $\boldsymbol{\phi}$ induces high distortion by mapping an n-dimensional solution space to a 2D feature space [14]. A harsh penalty is applied outside the bounds of the feature space to hinder exploration near the bounds. The feature function $\boldsymbol{\phi} : \mathbb{R}^n \to \left[-5.12 \cdot \frac{n}{2}, 5.12 \cdot \frac{n}{2}\right]^2$ is defined as:

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = \left(\sum_{i=1}^{\frac{n}{2}} \text{clip}(\theta_i), \sum_{i=\frac{n}{2}+1}^{n} \text{clip}(\theta_i)\right) \tag{5}$$

$$\text{clip}(\theta_i) = \begin{cases} \theta_i & \text{if } |\theta_i| \leq 5.12 \\ 5.12/\theta_i & \text{otherwise} \end{cases} \tag{6}$$

where θ_i is the ith component of $\boldsymbol{\theta}$, and we assume that n is divisible by 2. $\boldsymbol{\phi}(\boldsymbol{\theta})$ applies $\text{clip}(\cdot)$ to each θ_i and sums the two halves of $\boldsymbol{\theta}$. Since $\text{clip}(\theta_i)$ restricts θ_i to the interval $[-5.12, 5.12]$, $\boldsymbol{\phi}(\boldsymbol{\theta})$ is bounded by the closed interval $\left[-5.12 \cdot \frac{n}{2}, 5.12 \cdot \frac{n}{2}\right]^2$.

For DO experiments, we set the objective function $f(\boldsymbol{\theta}) = 1$. Following prior work [11, 12], we let the solution dimension be $n = 100$.

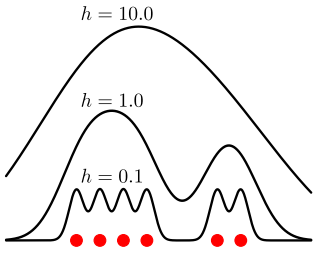
Arm Repertoire. The goal of Arm Repertoire [10, 47] is to search for a diverse collection of arm positions for a planar robotic arm with n revolving joints. In this domain, $\boldsymbol{\theta} \in [-\pi, \pi]^n$ represents the angles of the n joints, and $\boldsymbol{\phi}(\boldsymbol{\theta})$ computes the (x, y) position of the arm's end-effector using forward kinematics. While all other domains in this paper have a maximum of 100% coverage, Arm Repertoire has a maximum coverage of 80.24% when using a 100×100 grid archive [12], since the arm can only move in a circle of radius n. Similar to LP, we set $f(\boldsymbol{\theta}) = 1$ and $n = 100$.

Deceptive Maze. Deceptive Maze is a DO domain that challenges the algorithm to discover a diverse set of final positions for robots navigating a maze (Fig. 2b) [29]. In this domain, $\boldsymbol{\theta}$ parameterizes the robot's neural network controller. $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the final position of the robot after evaluating its path in the maze. As a DO domain, this has no objective function. In our experiments, the neural network is a MLP with $n = 66$ parameters.

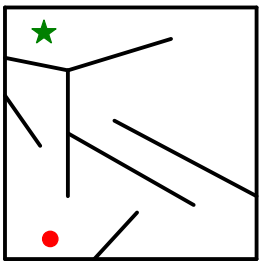
Multi-feature Linear Projection (Multi-feature LP). We introduce a generalized version of LP that scales to m-dimensional feature spaces. Assuming that n is divisible by m, let $r = \frac{n}{m}$. The feature function $\boldsymbol{\phi} : \mathbb{R}^n \to [-5.12 \cdot r, 5.12 \cdot r]^m$ is defined as

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = \left(\sum_{i=jr+1}^{(j+1)r} \text{clip}(\theta_i) : j \in \{0, \ldots, m-1\}\right) \tag{7}$$

When $m = 2$, this is equivalent to LP definition in Eq. 5. Our experiments use $n = 100$ and $m = 10$.



(a) The effect of the bandwidth h on a one-dimensional KDE. Red dots represent the data, and black lines depict the KDE. When h is too small, the KDE reveals many misleading local maxima. When h is too large, the KDE conceals modes from the underlying distribution.



(b) The layout in the deceptive maze domain. The red dot indicates the start, and the green dot indicates the goal. While there is a goal, DO seeks to find agents that reach every position in the maze. Adapted from prior work [29].

