Abstract

Quality diversity (QD) is a growing branch of stochastic optimization research that studies the problem of generating an archive of solutions that maximize a given objective function but are also diverse with respect to a set of specified measure functions. However, even when these functions are differentiable, QD algorithms treat them as “black boxes”, ignoring gradient information. We present the differentiable quality diversity (DQD) problem, a special case of QD, where both the objective and measure functions are first order differentiable. We then present MAP-Elites via a Gradient Arborescence (MEGA), a DQD algorithm that leverages gradient information to efficiently explore the joint range of the objective and measure functions. Results in two QD benchmark domains and in searching the latent space of a StyleGAN show that MEGA significantly outperforms state-of-the-art QD algorithms, highlighting DQD’s promise for efficient quality diversity optimization when gradient information is available. Source code is available at https://github.com/icaros-usc/dqd.

1 Introduction

We introduce the problem of differentiable quality diversity (DQD) and propose the MAP-Elites via a Gradient Arborescence (MEGA) algorithm as the first DQD algorithm.

Unlike single-objective optimization, quality diversity (QD) is the problem of finding a range of high quality solutions that are diverse with respect to prespecified metrics. For example, consider the problem of generating realistic images that match as closely as possible a target text prompt “Elon Musk”, but vary with respect to hair and eye color. We can formulate the problem of searching the latent space of a generative adversarial network (GAN) [26] as a QD problem of discovering latent codes that generate images maximizing a matching score for the prompt “Elon Musk”, while achieving a diverse range of measures of hair and eye color, assessed by visual classification models [51]. More generally, the QD objective is to maximize an objective $f$ for each output combination of measure functions $m_i$. A QD algorithm produces an archive of solutions, where the algorithm attempts to discover a representative for each measure output combination, whose $f$ value is as large as possible.

While our example problem can be formulated as a QD problem, all current QD algorithms treat the objective $f$ and measure functions $m_i$ as a black box. This means, in our example problem, current QD algorithms fail to take advantage of the fact that both $f$ and $m_i$ are end-to-end differentiable neural networks. Our proposed differentiable quality diversity (DQD) algorithms leverage first-order derivative information to significantly improve the computational efficiency of solving a variety of QD problems where $f$ and $m_i$ are differentiable.

To solve DQD, we introduce the concept of a gradient arborescence. Like gradient ascent, a gradient arborescence makes greedy ascending steps based on the objective $f$. Unlike gradient ascent, a gradient arborescence encourages exploration by branching via the measures $m_i$. We adopt the term...
Adapt via CMA-ES
Sample Branch via
objective-measure
gradients
Update
archive
Ascend to
maximize

Figure 1: An overview of the Covariance Matrix Adaptation MAP-Elites via a Gradient Arborescence (CMA-MEGA) algorithm. The algorithm leverages a gradient arborescence to branch in objective-measure space, while dynamically adapting the gradient steps to maximize a QD objective (Eq. 1).

Our work makes four main contributions. 1) We introduce and formalize the problem of differentiable quality diversity (DQD). 2) We propose two DQD algorithms: Objective and Measure Gradient MAP-Elites via a Gradient Arborescence (OMG-MEGA), an algorithm based on MAP-Elites [13], which branches based on the measures \( m_i \) but ascends based on the objective function \( f \); and Covariance Matrix Adaptation MEGA (CMA-MEGA) which is based on the CMA-ME [19] algorithm, and which branches based on the objective-measure space but ascends based on maximizing the QD objective (Fig. 1). Both algorithms search directly in measure space and leverage the gradients of \( f \) and \( m_i \) to form efficient parameter space steps in \( \theta \). 3) We show in three different QD domains (the linear projection, the arm repertoire, and the latent space illumination (LSI) domains), that DQD algorithms significantly outperform state-of-the-art QD algorithms that treat the objective and measure functions as a black box. 4) We demonstrate how searching the latent space of a StyleGAN [36] in the LSI domain with CMA-MEGA results in a diverse range of high-quality images.

2 Problem Definition

Quality Diversity. The quality diversity (QD) problem assumes an objective \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) in an \( n \)-dimensional continuous space \( \mathbb{R}^n \) and \( k \) measures \( m_i : \mathbb{R}^n \rightarrow \mathbb{R} \) or, as a joint measure, \( m : \mathbb{R}^n \rightarrow \mathbb{R}^k \). Let \( S = m(\mathbb{R}^n) \) be the measure space formed by the range of \( m \). For each \( s \in S \) the QD objective is to find a solution \( \theta \in \mathbb{R}^n \) such that \( m(\theta) = s \) and \( f(\theta) \) is maximized.

However, we note that \( \mathbb{R}^k \) is continuous, and an algorithm solving the quality diversity problem would require infinite memory to store all solutions. Thus, QD algorithms in the MAP-Elites family approximate the problem by discretizing \( S \) via a tessellation method. Let \( T \) be the tessellation of \( S \) into \( M \) cells. We relax the QD objective to find a set of solutions \( \theta_i, i \in \{1, \ldots, M\} \), such that each \( \theta_i \) occupies one unique cell in \( T \). The occupants \( \theta_i \) of all \( M \) cells form an archive of solutions. Each solution \( \theta_i \) has a position in the archive \( m(\theta_i) \), corresponding to one out of \( M \) cells, and an objective value \( f(\theta_i) \).

The objective of QD can be rewritten as follows, where the goal is to maximize the objective value for each cell in the archive:

\[
\max_{i=1}^{M} f(\theta_i) \tag{1}
\]

Differentiable Quality Diversity. We define the differentiable quality diversity (DQD) problem, as a QD problem where both the objective \( f \) and measures \( m_i \) are first-order differentiable.

3 Preliminaries

We present several state-of-the-art derivative-free QD algorithms. Our proposed DQD algorithm MEGA builds upon ideas from these works, while introducing measure and objective gradients into the optimization process.
MAP-Elites and MAP-Elites (line). MAP-Elites [13, 43] first tessellates the measure space $\mathcal{S}$ into evenly-spaced grid cells. The upper and lower bounds for $m$ are given as input to constrain $\mathcal{S}$ to a finite region. MAP-Elites first samples solutions from a fixed distribution $\theta \sim \mathcal{N}(0, I)$, and populates an initial archive after computing $f(\theta)$ and $m(\theta)$. Each iteration of MAP-Elites selects $\lambda$ cells uniformly at random from the archive and perturbs each occupant $\theta_i$ with fixed-variance $\sigma_i$ isotropic Gaussian noise: $\theta' = \theta_i + \sigma_i \mathcal{N}(0, I)$. Each new candidate solution $\theta'$ is then evaluated and added to the archive if $\theta'$ discovers a new cell or improves an existing cell. The algorithm continues to generate solutions for a specified number of iterations.

Later work introduced the Iso+LineDD operator [61]. The Iso+LineDD operator samples two archive solutions $\theta_i$ and $\theta_j$, then blends a Gaussian perturbation with a noisy interpolation given hyperparameters $\sigma_1$ and $\sigma_2$: $\theta' = \theta_i + \sigma_1 \mathcal{N}(0, I) + \sigma_2 \mathcal{N}(0, 1)(\theta_i - \theta_j)$. In this paper we refer to MAP-Elites with an Iso+LineDD operator as MAP-Elites (line).

CMA-ME. Covariance Matrix Adaptation MAP-Elites (CMA-ME) [19] combines the archiving mechanisms of MAP-Elites with the adaptation mechanisms of CMA-ES [30]. While MAP-Elites creates new solutions by perturbing existing solutions with fixed-variance Gaussian noise, CMA-ME maintains a full-rank Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ in parameter space $\mathbb{R}^n$. Each iteration of CMA-ME samples $\lambda$ candidate solutions $\theta_i \sim \mathcal{N}(\mu, \Sigma)$, evaluates each solution, and updates the archive based on the following rule: if there is a previous occupant $\theta_p$ at the same cell, we compute $\Delta_i = f(\theta_i) - f(\theta_p)$, otherwise if the cell is empty we compute $\Delta_i = f(\theta_i)$. We then rank the sampled solutions by increasing improvement $\Delta_i$, with an extra criteria that candidates discovering new cells are ranked higher than candidates that improve existing cells. We then update $\mathcal{N}(\mu, \Sigma)$ with the standard CMA-ES update rules based on the improvement ranking. CMA-ME restarts when all $\lambda$ solutions fail to change the archive. On a restart we reset the Gaussian $\mathcal{N}(\theta_i, I)$, where $\theta_i$ is an archive solution chosen uniformly at random, and all internal CMA-ES parameters. In the supplemental material, we derive, for the first time, a natural gradient interpretation of CMA-ME’s improvement ranking mechanism, based on previous theoretical work on CMA-ES [1].

4 Algorithms

We present two variants of our proposed MEGA algorithm: OMG-MEGA and CMA-MEGA. We form each variant by adapting the concept of a gradient arborescence to the MAP-Elites and CMA-ME algorithms, respectively. Finally, we introduce two additional baseline algorithms, OG-MAP-Elites and OG-MAP-Elites (line), which operate only on the gradients of the objective.

OMG-MEGA. We first derive the Objective and Measure Gradient MAP-Elites via Gradient Arborescence (OMG-MEGA) algorithm from MAP-Elites.

First, we observe how gradient information could benefit a QD algorithm. Note that the QD objective is to explore the measure space, while maximizing the objective function $f$. We observe that maximizing a linear combination of measures: $\sum_{j=1}^{k} c_j m_j(\theta)$, where $c$ is a $k$-dimensional vector of coefficients, enables movement in a $k$-dimensional measure space. Adding the objective function $f$ to the linear sum enables movement in an objective-measure space. Maximizing $g$ with a positive coefficient of $f$ results in steps that increasing $f$, while the direction of movement in the measure space is determined by the sign and magnitude of the coefficients $c_j$.

$$g(\theta) = |c_0| f(\theta) + \sum_{j=1}^{k} c_j m_j(\theta)$$ (2)

We can then derive a direction function that perturbs a given solution $\theta$ based on the gradient of our linear combination $g$: $\nabla g(\theta) = |c_0| \nabla f(\theta) + \sum_{j=1}^{k} c_j \nabla m_j(\theta)$. We incorporate the direction function $\nabla g$ to derive a gradient-based MAP-Elites variation operator.
We observe that MAP-Elites samples a cell \( \theta_i \) and perturbs the occupant with Gaussian noise: \( \theta' = \theta_i + \sigma N(0, I) \). Instead, we sample coefficients \( c \sim N(0, \sigma I) \) and step:

\[
\theta' = \theta_i + | c_0 | \nabla f(\theta_i) + \sum_{j=1}^{k} c_j \nabla m_j(\theta_i)
\]  

(3)

The value \( \sigma \) acts as a learning rate for the gradient step, because it controls the scale of the coefficients \( c \sim N(0, \sigma I) \). To balance the contribution of each function, we normalize all gradients. In the supplemental material, we further justify gradient normalization and provide an empirical ablation study. Other than our new gradient-based operator, OMG-MEGA is identical to MAP-Elites.