Figure 2

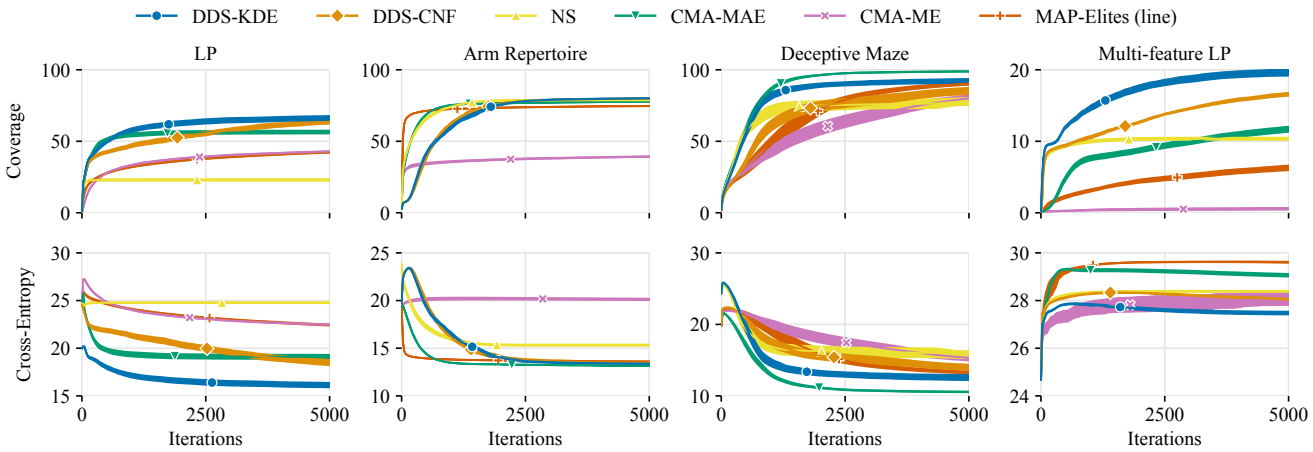
6.3 Analysis

Fig. 3 shows the mean coverage and cross-entropy over 10 trials. In each domain, we conducted a one-way ANOVA to check if the algorithms differed in their coverage and cross-entropy. Since all ANOVAs were significant ($p < 0.001$), we followed up with pairwise comparisons via Tukey's HSD test. We include full statistical analysis in Appendix F.

Coverage. DDS-KDE and DDS-CNF outperform all baselines on LP, Arm Repertoire, and Multi-feature LP. They exhibit no statistical difference in performance with the best-performing algorithm on Deceptive Maze (CMA-MAE). Notably, DDS-KDE solves Arm Repertoire nearly perfectly, as the maximum coverage in the domain is 80.24%.

We attribute the high coverage of DDS-KDE and DDS-CNF on these domains to the continuity of their density estimate, which prevents DDS-KDE from converging prematurely. For example, on LP, our algorithms discover more solutions near the edges of the feature space than CMA-MAE (Appendix G). The passive archive maintained by CMA-MAE converges as all the solutions fall into previously discovered cells [14]. In contrast, the continuity of our density estimate and the online buffer updates facilitate DDS algorithms to achieve slow but continual progress in exploring the feature space (Fig. 3, LP and Arm Repertoire). This is because the shape of the continuous density estimate always changes slightly after each iteration of the algorithm.

For multi-feature LP, we observe that algorithms leveraging continuous representations of the feature space, i.e., DDS and NS, explore the feature space much faster than other algorithms driven by discrete feature space representations, e.g., CMA-MAE (Fig. 3). This is because CMA-MAE is optimizing on a centroidal Voronoi tessellation with 10,000 cells, where each cell has significantly more volume compared to that of a 100×100 grid due to the increased dimensionality of the feature space. Hence, more solutions map to the same cells, making it more difficult to find solutions outside of previously explored cells.