CMA-MEGA. Next, we derive the Covariance Matrix Adaptation MAP-Elites via a Gradient Arborecence (CMA-MEGA) algorithm from CMA-ME. Fig. 1 shows an overview of the algorithm.

First, we note that we sample \( c \) in OMG-MEGA from a fixed-variance Gaussian. However, it would be beneficial to select \( c \) based on how \( c \), and the subsequent gradient step on \( \theta \), improve the QD objective defined in equation 1.

We frame the selection of \( c \) as an optimization problem with the objective of maximizing the QD objective (Eq. 1). We model a distribution of coefficients \( c \sim N(\mu, \Sigma) \). Given a \( \theta \), we can sample \( c \sim N(\mu, \Sigma) \), compute \( \theta' \) via Eq. 3 and adapt \( N(\mu, \Sigma) \) towards the direction of maximum increase of the QD objective (see Eq. 1).

We follow an evolution strategy approach to model and dynamically adapt the sampling distribution of coefficients \( N(\mu, \Sigma) \). We sample a population of \( \lambda \) coefficients from \( c_i \sim N(\mu, \Sigma) \) and generate \( \lambda \) solutions \( \theta_i \). We then compute \( \Delta_i \) from CMA-ME’s improvement ranking for each candidate solution \( \theta_i \). By updating \( N(\mu, \Sigma) \) with CMA-ES update rules for the ranking \( \Delta_i \), we dynamically adapt the distribution of coefficients \( c \) to maximize the QD objective.

Algorithm 1 shows the pseudocode for CMA-MEGA. In line 3 we evaluate the current solution and compute an objective value \( f \), a vector of measure values \( m \), and gradient vectors. As we dynamically adapt the coefficients \( c \), we normalize the objective and measure gradients (line 4) for stability. Because the measure space is tessellated, the measures \( m \) place solution \( \theta \) into one of the \( M \) cells in the archive. We then add the solution to the archive (line 5), if the solution discovers an empty cell in the archive, or if it improves an existing cell, identically to MAP-Elites.

We then use the gradient information to compute a step that maximizes improvement of the archive. In lines 6-12, we sample a population of \( \lambda \) coefficients from a multi-variate Gaussian retained by CMA-ES, and take a gradient step for each sample. We evaluate each sampled solution \( \theta'_i \), and compute the improvement \( \Delta_i \) (line 11). As in CMA-ME, we specify \( \Delta_i \) as the difference in the objective value between the sampled solution \( \theta_i \) and the existing solution, if one exists, or as the absolute objective value of the sampled solution if \( \theta_i \) belongs to an empty cell.

In line 13, we rank the sampled gradients \( \nabla \theta_i \) based on their respective improvements. As in CMA-ME, we prioritize exploration of the archive by ranking first by their objective values all samples that discover new cells, and subsequently all samples that improve existing cells by their difference in improvement. We then compute an ascending gradient step as a linear combination of gradients (line 14), following the recombination weights \( w_i \) from CMA-ES based on the computed improvement ranking. These weights correspond to the log-likelihood probabilities of the samples in the natural gradient interpretation of CMA-ES (1).

In line 16, CMA-ES adapts the multi-variate Gaussian \( N(\mu, \Sigma) \), as well as internal search parameters \( p \), from the improvement ranking of the coefficients. In the supplemental material, we provide a natural gradient interpretation of the improvement ranking rules of CMA-MEGA, where we show that the coefficient distribution of CMA-MEGA approximates natural gradient steps of maximizing a modified QD objective.

CMA-MEGA (Adam). We add an Adam-based variant of CMA-MEGA, where we replace line 15 with an Adam gradient optimization step 38.

OG-MAP-Elites. To show the importance of taking gradient steps in the measure space, as opposed to only taking gradient steps in objective space and directly perturbing the parameters, we derive two variants of MAP-Elites as a baseline that draw insights from the recently proposed Policy Gradient
Algorithm 1 Covariance Matrix Adaptation MAP-Elites via a Gradient Aborescence (CMA-MEGA)

**CMA-MEGA** (evaluate, \( \theta_0, N, \lambda, \eta, \sigma_g \))

**input**: An evaluation function `evaluate` which computes the objective, the measures, gradients of the objective and measures, an initial solution \( \theta_0 \), a desired number of iterations \( N \), a branching population size \( \lambda \), a learning rate \( \eta \), and an initial step size for CMA-ES \( \sigma_g \).

**result**: Generate \( N(\lambda + 1) \) solutions storing elites in an archive \( A \).

1. Initialize solution parameters \( \theta \) to \( \theta_0 \), CMA-ES parameters \( \mu = 0, \Sigma = \sigma_g I \), and \( p \), where we let \( p \) be the CMA-ES internal parameters.

2. for `iter ← 1 to N` do
   
   3. \( f, \nabla_f, m, \nabla_m \leftarrow \text{evaluate}(\theta) \)
   
   4. \( \nabla_f \leftarrow \text{normalize}(\nabla_f), \nabla_m \leftarrow \text{normalize}(\nabla_m) \)
   
   5. `update_archive(\( \theta, f, m \))`

   6. for `i ← 1 to \( \lambda \)` do

   7. \( c \sim \mathcal{N}(\mu, \Sigma) \)
   
   8. \( \nabla_i \leftarrow c_0 \nabla_f + \sum_{j=1}^{\lambda} c_j \nabla_m \)
   
   9. \( \theta'_i \leftarrow \theta + \Delta_i \)
   
   10. \( f', m', \ast \leftarrow \text{evaluate}(\theta'_i) \)
   
   11. \( \Delta_i \leftarrow \text{update_archive}(\theta_i, f', m') \)

   end

12. rank \( \nabla_i \) by \( \Delta_i \)

13. \( \nabla_{\text{step}} \leftarrow \sum_{i=1}^{\lambda} \nabla \text{rank}[i] \)

14. \( \theta \leftarrow \theta + \eta \nabla_{\text{step}} \)

15. `Adapt CMA-ES parameters \( \mu, \Sigma, p \)` based on improvement ranking \( \Delta_i \)

16. if `there is no change in the archive` then

17.  `Restart CMA-ES with \( \mu = 0, \Sigma = \sigma_g I \).`

18.  `Set \( \theta \)` to a randomly selected existing cell \( \theta_i \) from the archive

end

end

---

Assisted MAP-Elites (PGA-ME) algorithm [45]. PGA-ME combines the Iso+LineDD operator [61] with a policy gradient operator only on the objective. Similarly, our proposed Objective-Gradient Assisted MAP-Elites (OG-MAP-Elites) algorithm combines an objective gradient step with a MAP-Elites style perturbation operator. Each iteration of OG-MAP-Elites samples \( \lambda \) solutions \( \theta_i \) from the archive. Each \( \theta_i \) is perturbed with Gaussian noise to form a new candidate solution \( \theta'_i = \theta_i + \sigma \mathcal{N}(0, I) \). OG-MAP-Elites evaluates the solution and updates the archive, exactly as in MAP-Elites. However, OG-MAP-Elites takes one additional step: for each \( \theta'_i \), the algorithm computes \( \nabla f(\theta'_i) \), forms a new solution \( \theta''_i = \theta'_i + \eta \nabla f(\theta'_i) \) with an objective gradient step, and evaluates \( \theta''_i \). Finally, we update the archive with all solutions \( \theta'_i \) and \( \theta''_i \).

OG-MAP-Elites (line). Our second baseline, OG-MAP-Elites (line) replaces the Gaussian operator with the Iso+LineDD operator [61]: \( \theta' = \theta_i + \sigma_1 \mathcal{N}(0, I) + \sigma_2 \mathcal{N}(0, .1) (\theta_i - \theta_j) \). We consider OG-MAP-Elites (line) a DQD variant of PGA-ME. However, PGA-ME was designed as a reinforcement learning (RL) algorithm, thus many of the advantages gained in RL settings are lost in OG-MAP-Elites (line). We provide a detailed discussion and ablations in the supplemental material.

---

5 Domains

DQD requires differentiable objective and measures, thus we select benchmark domains from previous work in the QD literature where we can compute the gradients of the objective and measure functions.

Linear Projection. To show the importance of adaptation mechanisms in QD, the CMA-ME paper [19] introduced a simple domain, where reaching the extremes of the measures is challenging for non-adaptive QD algorithms. The domain forms each measure \( m_i \) by a linear projection from \( \mathbb{R}^n \) to \( \mathbb{R} \), while bounding the contribution of each component \( \theta_i \) to the range \([-5, 12, 5, 12]\).