	LP		Arm Repertoire		Deceptive Maze		Multi-feature LP	
	Coverage	Cross-Entropy	Coverage	Cross-Entropy	Coverage	Cross-Entropy	Coverage	Cross-Entropy
DDS-KDE	**67.67 ±2.13%**	**17.57 ±0.41**	**80.22 ±0.01%**	14.14 ±0.02	94.93 ±0.84%	11.84 ±0.21	**50.22 ±0.45%**	**22.20 ±0.09**
DDS-CNF	63.65 ±1.38%	18.50 ±0.29	79.82 ±0.19%	13.48 ±0.12	85.17 ±2.28%	14.01 ±0.38	16.57 ±0.22%	28.07 ±0.03
NS	22.96 ±1.59%	24.78 ±0.06	78.67 ±0.23%	15.32 ±0.07	77.70 ±2.47%	15.78 ±0.50	10.36 ±0.16%	28.39 ±0.03
CMA-MAE	56.47 ±1.06%	19.09 ±0.23	77.77 ±0.10%	**13.14 ±0.01**	**98.83 ±0.18%**	**10.55 ±0.06**	11.68 ±0.37%	29.06 ±0.06
CMA-ME	42.90 ±0.25%	22.45 ±0.05	39.29 ±0.24%	20.13 ±0.09	79.97 ±2.17%	15.33 ±0.39	0.56 ±0.14%	28.10 ±0.28
MAP-Elites (line)	42.30 ±0.31%	22.43 ±0.07	74.62 ±0.07%	13.60 ±0.01	90.32 ±0.87%	13.28 ±0.16	6.28 ±0.34%	29.61 ±0.02

Figure 3: Coverage and cross-entropy (CE) after 5,000 iterations of each algorithm in all domains. We report the mean over 10 trials, with error bars showing the standard error of the mean. Higher coverage and lower cross-entropy are better.

A similar phenomenon was observed in prior work when increasing the dimensionality of the solution space in the LP domain [12]. Increased solution dimensionality more heavily distorts the feature mapping and, similarly, causes most solutions to fall in the same cells of the feature space. Consequently, the LP domain becomes exceptionally challenging for QD algorithms working with a tessellation (like CMA-MAE), as there is insufficient signal to drive the algorithm towards the boundaries of the feature space.

DDS-KDE circumvents this limitation of tessellations with its continuous density estimation of the feature space. While in tesselation-based algorithms, solutions will fall into the same cell, DDS-KDE will retain the solutions in its buffer, which generates signal to drive the search towards the feature space boundaries. Thus, DDS-KDE is more resilient to the distortions of the feature mapping and can better scale with the dimensionality of the feature space.

Cross-Entropy. DDS-KDE and DDS-CNF outperform[1] all baselines in LP, with the exception of DDS-CNF, whose performance is not significantly different from CMA-MAE. On Arm Repertoire and Deceptive Maze, DDS-KDE and DDS-CNF outperform NS. However, while DDS-KDE is on par with CMA-MAE on Arm Repertoire , CMA-MAE outperforms DDS-KDE on Deceptive Maze and DDS-CNF on both Arm Repertoire and Deceptive Maze. Finally, DDS-KDE outperforms all algorithms on Multi-feature LP, while DDS-CNF outperforms CMA-MAE and MAP-Elites (line).

We attribute the improved performance of CMA-MAE to the nature of the cross-entropy metric. Cross-entropy is intended to approximate the distributional similarity between the exploration of the feature space and the uniform distribution. CMA-MAE directly optimizes a passive archive with uniform tessellation, unlike DDS and NS, which are agnostic to the passive archive. This experimental setup naturally favors CMA-MAE with respect to the cross-entropy metric.

DDS-KDE performs as well as DDS-CNF in terms of coverage and cross-entropy across most domains. However, on Multi-feature LP, DDS-KDE outperforms DDS-CNF on both metrics; on LP and Deceptive Maze, DDS-KDE outperforms DDS-CNF in terms of cross-entropy.

We attribute the difference between the performance of DDS-KDE and DDS-CNF to the bandwidth parameter in KDE, which allows us to adjust the smoothness of the KDE (Fig. 2a). On the other hand, CNF lacks explicit control over its smoothness. Thus, while we adjust the smoothness of the KDE to improve performance for DDS-KDE (Sec. 5), we can not apply the same techniques to boost the performance of DDS-CNF.

7 LIMITATIONS AND FUTURE WORK

We introduce a new diversity optimization algorithm with two variants, DDS-KDE and DDS-CNF. We provide theoretical insight into the connection between DDS-KDE and novelty search and show that both DDS algorithms excel at discovering diverse solutions.

We envision several extensions to DDS. While we proposed two DDS variants, DDS is a general method that can integrate with a wide variety of distribution fitting techniques. DDS can be combined with parametric estimators like mixture models [32] and moment matching [22], as well as with non-parametric estimators like vector quantization [27] and generative models [19]. In general,

[1]Recall that lower CE is better as it indicates a more uniform exploration of the feature space (Sec. 6.1)

we believe that DDS will interface with any method that learns a continuous density representation of the feature space.