We note that uniformly sampling from a hypercube in \( \mathbb{R}^n \) results in a narrow distribution of the linear projection in \( \mathbb{R} \) [19]. Increasing the number of parameters \( n \) makes the problem of covering the
Table 1: Mean QD-score and coverage values after 10,000 iterations for each algorithm per domain.

| Algorithm       | LP (sphere) | LP (Rastrigin) | Arm Repertoire | LSI          |
|-----------------|-------------|----------------|----------------|--------------|
|                 | QD-score    | Coverage       | QD-score       | Coverage     | QD-score      | Coverage     |
| MAP-Elites      | 1.04        | 1.17%          | 1.18           | 1.72%        | 1.97          | 8.06%        | 13.88        | 23.15%       |
| MAP-Elites (line) | 12.21       | 14.32%         | 8.12           | 11.79%       | 33.51         | 35.79%       | 16.54        | 25.73%       |
| CMA-ME          | 1.08        | 1.21%          | 1.21           | 1.76%        | 55.98         | 56.95%       | 18.96        | 26.18%       |
| OG-MAP-Elites   | 1.52        | 1.67%          | 0.83           | 1.26%        | 57.17         | 58.08%       | N/A          | N/A          |
| OG-MAP-Elites (line) | 15.01       | 17.41%         | 6.10           | 8.85%        | 59.66         | 60.28%       | N/A          | N/A          |
| OMG-MEGA        | 71.58       | 92.09%         | 55.90          | 77.00%       | 44.12         | 44.13%       | N/A          | N/A          |
| CMA-MEGA        | 75.29       | 100.00%        | 62.54          | 100.00%      | 74.18         | 74.18%       | 5.36         | 8.61%        |
| CMA-MEGA (Adam) | 75.30       | 100.00%        | 62.58          | 100.00%      | 73.82         | 73.82%       | 21.82        | 30.73%       |

measure space more challenging, because to reach an extremum \( m_i(\theta) = \pm 5.12n \), all components must equal the extremum: \( \theta[i] = \pm 5.12 \).

We select this domain as a benchmark to highlight the need for adaptive gradient coefficients for CMA-MEGA as opposed to constant coefficients for OMG-MEGA, because reaching the edges of the measure space requires dynamically shrinking the gradient steps.

As a QD domain, the domain must provide an objective. The CMA-ME study [19] introduces two variants of the linear projection domain with an objective based on the sphere and Rastrigin functions from the continuous black-box optimization set of benchmarks [29, 31]. We optimize an \( n = 1000 \) unbounded parameter space \( \mathbb{R}^n \). We provide more detail in the supplemental material.

**Arm Repertoire.** We select the robotic arm repertoire domain from previous work [13, 61]. The goal in this domain is to find an inverse kinematics (IK) solution for each reachable position of the end-effector of a planar robotic arm with revolute joints. The objective \( f \) of each solution is to minimize the variance of the joint angles, while the measure functions are the positions of the end effector in the \( x \) and \( y \)-axis, computed with the forward kinematics of the planar arm [44]. We selected a 1000-DOF robotic arm.

**Latent Space Illumination.** Previous work [20] introduced the problem of exploring the latent space of a generative model directly with a QD algorithm. The authors named the problem latent space illumination (LSI). As the original LSI work evaluated non-differentiable objectives and measures, we create a new benchmark for the differentiable LSI problem by generating images with StyleGAN [36] and leveraging CLIP [51] to create differentiable objective and measure functions. We adopt the StyleGAN+CLIP [48] pipeline, where StyleGAN-generated images are passed to CLIP, which in turn evaluates how well the generated image matches a given text prompt. We form the prompt “Elon Musk with short hair,” as the objective and for the measures we form the prompts “A person with red hair,” and “A man with blue eyes.”. The goal of DQD becomes generating faces similar to Elon Musk with short hair, but varying with respect to hair and eye color.

6 Experiments

We conduct experiments to assess the performance of the MEGA variants. In addition to our OG-MAP-Elites baselines, which we propose in section 4, we compare the MEGA variants with the state-of-the-art QD algorithms presented in section 3. We implemented MEGA and OG-MAP-Elites variants in the Pyribs [59] QD library and compare against the existing Pyribs implementations of MAP-Elites, MAP-Elites (line), and CMA-ME.

6.1 Experiment Design

**Independent Variables.** We follow a between-groups design, where the independent variables are the algorithm and the domain (linear projection, arm repertoire, and LSI). We did not run OMG-MEGA and OG-MAP-Elites in the LSI domain; while CMA-MEGA computes the \( f \) and \( m_i \) gradients once per iteration (line 3 in Algorithm 1), OMG-MEGA and OG-MAP-Elites compute the \( f \) and \( m_i \) gradients for every sampled solution, making their execution cost-prohibitive for the LSI domain.

**Dependent Variables.** We measure both the diversity and the quality of the solutions returned by each algorithm. These are combined by the QD-score metric [49], which is defined as the sum of \( f \) values of all cells in the archive (Eq. 1). To make the QD-score invariant with respect to the
resolution of the archive, we normalize QD-score by the archive size (the total number of cells from
the tessellation of the measure space). As an additional metric of diversity we compute the coverage
as the number of occupied cells in the archive divided by the total number of cells. We run each
algorithm for 20 trials in the linear projection and arm repertoire domains, and for 5 trials in the LSI
domain, resulting in a total of 445 trials.

6.2 Analysis

Table 1 shows the metrics of all the algorithms, averaged over 20 trials for the benchmark domains
and over 5 trials for the LSI domain. We conducted a two-way ANOVA to examine the effect of
algorithm and domain (linear projection (sphere), linear projection (Rastrigin), arm repertoire) on
the QD-Score. There was a statistically significant interaction between the search algorithm and
the domain \( F(14, 456) = 7328.18, p < 0.001 \). Simple main effects analysis with Bonferroni
corrections showed that CMA-MEGA and OMG-MEGA performed significantly better than each of
the baselines in the sphere and Rastrigin domains, with CMA-MEGA significantly outperforming
OMG-MEGA. CMA-MEGA also outperformed all the other algorithms in the arm repertoire domain.
We additionally conducted a one-way ANOVA to examine the effect of algorithm on the LSI domain. There was a statistically significant difference between groups \(F(4, 20) = 260.64, p < 0.001\). Post-hoc pairwise comparisons with Bonferroni corrections showed that CMA-MEGA (Adam) significantly outperformed all other algorithms, while CMA-MEGA without the Adam implementation had the worst performance.

Both OMG-MEGA and CMA-MEGA variants perform well in the linear projection domain, where the objective and measure functions are additively separable, and the partial derivatives with respect to each parameter independently capture the steepest change of each function. We observe that OG-MAP-Elites performs poorly in this domain. Analysis shows that the algorithm finds a nearly perfect best solution for the sphere objective, but it interleaves following the gradient of the objective with exploring the archive as in standard MAP-Elites, resulting in smaller coverage of the archive.

In the arm domain, OMG-MEGA manages to reach the extremes of the measure space, but the algorithm fails to fill in nearby cells. The OG-MAP-Elites variants perform significantly better than OMG-MEGA, because the top-performing solutions in this domain tend to be concentrated in an “elite hypervolume” [61]; moving towards the gradient of the objective finds top-performing cells, while applying isotropic perturbations to these cells fills in nearby regions in the archive. CMA-MEGA variants retain the best performance in this domain. Fig. 1 shows a high-precision view of the CMA-MEGA (Adam) archive for the arm repertoire domain.

We did not observe a large difference between the CMA-MEGA (Adam) and our gradient descent implementation in the first two benchmark domains, where the curvature of the search space is well-conditioned. On the other hand, in the LSI domain CMA-MEGA without the Adam implementation performed poorly. We conjecture that this is caused by the conditioning of the mapping from the latent space of the StyleGAN to the CLIP score.

Fig. 2 shows the QD-score values for increasing number of iterations for each of the tested algorithms, with 95% confidence intervals. The figure also presents heatmaps of the CMA-MEGA (Adam) and the generated archive of the strongest QD competitor for each of the three domains. We provide generated archives of all algorithms in the supplemental material.

We visualize the top performing solutions in the LSI domain by uniformly sampling solutions from the archive of CMA-MEGA (Adam) and showing the generated faces in Fig. 3. We observe that
as we move from the top right to the bottom left, the features matching the captions “a man with blue eyes” and “a person with red hair” become more prevalent. We note that these solutions were generated from a single run of CMA-MEGA (Adam) for 10,000 iterations.

Overall, these results show that using the gradient information in quality diversity optimization results in significant benefits to search efficiency, but adapting the gradient coefficients with CMA-ES is critical in achieving top performance.

7 Related Work

**Quality Diversity.** The precursor to QD algorithms [50] originated with diversity-driven algorithms as a branch of evolutionary computation. Novelty search [39], which maintains an archive of diverse solutions, ensures diversity though a provided metric function and was the first diversity-driven algorithm. Later, objectives were introduced as a quality metric resulting in the first QD algorithms: Novelty Search with Local Competition (NSLC) [40] and MAP-Elites [13, 43]. Since their inception, many works have improved the archives [18, 62, 57], the variation operators [61, 19, 11, 56], and the selection mechanisms [12, 56] of both NSLC and MAP-Elites. While the original QD algorithms were based on genetic algorithms, algorithms based on other derivative-free approaches such as evolution strategies [19, 10, 46] and Bayesian optimization [37] have recently emerged.

Being stochastic derivative-free optimizers [7], QD algorithms are frequently applied to reinforcement learning (RL) problems [47, 2, 14] as derivative information must be estimated in RL. Naturally, approaches combining QD and RL have started to emerge [45, 9]. Unlike DQD, these approaches estimate the gradient of the reward function, and in the case of QD-RL a novelty function, in action space and backpropagate this gradient through a neural network. Our proposed DQD problem differs by leveraging provided – rather than approximated – gradients of the objective and measure functions.

Several works have proposed model-based QD algorithms. For example, the DDE-Elites algorithm [23] dynamically trains a variational autoencoder (VAE) on the MAP-Elites archive, then leverages the latent space of this VAE by interpolating between archive solutions in latent space as a type of crossover operator. DDE-Elites learns a manifold of the archive data as a representation, regularized by the VAE’s loss function, to solve downstream optimization tasks efficiently in this learned representation. The PoMS algorithm [52] builds upon DDE-Elites by learning an explicit manifold of the archive data via an autoencoder. To overcome distortions introduced by an explicit manifold mapping, the authors introduce a covariance perturbation operator based on the Jacobian of the decoder network. These works differ from DQD by dynamically constructing a learned representation of the search space instead of leveraging the objective and measure gradients directly.