For DDS-KDE, while we assumed a constant bandwidth, prior work has adapted the bandwidth online [28] and varied the bandwidth based on query location [43]. These methods could be applied to secure tighter uniform stability bounds for the KDE, making feature space exploration more efficient.

A key limitation of our algorithms is their significant computational cost. Querying the KDE incurs a runtime of $O(|\mathcal{B}|)$, where $|\mathcal{B}|$ is the buffer size, and fine-tuning the CNF is computationally expensive due to backpropagation, even when accelerated on a GPU. These issues could be mitigated by accelerating KDE computation [34, 49] and improving CNF training efficiency [25, 35].

We proposed a generalization of the standard linear projection domain to higher dimensional feature spaces, and demonstrate that DDS exhibits superior performance to existing algorithms. Our results could be further strengthened by evaluating DDS's performance on more complex domains. For example, we could evaluate the performance of DDS on locomotion tasks, where the solutions are neural network controllers and the task is to control a multi-pedal walker to move efficiently [39, 41].

Finally, we are excited to see how the underlying insight of DDS — leveraging continuous density representations of the feature space — can improve the exploration power of QD algorithms, especially in high-dimensional feature spaces. Applying DDS to QD will improve its performance on applications such as scenario generation for complex systems [13], generative design [15], damage recovery in robotics [9], reinforcement learning [2], and procedural content generation [20].

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was supported by the NSF NRI (#2024949), NSF GRFP (#DGE-1842487), NSF CAREER (#2145077), and the NVIDIA Academic Hardware Grant.

REFERENCES

[1] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. 2010. Bidirectional relation between CMA evolution strategies and natural evolution strategies. In *International Conference on Parallel Problem Solving from Nature*. Springer, 154–163.

[2] Sumeet Batra, Bryon Tjanaka, Matthew C Fontaine, Aleksei Petrenko, Stefanos Nikolaidis, and Gaurav Sukhatme. 2023. Proximal Policy Gradient Arborescence for Quality Diversity Reinforcement Learning. *arXiv preprint arXiv:2305.13795* (2023).

[3] Adrian W. Bowman. 1984. An Alternative Method of Cross-Validation for the Smoothing of Density Estimates. *Biometrika* 71, 2 (1984), 353–360. http://www.jstor.org/stable/2336252

[4] Yen-Chi Chen. 2017. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology* 1, 1 (2017), 161–187.

[5] Alexandre Chenu, Nicolas Perrin-Gilbert, Stephane Doncieux, and Olivier Sigaud. 2021. Selection-Expansion: A Unifying Framework for Motion-Planning and Diversity Search Algorithms. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part IV 30*. Springer, 568–579.

[6] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. 2018. Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/b1301141feffabac455e1f90a7de2054-Paper.pdf

[7] R Dennis Cook and Sanford Weisberg. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics* 22, 4 (1980), 495–508.

[8] Antoine Cully. 2019. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 81–89.

[9] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

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

[11] Matthew Fontaine and Stefanos Nikolaidis. 2021. Differentiable Quality Diversity. *Advances in Neural Information Processing Systems* 34 (2021).

[12] Matthew Fontaine and Stefanos Nikolaidis. 2023. Covariance matrix adaptation map-annealing. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 456–465.

[13] Matthew C Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. 2021. On the importance of environments in human-robot coordination. *Proceedings of Robotics: Science and Systems 17* (2021).

[14] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. 2020. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*. 94–102.

[15] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2018. Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation* 26, 3 (2018), 381–410.

[16] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2019. Are Quality Diversity Algorithms Better at Generating Stepping Stones than Objective-Based Search?. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 115–116. https://doi.org/10.1145/3319619.3321897

[17] Tobias Glasmachers, Tom Schaul, Sun Yi, Daan Wierstra, and Jürgen Schmidhuber. 2010. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. 393–400.

[18] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. 2015. Devising Effective Novelty Search Algorithms: A Comprehensive Empirical Study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (Madrid, Spain) *(GECCO '15)*. Association for Computing Machinery, New York, NY, USA, 943–950. https://doi.org/10.1145/2739480.2754736

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[20] Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. 2019. Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*. IEEE, 1–8.

[21] Luca Grillotti and Antoine Cully. 2023. Kheperax: A Lightweight JAX-Based Robot Control Environment for Benchmarking Quality-Diversity Algorithms. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (Lisbon, Portugal) *(GECCO '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 2163–2165. https://doi.org/10.1145/3583133.3596387

[22] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. 2018. Local moment matching: A unified methodology for symmetric functional estimation and distribution estimation under wasserstein distance. In *Conference On Learning Theory*. PMLR, 3189–3221.