**Latent Space Exploration.** Several works have proposed a variety of methods for directly exploring the latent space of generative models. Methods on GANs include interpolation [60], gradient descent [6], importance sampling [64], and latent space walks [53]. Derivative-free optimization methods mostly consist of latent variable evolution (LVE) [4, 24], the method of optimizing latent space with an evolutionary algorithm. LVE was later applied to generating Mario levels [63] with targeted gameplay characteristics. Later work [20] proposed latent space illumination (LSI), the problem of exploring the latent space of a generative model with a QD algorithm. The method has only been applied to procedurally generating video game levels [20, 58, 55, 17] and generating MNIST digits [65]. Follow-up work explored LSI on VAEs [54]. Our work improves LSI on domains where gradient information on the objective and measures is available with respect to model output.

8 Societal Impacts

By proposing gradient-based analogs to derivative-free QD methods, we hope to expand the potential applications of QD research and bring the ideas of the growing QD community to a wider machine learning audience. We are excited about future mixing of ideas between QD, generative modeling, and other machine learning subfields.

In the same way that gradient descent is used to synthesize super-resolution images [42], our method can be used in the same context, which would raise ethical considerations due to potential biases present in the trained model [6]. On the other hand, we hypothesize that thoughtful selection of the measure functions may help counterbalance this issue, since we can explicitly specify the measures
that ensure diversity over the collection of generated outputs. For example, a model may be capable of generating a certain type of face, but the latent space may be organized in a way which biases a gradient descent on the latent space away from a specific distribution of faces. If the kind of diversity required is differentiably measurable, then DQD could potentially help resolve which aspect of the generative model, i.e., the structure of the latent space or the representational capabilities of the model, is contributing to the bias.

Finally, we recognize the possibility of using this technology for malicious purposes, including generation of fake images (“DeepFakes”), and we highlight the utility of studies that help identify DeepFake models [28].

9 Limitations and Future Work

Quality diversity (QD) is a rapidly emerging field [7] with applications including procedural content generation [27], damage recovery in robotics [13,43], efficient aerodynamic shape design [22], and scenario generation in human-robot interaction [16,17]. We have introduced differentiable quality diversity (DQD), a special case of QD, where measure and objective functions are differentiable, and have shown how a gradient arborescence results in significant improvements in search efficiency.

As both MEGA variants are only first order differentiable optimizers, we expect them to have difficulty on highly ill-conditioned optimization problems. CMA-ES, as an approximate second order optimizer, retains a full-rank covariance matrix that approximates curvature information and is known to outperform quasi-Newton methods on highly ill-conditioned problems [25]. CMA-ME likely inherits these properties by leveraging the CMA-ES adaptation mechanisms and we expect it to have an advantage on ill-conditioned objective and measure functions.

While we found CMA-MEGA to be fairly robust to hyperparameter changes in the first two benchmark domains (linear projection, arm repertoire), small changes of the hyperparameters in the LSI domain led CMA-MEGA, as well as all the QD baselines, to stray too far from the mean of the latent space, which resulted in many artifacts and unrealistic images. One way to address this limitation is to constrain the search region to a hypersphere of radius \(\sqrt{d}\), where \(d\) is the dimensionality of the latent space, as done in previous work [42].

While CLIP achieves state-of-the-art performance in classifying images based on visual concepts, the model does not measure abstract concepts. Ideally, we would like to specify “age” as a measure function and obtain quantitative estimates of age given an image of a person. We believe that the proposed work on the LSI domain will encourage future research on this topic, which we would in turn be able to integrate with DQD implementations to generate diverse, high quality content.

Many problems, currently modelled as optimization problems, may be fruitfully redefined as QD problems, including the training of deep neural networks. Our belief stems from recent works [53,51], which reformulated deep learning as a multi-objective optimization problem. However, QD algorithms struggle with high-variance stochastic objectives and measures [35,15], which naturally conflicts with minibatch training in stochastic gradient descent [5]. These challenges need to be addressed before DQD training of deep neural networks becomes tractable.

Acknowledgments and Disclosure of Funding

We would like to thank the anonymous reviewers fNzE, 4FYu, T1rp, and ghwT for their detailed feedback, thorough comments, and corrections throughout the review process that helped us improve the quality of the paper. We thank the area chair and senior area chair for their assessment, paper recommendation, and kind words about the paper. We thank Lisa B. Soros for her feedback on a preliminary version of this work and Varun Bhatt for his assistance running additional experiments for the final version of the paper.

This work was partially supported by the National Science Foundation NRI (# 2024936) and the Alpha Foundation (# AFC820-68).
References

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

[2] Kai Arulkumaran, Antoine Cully, and Julian Togelius. Alphastar: An evolutionary computation perspective.
    In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pages 314–315, 2019.

[3] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of
    generative networks. arXiv preprint arXiv:1707.05776, 2017.

[4] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. Deepmasterprints: Generating
    masterprints for dictionary attacks via latent variable evolution. In 2018 IEEE 9th International Conference
    on Biometrics Theory, Applications and Systems (BTAS), pages 1–9. IEEE, 2018.

[5] Léon Bottou. Stochastic gradient descent tricks. In Neural networks: Tricks of the trade, pages 421–436.
    Springer, 2012.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial
    gender classification. In Conference on fairness, accountability and transparency, pages 77–91. PMLR, 2018.

[7] Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-
    diversity optimization: a novel branch of stochastic optimization. In Black Box Optimization, Machine
    Learning, and No-Free Lunch Theorems, pages 109–135. Springer, 2021.

[8] Yoeng-Jin Chu. On the shortest arborescence of a directed graph. Scientia Sinica, 14:1396–1400, 1965.

[9] Geoffroy Cideron, Thomas Pierrot, Nicolas Perrin, Karim Beguir, and Olivier Sighaud. Qd-r1: Efficient
    mixing of quality and diversity in reinforcement learning. arXiv preprint arXiv:2006.08505, 2020.

[10] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. Scaling map-elites to deep neuroevolu-
    tion. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pages 67–75, 2020.

[11] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O Stanley, and Jeff
    Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of
    novelty-seeking agents. arXiv preprint arXiv:1712.06560, 2017.

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

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

[14] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore.
    Nature, 590(7847):580–586, 2021.

[15] Manon Flageat and Antoine Cully. Fast and stable map-elites in noisy domains using deep grids. In
    Artificial Life Conference Proceedings, pages 273–282. MIT Press, 2020.

[16] Matthew Fontaine and Stefanos Nikolaidis. A quality diversity approach to automatically generating
    human-robot interaction scenarios in shared autonomy. Robotics: Science and Systems, 2021. to appear.

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

[18] Matthew Fontaine, Scott Lee, Lisa B Soros, Fernando de Mesentier Silva, Julian Togelius, and Amy K
    Hoover. Mapping hearthstone deck spaces through map-elites with sliding boundaries. In Proceedings of
    The Genetic and Evolutionary Computation Conference, pages 161–169, 2019.

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

[20] Matthew C Fontaine, Ruilin Liu, Julian Togelius, Amy K Hoover, and Stefanos Nikolaidis. Illuminating
    mario scenes in the latent space of a generative adversarial network. In Proceedings of the AAAI Conference
    on Artificial Intelligence, 2021.

[21] Kevin Frans, LB Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through
    language-image encoders. arXiv preprint arXiv:2106.14843, 2021.

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

[23] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Automating representation discovery with
    map-elites. arXiv preprint arXiv:2003.04389, 2020.
[24] Federico A Galatolo, Mario GCA Cimino, and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. arXiv preprint arXiv:2102.01645, 2021.

[25] Tobias Glasmachers and Oswin Krause. The hessian estimation evolution strategy. In International Conference on Parallel Problem Solving from Nature, pages 597–609. Springer, 2020.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.

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

[28] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2018.

[29] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockhoff. Coco: A platform for comparing continuous optimizers in a black-box setting. 2016.

[30] Nikolaus Hansen. The cma evolution strategy: A tutorial. arXiv preprint arXiv:1604.00772, 2016.

[31] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbo-b-2009. pages 1689–1696, 07 2010. doi: 10.1145/1830761.1830790.

[32] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019. URL https://doi.org/10.5281/zenodo.2559634.

[33] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. arXiv preprint arXiv:1907.07171, 2019.

[34] Norman L Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. Continuous univariate distributions, volume 2, volume 289. John wiley & sons, 1995.

[35] Niels Justesen, Sebastian Risi, and Jean-Baptiste Mouret. Map-elites for noisy domains by adaptive sampling. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pages 121–122, 2019.

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4401–4410, 2019.

[37] Paul Kent and Juergen Branke. Bop-elites, a bayesian optimisation algorithm for quality-diversity search. arXiv preprint arXiv:2005.04320, 2020.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

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

[40] Joel Lehman and Kenneth O Stanley. 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, 2011.

[41] Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. Annals of Operations Research, pages 1–30, 2021.

[42] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2437–2445, 2020.

[43] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909, 2015.

[44] Richard M Murray, Zexiang Li, and S Shankar Sastry. A mathematical introduction to robotic manipulation. CRC press, 2017.

[45] Olle Nilsson and Antoine Cully. Policy gradient assisted map-elites. Proceedings of the Genetic and Evolutionary Computation Conference, 2021.

[46] Jørgen Nordmoen, Eivind Samuelsen, Kai Olav Ellefsen, and Kyrre Glette. Dynamic mutation in map-elites for robotic repertoire generation. In Artificial Life Conference Proceedings, pages 598–605. MIT Press, 2018.
[47] Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population-based reinforcement learning. arXiv preprint arXiv:2002.00632, 2020.

[48] Victor Perez. Generating images from prompts using clip and stylegan, 2021. URL https://towardsdatascience.com/ MIT License.

[49] Justin K Pugh, Lisa B Soros, Paul A Szerlip, and Kenneth O Stanley. Confronting the challenge of quality diversity. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pages 967–974, 2015.

[50] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics and AI, 3:40, 2016.

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning ICML 2021, 2021.

[52] Nemanja Rakicevic, Antoine Cully, and Petar Kormushev. Policy manifold search: Exploring the manifold hypothesis for diversity-based neuroevolution. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’21, page 901–909, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383509. doi: 10.1145/3449639.3459320.

[53] Michael Ruchte and Josif Grabocka. Efficient multi-objective optimization for deep learning. arXiv preprint arXiv:2103.13392, 2021.

[54] Anurag Sarkar and Seth Cooper. Generating and blending game levels via quality-diversity in the latent space of a variational autoencoder. arXiv preprint arXiv:2102.12463, 2021.