[23] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. *arXiv preprint arXiv:1604.00772* (2016).

[24] Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*. PMLR, 1225–1234.

[25] Han-Hsien Huang and Mi-Yen Yeh. 2021. Accelerating continuous normalizing flow with trajectory polynomial regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7832–7839.

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[27] Teuvo Kohonen. 1990. Improved versions of learning vector quantization. In *1990 ijcnn international joint conference on Neural networks*. IEEE, 545–550.

[28] Matej Kristan, Aleš Leonardis, and Danijel Skočaj. 2011. Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition* 44, 10 (2011), 2630–2642. https://doi.org/10.1016/j.patcog.2011.03.019 Semi-Supervised Learning for Visual Content Analysis and Understanding.

[29] Joel Lehman and Kenneth O Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19, 2 (2011), 189–223.

[30] Kim-Hung Li. 1994. Reservoir-Sampling Algorithms of Time Complexity O(n(1 + Log(N/n))). *ACM Trans. Math. Softw.* 20, 4 (dec 1994), 481–493. https://doi.org/10.1145/198429.198435

[31] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=PqvMRDCJT9t

[32] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. 2019. Finite mixture models. *Annual review of statistics and its application* 6 (2019), 355–378.

[33] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

[34] Travis A. O'Brien, Karthik Kashinath, Nicholas R. Cavanaugh, William D. Collins, and John P. O'Brien. 2016. A Fast and Objective Multidimensional Kernel Density Estimation Method: fastKDE. *Computational Statistics & Data Analysis* 101 (2016), 148–160. https://doi.org/10.1016/j.csda.2016.02.014

[35] Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. 2021. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9223–9232.

[36] Giuseppe Paolo, Alban Laflaquiere, Alexandre Coninx, and Stephane Doncieux. 2020. Unsupervised learning and exploration of reachable outcome space. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2379–2385.

[37] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI* 3 (2016), 40. https://doi.org/10.3389/frobt.2016.00040

[38] Justin K. Pugh, L. B. Soros, Paul A. Szerlip, and Kenneth O. Stanley. 2015. Confronting the Challenge of Quality Diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (Madrid, Spain) *(GECCO '15)*. Association for Computing Machinery, New York, NY, USA, 967–974. https://doi.org/10.1145/2739480.2754664

[39] Nemanja Rakicevic, Antoine Cully, and Petar Kormushev. 2021. Policy manifold search: Exploring the manifold hypothesis for diversity-based neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 901–909.

[40] Mats Rudemo. 1982. Empirical Choice of Histograms and Kernel Density Estimators. *Scandinavian Journal of Statistics* 9, 2 (1982), 65–78. http://www.jstor.org/stable/4615859

[41] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJgLZR4KvH

[42] Bernard W Silverman. 1986. *Density estimation for statistics and data analysis*. Vol. 26. CRC press.

[43] George R. Terrell and David W. Scott. 1992. Variable Kernel Density Estimation. *The Annals of Statistics* 20, 3 (1992), 1236 – 1265. https://doi.org/10.1214/aos/1176348768

[44] Bryon Tjanaka, Matthew C Fontaine, David H Lee, Yulun Zhang, Nivedit Reddy Balam, Nathaniel Dennler, Sujay S Garlanka, Nikitas Dimitri Klapsis, and Stefanos Nikolaidis. 2023. Pyribs: A Bare-Bones Python Library for Quality Diversity Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '23)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3583131.3590374

[45] Bryon Tjanaka, Matthew C Fontaine, Julian Togelius, and Stefanos Nikolaidis. 2022. Approximating gradients for differentiable quality diversity in reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1102–1111.

[46] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2018. Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm. *IEEE Transactions on Evolutionary Computation* 22, 4 (2018), 623–630. https://doi.org/10.1109/TEVC.2017.2735550

[47] Vassilis Vassiliades and Jean-Baptiste Mouret. 2018. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 149–156.

[48] Stanislaw Weglarczyk. 2018. Kernel density estimation and its application. *ITM Web Conf.* 23 (2018), 00037. https://doi.org/10.1051/itmconf/20182300037

[49] Yang, Duraiswami, and Gumerov. 2003. Improved fast gauss transform and efficient kernel density estimation. In *Proceedings ninth IEEE international conference on computer vision*. IEEE, 664–671.