[55] Jacob Schrum, Vanessa Volz, and Sebastian Risi. Cppn2gan: Combining compositional pattern producing networks and gans for large-scale pattern generation. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pages 139–147, 2020.

[56] Konstantinos Sfikas, Antonios Liapis, and Georgios N. Yannakakis. Monte carlo elites: Quality-diversity selection as a multi-armed bandit problem. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’21, page 149–156, 2018.

[57] Vassilis Vassiliades and Jean-Baptiste Mouret. Discovering the elite hypervolume by leveraging interspecies correlation. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 221–228, 2018.

[58] Tom White. Sampling generative networks. arXiv preprint arXiv:1609.04468, 2016.

[59] Yulun Zhang, Bryon Tjanaka, Matthew C. Fontaine, and Stefanos Nikolaidis. pyribs: A bare-bones python library for quality diversity optimization. https://github.com/icaros-usc/pyribs, 2021.

[60] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snively, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7064–7073, 2017.

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

[62] Yulun Zhang, Bryon Tjanaka, Matthew C. Fontaine, and Stefanos Nikolaidis. Illuminating the latent space of an mnist gan. https://docs.pyribs.org/en/stable/tutorials/lsl_mnist.html MIT License.
Appendix

A Hyperparameter Selection

For the arm and linear projection domains we mirror the hyperparameter selections from previous work \cite{61, 19} and tuned manually the hyperparameters of our newly proposed algorithms: OG-MAP-Elites, OG-MAP-Elites (line), OMG-MEGA, and CMA-MEGA. For the latent space illumination domain, since the domain is new, we manually tuned each algorithm so that the learning rate was as small as possible while still have perceptible differences in the first 10 iterations of each algorithm. We chose small learning rates because for large learning rates, the search moved far away from the training distribution, resulting in unrealistic images. We report the parameters of each algorithm below. In all algorithms we used a batch size $\lambda = 36$ following previous work \cite{19}. MAP-Elites and the algorithms derived from MAP-Elites (MAP-Elites (line), OG-MAP-Elites, OG-MAP-Elites (line), OMG-MEGA) had an initial population size of 100, sampled from a distribution $\mathcal{N}(0, I)$. Initial solutions used to seed the initial archive do not count as an iteration in our experiments. We set the initial search position $\theta_0 = 0$ for CMA-ME, CMA-MEGA and CMA-MEGA (Adam) in all domains.

Linear Projection (Sphere, Rastrigin).

- MAP-Elites: $\sigma = 0.5$
- MAP-Elites (line): $\sigma_1 = 0.5, \sigma_2 = 0.2$
- CMA-ME: $\sigma = 0.5$
- OG-MAP-Elites: $\sigma = 0.5, \eta = 0.5$
- OG-MAP-Elites (line): $\sigma_1 = 0.5, \sigma_2 = 0.2, \eta = 0.5$
- OMG-MEGA: $\sigma_g = 10.0$
- CMA-MEGA: $\sigma_g = 10.0, \eta = 1.0$
- CMA-MEGA (Adam): $\sigma_g = 10.0, \eta = 0.002$

Arm Repertoire.

- MAP-Elites: $\sigma = 0.1$
- MAP-Elites (line): $\sigma_1 = 0.1, \sigma_2 = 0.2$
- CMA-ME: $\sigma = 0.2$
- OG-MAP-Elites: $\sigma = 0.1, \eta = 100$
- OG-MAP-Elites (line): $\sigma_1 = 0.1, \sigma_2 = 0.2, \eta = 100$
- OMG-MEGA: $\sigma_g = 1.0$
- CMA-MEGA: $\sigma_g = 0.05, \eta = 1.0$
- CMA-MEGA (Adam): $\sigma_g = 0.05, \eta = 0.002$

Latent Space Illumination of StyleGAN guided by CLIP.

- MAP-Elites: $\sigma = 0.2$
- MAP-Elites (line): $\sigma_1 = 0.1, \sigma_2 = 0.2$
- CMA-ME: $\sigma = 0.02$
- CMA-MEGA: $\sigma_g = 0.002, \eta = 1.0$
- CMA-MEGA (Adam): $\sigma_g = 0.002, \eta = 0.002$

Adam Hyperparameters. We use the same hyperparameters as the StyleGAN+CLIP implementation \cite{48}. We configure Adam with the same hyperparameters for each domain.

- $\beta_1 = 0.9$
- $\beta_2 = 0.999$
B Domain Details.

Linear Projection. We use the linear projection domains sphere and Rastrigin from previous work [19], and selected \( n = 1000 \) for both domains. Each domain has a different objective function (Eq. 4, Eq. 5) but identical measure functions. As in previous work [18], we offset the objective to move the optimal location away from the center of the search space, to \( x_i = 5.12 \cdot 0.4 = 2.048 \).

\[
sphere(x) = \sum_{i=1}^{n} x_i^2 \quad (4)
\]

\[
\text{Rastrigin}(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i)] \quad (5)
\]

We define the 2D measure space with the projection of the first and second half of the components \( x_i \) (see Eq. 7). We bound the contribution of each component through a \( \text{clip} \) function (Eq. 6), which restricts the measure contribution of each \( x_i \) to the range \([-5.12, 5.12] \).

\[
\text{clip}(x_i) = \begin{cases} 
  x_i & \text{if } -5.12 \leq x_i \leq 5.12 \\
  \frac{5.12}{x_i} & \text{otherwise} 
\end{cases} \quad (6)
\]

\[
m(x) = \left( \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \text{clip}(x_i), \sum_{i=\lfloor \frac{n}{2} \rfloor+1}^{n} \text{clip}(x_i) \right) \quad (7)
\]

We observe that the partial derivatives for \( m_i \) are 1 for the range \([-5.12, 5.12] \). A constant derivative means that the gradient step will not shrink as OMG-MEGA approaches an extreme point in measure space, thus OMG-MEGA often overshoots the bounds; on the other hand, CMA-MEGA dynamically adapts the gradient steps, allowing for efficient coverage the measure space.

Fig. 4 visualizes the challenge of the linear projection domain. Observe that if we sample uniformly on the hypercube \([-5.12, 5.12]^n \), then each of our measure functions becomes a sum of random variables. If we normalize each measure by dividing by \( n \), then our measure functions become an average of random variables. The average of \( n \) random variables forms the Bates distribution [34], a distribution that narrows as \( n \) increases (Fig. 4(a)). At \( n = 1000 \) the solutions sampled from the hypercube are very close to 0 in measure space with high probability. A QD algorithm could simply increase its step-size to move to extremes of measure space, but the \( \text{clip} \) function prevents this by implicitly bounding the extremes of the measure space; each component of \( \theta \) can contribute at most \( \pm 5.12 \) to change the position in measure space. We note the heavy penalty in the \( \text{clip} \) function for a component leaving the range \([-5.12, 5.12] \). The combination of the narrowness of the Bates distribution and the implicit bounding of the \( \text{clip} \) function means that a QD algorithm must dynamically adapt its sampling distribution by shrinking step-sizes as the distribution gets close to the extremes of the measure space.

Arm Repertoire. We visualize example solutions for a \( n = 7 \) (7-DOF) planar arm in Fig. 5. The optimal solutions in this domain have zero joint angle variance from the mean (all angles are equal). We note that we selected as objective the variance, instead of the standard deviation used in previous work [61], so that the objective is differentiable at 0. The measures for the arm repertoire have range \([-n, n] \), since they are the positions of the end-effector, and the arm has \( n \) links of length \( l_i = 1 \). Thus, the reachable space forms a filled circle with radius \( n \), and the maximum archive coverage becomes \( \frac{\pi n^2}{4n^2} \approx 78.5\% \). We selected \( n = 1000 \) (1000-DOF) for the experiments.

Latent Space Illumination. The latent space of StyleGAN [36] has size 512 where a latent code is repeated 18 times for each level of detail.

Transformations of the Objective Function. The QD-score metric, which we use to estimate the performance of QD algorithms, associates solution quality with maximizing the objective value \( f \), and assumes a strictly positive objective value \( f \) for every solution. Therefore, in each domain we transform the objective through a linear transformation of the form \( f' = af + b \) to map the objective values to the range \([0, 100] \).
In the linear projection domain, we follow the objective transformation proposed in previous work [19]. The original objective is to minimize the sphere and Rastrigin functions. We compute an estimate on the largest sphere and Rastrigin values in the hypercube $[-5.12, 5.12]^n$ which contain only components in the linear portion of the clip function. We compute $f(-5.12, -5.12, ..., -5.12)$ as an estimated maximum of the function $f_{max}$. The minimum of each function is $f_{min} = 0$. We then remap the function values for both the sphere and Rastrigin objective functions through the linear transformation given by:

$$f'(\theta) = 100 \cdot \frac{f(\theta) - f_{max}}{f_{min} - f_{max}}$$  \hspace{1cm} (8)

For the arm domain we estimate $f_{max}$ to be 1 from an initial population of angles sampled from $\mathcal{N}(0, I)$. For the LSI domain we picked $f_{max} = 10$ by observing the CLIP loss function values for the objective text prompt “Elon Musk with short hair”.

C Implementation

Archives. In the both the linear projection and arm repertoire domains, the measure space is 2D, with resolution $100 \times 100$. We normalize the objective value $f$ so that it is in the range $[0,100]$ and initialize empty cells of the archive with an objective value $f$ of 0. For latent space illumination we form a 2D archive with the resolution $200 \times 200$. We double the resolution in each dimension as the examined QD algorithms fill roughly a quarter of possible cells in the archive.

Computational Resources. We ran 10,000 iterations in all algorithms. We ran all trials in parallel on an AMD Ryzen Threadripper 32-core (64 threads) processor and an GeForce RTX 3090 Nvidia GPU. A run of 20 trials in parallel required about 30’ for the linear projection domains and 2 hours
for the arm repertoire domain. One trial run for the latent space illumination domain took about 2 hours. We note runtime increases at a higher logging frequency and algorithms which perform better may run slower due to iterating over more archive solutions when QD statistics are calculated.

Software Implementation. We use the publicly available Pyribs library for all algorithms, where we implemented the OG-MAP-Elites, OG-MAP-Elites (line), OMG-MEGA and CMA-MEGA algorithms. We use the Adam implementation of ESTool.7

D Improvement Ranking as a Natural Gradient Approximation

We show for the first time that CMA-ME’s improvement ranking optimizes a modified QD objective (eq. 10) via a natural gradient approximation. We then extend our derivation to show that the coefficient distribution of CMA-MEGA approximates natural gradient steps of the same objective with respect to the gradient coefficients.

We let the original QD objective for an archive A be:

\[ \max J_1(A) = \sum_{i=1}^{M} f(\theta_i) \] (9)

where \( f(\theta_i) = 0 \) if a cell \( i \) is unfilled.

Consider that MAP-Elites solves a discrete optimization problem over archives \( A \) rather than a continuous optimization problem over solution parameters \( \theta \). To analyze the improvement ranking, we reparameterize the objective \( J_1 \) to include \( \theta \), the current search position for CMA-ES. By fixing the current archive \( A \), our goal is to show that improvement ranking approximates a natural gradient step of \( J_1 \) with respect to \( \theta \). In other words, we want to calculate the direction to change \( \theta \) that results in the steepest increase of \( J_1 \) after \( \theta \) is added to the archive.

Ranking candidate solutions \( \theta \) based on \( J_1 \) may prioritize solutions that improve existing cells on the archive over solutions that discover new cells. To show this possibility, let \( \theta_i \) and \( \theta_j \) be two arbitrary candidate solutions, where \( \theta_i \) replaces a cell with an existing occupant \( \theta_p \) and \( \theta_j \) discovers a new cell. If \( f(\theta_i) - f(\theta_p) > f(\theta_j) \), then \( \theta_i \) will be ranked higher than \( \theta_j \).

To strictly prioritize exploration, CMA-ME performs a two-stage improvement ranking on the objective \( J_1 \), where it ranks first all solutions \( \theta_j \) that discover new cells based on their objective value \( f(\theta_j) \), and subsequently all solutions \( \theta_i \) that improve existing cells based on the difference \( f(\theta_i) - f(\theta_p) \) between the \( f \) value of the new and previous solution \( \theta_p \).

To show that the ranking rules of CMA-ME are equivalent to a natural gradient of a QD objective, we specify a function \( J_2 \) (see Eq. 10), and we will show that sorting solutions purely by \( J_2 \) results in the same order as the improvement ranking by CMA-ME on the objective \( J_1 \).

\[ \max J_2(A) = \sum_{i=1}^{M} [f(\theta_i) + b_i C] \] (10)

\[ C > 2 \cdot \max_{\theta_i \in \mathbb{R}^n} f(\theta_i) - \min_{\theta_i \in \mathbb{R}^n} f(\theta_i) \] (11)

where \( C \) is larger than twice the largest gap between any pair of \( f(\theta_i) \) and \( f(\theta_j) \) (see Eq. 11), \( b_i \) is 1 if a cell is occupied and 0 otherwise, and \( f(\theta_i) = 0 \) if a cell \( i \) is unfilled.\footnote{https://github.com/hardmaru/estool/blob/master/es.py (line 70)}

Let \( \theta_i \) and \( \theta_j \) be two arbitrary candidate solutions whose addition to the archive changes the archive. We define \( \Delta_{ij}^1 = J_2(A + \theta_i) - J_2(A) \) and \( \Delta_{ij}^2 = J_2(A + \theta_j) - J_2(A) \), and we consider the following three cases:

1: Without loss of generality \( \theta_i \) improves an occupied cell and \( \theta_j \) discovers a new cell.

Based on the improvement ranking of CMA-ME, \( \theta_j \) will always be ranked higher than \( \theta_i \). We will show that the same ordering holds if we sort based on the objective \( J_2 \).
We let $\theta_p$ be the occupant replaced by $\theta_i$.

We show that $\Delta J^2_j > \Delta J^2_i$:

\[
\Delta J^2_j = f(\theta_j) + C \\
= f(\theta_j) + \frac{C}{2} + \frac{C}{2} \\
> f(\theta_j) + \frac{C}{2} + f(\theta_i) - f(\theta_p) \\
> f(\theta_i) - f(\theta_p) \\
= \Delta J^2_i
\] (12)

Where inequalities hold from the definition of $C$ (Eq. 11).

2: Both $\theta_i$ and $\theta_j$ discover new cells.

From the improvement ranking rule, $\theta_i$ will be ranked higher than $\theta_j$ iff $\Delta J^1_i > \Delta J^1_j$.

We show that this is equivalent to $\Delta J^2_i > \Delta J^2_j$:

\[
\Delta J^1_i > \Delta J^1_j \Leftrightarrow f(\theta_i) > f(\theta_j) \Leftrightarrow f(\theta_i) + C > f(\theta_j) + C \Leftrightarrow \Delta J^2_i > \Delta J^2_j
\] (13)

3: Both $\theta_i$ and $\theta_j$ improve existing cells.

We let $\theta_p$ the occupant replaced by $\theta_i$ and $\theta_r$ the occupant replaced by $\theta_j$.

\[
\Delta J^1_i > \Delta J^1_j \Leftrightarrow f(\theta_i) - f(\theta_p) > f(\theta_j) - f(\theta_r) \Leftrightarrow \Delta J^2_i > \Delta J^2_j
\] (14)

Therefore, the CMA-ME improvement ranking is identical to ranking purely based on $J^2$.

Previous work [1] has shown that CMA-ES approximates the natural gradient of its provided optimization objective function to a scale. At the same time, CMA-ES is invariant to order-preserving transformations of its objective function, since it is a comparison-based optimization method [30]. We provide an explicit objective function $J^2$, and we have shown that maximizing this function results in the same ordering as the one specified implicitly by the improvement ranking rules. Therefore, CMA-ME, which uses improvement ranking to update the CMA-ES parameters, approximates a natural gradient of $J^2$ to a scale.

CMA-MEGA sorts solutions via the same improvement ranking rules as CMA-ME to update an internal sampling distribution maintained by CMA-ES. Unlike CMA-ME, CMA-MEGA updates a sampling distribution of gradient step coefficients, and not solution parameters directly. Therefore, CMA-MEGA performs a form of adaptive search, where the algorithm dynamically changes the search hyperparameters (gradient step coefficients) to maximize $J^2$.

By connecting CMA-ME to previous work that connects natural gradient descent and CMA-ES [1], we gain a new perspective on both CMA-ME and CMA-MEGA. In each iteration, CMA-ME optimizes for a single solution whose addition to the archive results in the largest increase in $J^2$. However, after sampling, we update the archive which results in a small change to the optimization landscape. In other words, CMA-ME assumes that CMA-ES is a robust enough optimizer to handle small changes to its target objective.

To apply this new perspective to CMA-MEGA, we observe that CMA-MEGA solves the optimization problem of finding the chosen gradient coefficients that yield the largest increase in $J^2$ for a fixed $\theta$ and archive $A$. In other words, CMA-MEGA optimizes the objective $J^2$, as CMA-ME, but it does so directly in objective-measure space. However, after taking one optimization step, we change the single solution optimization problem by updating the archive and moving $\theta$. Just like CMA-ME, we assume that CMA-ES is a robust enough optimizer to handle changes in the optimization landscape brought about by changing the archive and $\theta$. 
We discuss the importance of normalizing gradients in the MAP-Elites via Gradient Arborescence algorithm and include results from an ablation over gradient normalization. We explore the importance of normalizing gradients through an ablation study on OMG-MEGA, where we compare OMG-MEGA with the normalization step to OMG-MEGA without the normalization step in the linear projection and arm repertoire domains.

To account for the scaled step-size after normalization in the linear projection domain, we set the $\sigma_g$ to $10$ in the normalized variant and to $0.5$ in the unnormalized variant, since the average gradient magnitude was approximately $20$. For the arm repertoire domain, we note that the objective gradient magnitude was approximately $10^{-2}$, while the measure gradient magnitudes were approximately $10^2$. We ran tuning experiments for $\sigma_g$ at scales $10^2$, $10^1$, $1$, $10^{-1}$, $10^{-2}$, $10^{-3}$, and $10^{-4}$ to scale the unnormalized gradient and selected $\sigma_g = 10^{-4}$ as the best performing hyperparameter.

Table 2 presents the results of the ablation study in the linear projection and arm domains. The quantitative results suggest that normalization greatly affects performance of the OMG-MEGA algorithm. Below we give our interpretation of why normalization plays such a significant factor on performance in each domain.

Consider the linear projection sphere domain. In this domain the objective is quadratic. As we move further away from the optimum, the magnitude of the gradient increases linearly. In OMG-MEGA, the gradient coefficients are sampled from a fixed distribution. This means the gradient of our objective will dominate the gradients of the measures for positions in measure space far away from the optimum. Fig. 6 visualizes a final archive from this domain. We see that the unnormalized OMG-MEGA easily fills cells near the global optimum, but struggles to fill cells with a lower objective value. In contrast OMG-MEGA with normalization fills the archive evenly.

### Table 2: OMG-MEGA with normalized (norm) and unnormalized (unnorm) gradients.

#### LP (sphere)

| Algorithm          | QD-score | Coverage | Best     |
|--------------------|----------|----------|----------|
| OMG-MEGA (norm)    | 71.58 ± 0.10 | 92.09 ± 0.21% | 100.00 ± 0.00 |
| OMG-MEGA (unnorm)  | 56.96 ± 0.24 | 67.30 ± 0.38% | 100.00 ± 0.00 |

#### LP (Rastrigin)

| Algorithm          | QD-score | Coverage | Best     |
|--------------------|----------|----------|----------|
| OMG-MEGA (norm)    | 55.90 ± 0.21 | 77.00 ± 0.34% | 97.55 ± 0.06 |
| OMG-MEGA (unnorm)  | 16.56 ± 0.11 | 25.97 ± 0.18% | 89.76 ± 0.42 |

#### Arm Repertoire

| Algorithm          | QD-score | Coverage | Best     |
|--------------------|----------|----------|----------|
| OMG-MEGA (norm)    | 44.12 ± 0.06 | 44.13 ± 0.06% | 100.00 ± 0.00 |
| OMG-MEGA (unnorm)  | 0.85 ± 0.04  | 7.02 ± 0.21%  | 12.58 ± 0.36  |

E On the Importance of Normalizing Gradients

We discuss the importance of normalizing gradients in the MAP-Elites via Gradient Arborescence algorithm and include results from an ablation over gradient normalization. We explore the importance of normalizing gradients through an ablation study on OMG-MEGA, where we compare OMG-MEGA with the normalization step to OMG-MEGA without the normalization step in the linear projection and arm repertoire domains.
In the arm repertoire domain, we observe that the objective gradient is on a different scale than the measure gradients and this affects the performance of the unnormalized OMG-MEGA. The average objective gradient magnitude was $10^{-2}$. In contrast, the average measure gradient magnitude was $10^2$. This means that the objective gradient barely contributes to the search for a fixed coefficient distribution. However, the fact that OG-MAP-Elites performs well in this domain indicates that the objective gradient is an important factor in filling the archive. Because of the different scale, the objective gradient is largely ignored by the OMG-MEGA variant that does not normalize gradients.

While in OMG-MEGA the benefit of normalizing gradients is clear, we speculate that normalizing gradients is beneficial also for CMA-MEGA. In CMA-MEGA, the CMA-ES subroutine solves a non-stationary optimization problem for gradient coefficients that maximize the QD objective. The coefficients can be viewed as a learning rate for each gradient. In this sense, CMA-MEGA has adaptive learning rates, where CMA-ES acts as the adaptation mechanism. However, in standard gradient ascent the magnitude of the gradient acts as a step size control. If the gradients are not normalized, the magnitudes and the adaptive step size control may end up fighting each other. This may cause oscillations in CMA-ES’s adaptation mechanisms and lead to instability. We leave a thorough study to evaluate the exact benefits of normalization in CMA-MEGA as future work.

Finally, we note a conceptual difference for how we leverage measure gradients in DQD versus how objective gradients are used in optimization. In quality diversity, a goal of the algorithm is to cover the entire measure space. Intermediate solutions are of equal importance to solutions at the extremes of measure space. By normalizing gradients and giving control over the step-size to the DQD algorithm, CMA-MEGA can adapt its search distribution to more easily fill intermediate regions of the measure space, rather than only optimize for extrema.

### F On the Differences between OG-MAP-Elites (line) and PGA-MAP-Elites

We explore the design decisions between PGA-MAP-Elites [45], an reinforcement learning (RL) algorithm, and OG-MAP-Elites (line), a DQD baseline that draws insights from PGA-MAP-Elites. Specifically, PGA-MAP-Elites is an actor-critic reinforcement learning algorithm. Actor-critic methods combine elements of policy gradient methods, which approximate the gradient of the reward function with respect to a policy’s actions, and Q-learning methods, which approximate Q-values for every state-action pair. The computationally expensive part in this setting is the evaluation of a policy, while gradient computations do not factor into the running time of the algorithm.

On the other hand, the DQD problem assumes knowledge of the exact gradients of the objective and measure functions, which makes gradient approximation steps unnecessary. Thus, the most expensive part of the computation in many applications becomes the gradient computation for a given solution, rather than the evaluation of that solution. For example, if the function is a neural network, evaluation of a solution can be done efficiently with a forward pass. However, gradient computation requires an often expensive backpropagation of the loss function through the network. For efficient performance, a designer of a DQD algorithm would aim towards minimizing the number of gradient computations to achieve good results, instead of only the number of evaluations.

As an actor-critic method, PGA-MAP-Elites trains a Q-value approximator (a critic). As a quality diversity algorithm, the critic network can be trained from the rollout data of the entire archive of policies to form a better Q-value approximation. Since an accurate gradient approximation is important in RL settings, the shared critic network allows for efficient and accurate policy gradient calculation on demand for any new candidate policy evaluated. PGA-MAP-Elites consists of two independent operators, (1) the Iso+LineDD operator and (2) a policy gradient step.

In our implementation of the DQD algorithms in Pyribs’ ask-tell interface – adopted from Pycma [32], we evaluate a solution and pass the gradient simultaneously. If we directly implemented independent operators in OG-MAP-Elites (line), when using operator (2) we would need to evaluate a solution once to perform a gradient step and evaluate the new solution again after the gradient step. Instead, we apply in OG-MAP-Elites (line) sequentially the operators (1) and (2). This allows us to use the existing evaluation (and gradient computation) from (1) to perform a gradient step (2), reducing the number of evaluations needed per gradient step.

We implemented OG-MAP-Elites and OG-MAP-Elites (line) with the two independent operators to evaluate the effect of this design decision. We can divide our computation budget per iteration...
Table 3: OG-MAP-Elites with independent gradient and perturbation operators (denoted with †) compared against our sequential operators derivation from the main paper.

### Arm Repertoire

| Algorithm       | QD-score | Coverage | Best       |
|-----------------|----------|----------|------------|
| OG-MAP-Elites   | 57.21 ± 0.03 | 58.11 ± 0.03% | 98.63 ± 0.00 |
| OG-MAP-Elites (line) | 59.57 ± 0.03 | 60.19 ± 0.03% | 99.17 ± 0.00 |
| OG-MAP-Elites†  | 65.72 ± 0.06 | 65.95 ± 0.06% | 100.00 ± 0.00 |
| OG-MAP-Elites (line)† | 65.73 ± 0.04 | 65.96 ± 0.04% | 100.00 ± 0.00 |

### LP (sphere)

| Algorithm       | QD-score | Coverage | Best       |
|-----------------|----------|----------|------------|
| OG-MAP-Elites   | 1.36 ± 0.08 | 1.50 ± 0.09% | 100.00 ± 0.00 |
| OG-MAP-Elites (line) | 14.91 ± 0.08 | 17.29 ± 0.09% | 100.00 ± 0.00 |
| OG-MAP-Elites†  | 1.15 ± 0.10 | 1.27 ± 0.11% | 100.00 ± 0.00 |
| OG-MAP-Elites (line)† | 13.31 ± 0.09 | 15.42 ± 0.10% | 100.00 ± 0.00 |

### LP (Rastrigin)

| Algorithm       | QD-score | Coverage | Best       |
|-----------------|----------|----------|------------|
| OG-MAP-Elites   | 0.83 ± 0.03 | 1.28 ± 0.04% | 74.38 ± 0.04 |
| OG-MAP-Elites (line) | 6.09 ± 0.04 | 8.84 ± 0.07% | 76.67 ± 0.05 |
| OG-MAP-Elites†  | 0.76 ± 0.03 | 1.14 ± 0.03% | 74.43 ± 0.04 |
| OG-MAP-Elites (line)† | 5.14 ± 0.04 | 7.46 ± 0.06% | 76.17 ± 0.04 |

We run our ablation on the arm repertoire domain and both variants of the linear projection domain. Table 3 contains the results of the ablation. On both linear projection domains, the variants of OG-MAP-Elites and OG-MAP-Elites (line) with sequential operators outperform their counterparts with independent operators. However, in the arm repertoire domain, the independent operators implementation outperforms the sequential operator counterpart. We conjecture that because exploration is more difficult in the linear projection domain, the smaller number of perturbation operations in the independent operators variants hurts performance. On the other hand, in the arm repertoire domain, optimal solutions are concentrated in a small elite hypervolume. The independent operators can apply two or more gradient steps sequentially, directing the search efficiently towards the elite hypervolume.

### G On the Effect of the Archive Resolution on the Performance of CMA-ME

In this section, we comment on the performance of CMA-ME on the linear projection domain in the main paper and include an extra experiment with a different archive resolution.

The linear projection domain was first introduced by the authors of CMA-ME [19] to show the importance of adaptation mechanisms in quality diversity algorithms. In the paper, each domain was run with search dimensions of $n = 20$ and $n = 100$ for both the sphere and the Rastrigin objective. Experiments were run for $2.5 \times 10^6$ evaluations ($\approx 65,000$ iterations) and, due to the large number of evaluations, the archive resolution was $500 \times 500$. The paper reports CMA-ME outperforming both MAP-Elites and MAP-Elites (line) in this setting. In contrast, our experiments were run for 10,000 iterations with a search space dimension of $n = 1000$ and an archive resolution of $100 \times 100$.

We ran two additional experiments where we evaluated MAP-Elites, MAP-Elites (line), and CMA-ME on both linear projection domains with archive resolution of $500 \times 500$. Table 4 contains the results of the extra experiments and Fig. 7 visualizes a final archive of each algorithm for each experiment. We observe that under increased archive resolution CMA-ME outperforms both variants of MAP-Elites. We also observe that coverage increases for both MAP-Elites and CMA-ME when...
### Table 4: Results for each derivative-free QD algorithm evaluated on the linear projection domains. These experiments were run with archives of resolution 500 × 500 instead of 100 × 100.

| Algorithm           | QD-score | Coverage   | Best       |
|---------------------|----------|------------|------------|
|                     | LP (sphere)               |            |            |
| MAP-Elites          | 2.37 ± 0.01 | 2.86 ± 0.01% | 90.56 ± 0.02 |
| MAP-Elites (line)   | 4.96 ± 0.03 | 5.82 ± 0.04% | 92.87 ± 0.03 |
| CMA-ME              | 6.82 ± 0.05 | 7.74 ± 0.06% | 94.80 ± 0.07 |
|                     | LP (Rastrigin)             |            |            |
| MAP-Elites          | 1.87 ± 0.01 | 2.84 ± 0.01% | 74.46 ± 0.03 |
| MAP-Elites (line)   | 3.75 ± 0.02 | 5.45 ± 0.03% | 76.25 ± 0.04 |
| CMA-ME              | 4.92 ± 0.04 | 7.26 ± 0.05% | 76.15 ± 0.08 |

the archive is higher resolution. In contrast, coverage decreases for the MAP-Elites (line) algorithm when the archive resolution increases.

We explain why the low resolution of the archive negatively affects the performance of CMA-ME. For archives of resolution 100 × 100 there is not enough granularity in the archive to easily move through measure space, negatively affecting both MAP-Elites and CMA-ME. Moreover, CMA-ME restarts frequently due to the rule that the algorithm will restart from a random elite if none of the \( \lambda \) sampled solutions in an iteration improve the archive. Everytime CMA-ME restarts, it samples \( \lambda \) solutions by perturbing an elite chosen uniformly at random, with isotropic Gaussian noise. This limit case is nearly equivalent to the MAP-Elites algorithm, with the caveat that all generated solution come from one elite rather than \( \lambda \) different elites. In the 100 × 100 setting, CMA-ME performs similarly to MAP-Elites.

However, at a higher archive resolution of 500 × 500 restarts become less frequent and generated solutions in each iteration fall into different cells in the archive. The higher granularity of the archive causes CMA-ME to restart fewer times. Under this setting we see performance of CMA-ME consistent with the CMA-ME paper.

Overall, this study indicates the effect of archive resolution on the performance of QD algorithms.

### H Additional Results

#### H.1 Generated Archives and Additional Metrics.

Fig. 10 presents example archives for each algorithm and domain combination. Table 5 presents the values of the QD-score, coverage, and best solution for each algorithm and domain. We additionally ran MAP-Elites, MAP-Elites (line), and CMA-ME for 10^6 iterations, 20 trials, in the linear projection and arm repertoire domains, and present the results as MAP-Elites*, MAP-Elites (line)*, and CMA-ME*. We observe that running CMA-MEGA for only 10,000 iterations outperforms each algorithm in both domains.

We note that the QD-score metric combines both the number of cells filled in the archive (diversity) and the objective value of each occupant (quality). To disambiguate the two, we show in Fig. 11 the percentage of cells (y-axis) that have objective value greater than the threshold specified in the x-axis.

**Qualitative Results.** We can also compare the quality of the generated images with latent space illumination, with the images generated when using a single-objective Adam optimizer, where we optimize StyleGAN+CLIP [48] with Adam as a baseline (instead of running a QD algorithm). We run 5 different trials for 10,000 iterations, and for each trial we present the image that the algorithm

---

1 We note that the QD-score and coverage values of OG-MAP-Elites are slightly different between Table 1 and Table 5. Table 5 contains the correct values and we will update Table 1 in the revised version. The change does not affect the tests for statistical significance or any of our quantitative and qualitative findings of the experiments in section 6.
converged to. StyleGAN+CLIP with Adam gradually drifted towards unrealistic images, so we excluded from our selection images that were unrealistic or with large artifacts. We used the same prompt as the objective of section 6: “Elon Musk with short hair.”. We compare these images with the best image, according to the objective, of each of the 5 generated archives of CMA-MEGAd (Adam). We used in both conditions the Adam implementation of ESTool (see accompanying source code) with the same hyperparameters. We observe that the quality of the images is comparable (Fig. 12).

While running the LSI experiments, we discovered that several intermediate solutions were of better visual quality than the final images present in the generated collage. Fig. 13 shows two such images cherry-picked from all intermediate solutions generated by CMA-MEGAd (Adam) during the experiments. According to CLIP, these images are of lower quality than the images present in the collage of Fig. 3 in the main paper. We interpret this finding as CMA-MEGAd overfitting to the CLIP loss function. Several methods exist in the generative art community for mitigating this effect. For example, CLIPDraw [21] augments the generated image via randomized data augmentations then evaluates the generated batch with CLIP. The average loss of the entire batch replaces evaluating a single generated image with CLIP. A similar method could be applied to our LSI approach to improve the perceptual quality of generated images.

**H.2 Additional Experiments in the LSI Domain**

We include additional experiments in the LSI domain to assess DQD’s qualitative and quantitative performance on different prompts. The two additional experiments were run with the objective prompts “A photo of Beyonce” and “A photo of Jennifer Lopez” with the measure prompts “A small child.” and “A woman with long blonde hair.”.

Table 6 presents quantitative metrics of the additional runs as well as the Elon Musk experiment from the main paper. We observe that CMA-MEGAd (Adam) retains top quantitative performance against each baseline for each experiment.

Fig. 14 and Fig. 15 contain collages for the Beyoncé and Jennifer Lopez experiments, respectively. For the Jennifer Lopez and Elon Musk experiments, we report that all archive collages were of similar quality. However, for Beyoncé only one of the collages, the one presented in Fig. 14, did not have significant artifacts. CLIP guided CMA-MEGAd (Adam) far out of the latent Gaussian distribution for the remaining runs of the Beyoncé prompt and became confident in low quality images that overwrote
Figure 8: Example archives for each algorithm evaluated in the linear projection domain.

previously discovered high quality images in the archive. We highlight the concern that CLIP may perform differently for text prompts identifying celebrities of different ethnic backgrounds.

For Beyoncé, the collage contains faces of different ages for the measure “A small child”, a proxy prompt for “age”. However, for the Jennifer Lopez prompt, DQD constructs younger faces of Jennifer Lopez by places an older looking face on a younger looking head. We include an additional qualitative run of CMA-MEGA (Adam) on Jennifer Lopez (see Fig. 16), where we replace the measure prompt “A small child.” with the prompt “A frowning person”. In this prompt, we find faces of a younger Jennifer Lopez not discovered in Fig. 15. This result shows that younger faces of Jennifer Lopez can be generated by StyleGAN and identified by CLIP, but not discovered by DQD when the measure function approximated “age”. We hypothesize that this is a limitation of using a proxy measure for age, rather than a predictive model that can directly measure the abstract concept of age.
Arm Repertoire

CMA-MEGA (Adam)
CMA-MEGA
OMG-MEGA
CMA-ME

MAP-Elites
MAP-Elites (line)
OG-MAP-Elites
OG-MAP-Elites (line)

Figure 9: Example archives for each algorithm evaluated in the arm repertoire domain.

Latent Space Illumination

CMA-MEGA (Adam)
CMA-MEGA
CMA-ME
MAP-Elites (line)
MAP-Elites

Figure 10: Example archives for each algorithm in the latent space illumination (LSI) domain.

Figure 11: Percentage of cells in the y-axis with objective values larger than or equal to a threshold specified in the x-axis. The percentage is the number of filled cells (filtered by the threshold) over the archive size. Larger area under each curve indicates better performance.
| Algorithm       | QD-score | Coverage | Best         |
|-----------------|----------|----------|--------------|
| MAP-Elites      | 1.04 ± 0.03 | 1.17 ± 0.04% | 90.60 ± 0.03 |
| MAP-Elites*     | 5.19 ± 0.05 | 5.93 ± 0.06% | 91.39 ± 0.02 |
| MAP-Elites (line) | 12.21 ± 0.06 | 14.32 ± 0.07% | 94.89 ± 0.04 |
| MAP-Elites (line)* | 34.74 ± 0.05 | 41.31 ± 0.06% | 98.83 ± 0.01 |
| CMA-ME          | 1.08 ± 0.03 | 1.21 ± 0.04% | 91.66 ± 0.01 |
| CMA-ME*         | 5.30 ± 0.04 | 6.03 ± 0.05% | 92.01 ± 0.02 |
| OG-MAP-Elites   | 1.52 ± 0.09 | 1.67 ± 0.10% | 100.00 ± 0.00 |
| OG-MAP-Elites (line) | 15.01 ± 0.06 | 17.41 ± 0.08% | 100.00 ± 0.00 |
| CMA-MEGA        | 75.29 ± 0.00 | 100.00 ± 0.00% | 100.00 ± 0.00 |
| CMA-MEGA (Adam) | 75.30 ± 0.00 | 100.00 ± 0.00% | 100.00 ± 0.00 |

Table 5: Results: Values with standard errors of QD-score, coverage, and best solution after 10,000 iterations for each algorithm and domain. Starred algorithms are run for 10^6 iterations.
Figure 12: Images generated with the single-objective Adam optimizer (top) and with CMA-MEGA (bottom) on the StyleGAN+CLIP model. Each algorithm optimizes the objective prompt “Elon Musk with short hair.”

Figure 13: Cherry-picked intermediate images generated from CMA-MEGA (Adam) during the LSI experiments for the prompt “Elon Musk with short hair.” These images were discovered intermediate solutions, but did not end up in the final archive returned by CMA-MEGA (Adam).
### Table 6: Results from additional runs for Jennifer Lopez and Beyoncé. For comparison purposes, we include the original results for Elon Musk in the top table.

| Algorithm            | QD-score | Coverage | Best     |
|----------------------|----------|----------|----------|
| **LSI: Jennifer Lopez** |
| MAP-Elites           | 12.51 ± 0.28 | 19.18 ± 0.48 % | 70.87 ± 0.27 |
| MAP-Elites (line)    | 14.73 ± 0.06 | 21.60 ± 0.08 % | 73.50 ± 0.13 |
| CMA-ME               | 15.24 ± 0.37 | 20.86 ± 0.50 % | 75.39 ± 0.09 |
| CMA-MEGA             | 3.14 ± 0.14  | 5.58 ± 0.13 %  | 64.88 ± 2.89 |
| CMA-MEGA (Adam)      | **17.06 ± 0.10** | **23.40 ± 0.14 %** | **76.02 ± 0.08** |
| **LSI: Beyoncé**     |
| MAP-Elites           | 12.84 ± 0.10 | 19.41 ± 0.16 % | 71.42 ± 0.14 |
| MAP-Elites (line)    | 14.40 ± 0.09 | 21.11 ± 0.16 % | 73.04 ± 0.05 |
| CMA-ME               | 14.00 ± 0.62 | 19.57 ± 0.90 % | 74.11 ± 0.08 |
| CMA-MEGA             | 3.37 ± 0.44  | 6.38 ± 0.71 %  | 59.96 ± 1.40 |
| CMA-MEGA (Adam)      | **16.08 ± 0.37** | **22.58 ± 0.57 %** | **74.95 ± 0.27** |

Figure 14: Result of latent space illumination for objective “A photo of Beyoncé.” and for measures “A small child.” and “A woman with long blonde hair.”. The axes values indicate the score returned by the CLIP model, where lower score indicates a better match.
Figure 15: Result of latent space illumination for objective “A photo of Jennifer Lopez.” and for measures “A small child.” and “A woman with long blonde hair.”. The axes values indicate the score returned by the CLIP model, where lower score indicates a better match.

Figure 16: Result of latent space illumination for objective “A photo of Jennifer Lopez.” and for measures “A frowning person.” and “A woman with long blonde hair.”. The axes values indicate the score returned by the CLIP model, where lower score indicates a better match.