Fully Parallel Hyperparameter Search: Reshaped Space-Filling

M.-L. Cauwet [1] **C. Couprie** [2] **J. Dehos** [3] **P. Luc** [2] **J. Rapin** [2] **M. Riviere** [2] **F. Teytaud** [3] **O. Teytaud** [2] **N. Usunier** [2]

Abstract

Space-filling designs such as Low Discrepancy Sequence (LDS), Latin Hypercube Sampling (LHS) and Jittered Sampling (JS) were proposed for fully parallel hyperparameter search, and were shown to be more effective than random and grid search. We prove that LHS and JS outperform random search only by a constant factor. Consequently, we introduce a new sampling approach based on the *reshaping* of the search distribution, and we show both theoretically and numerically that it leads to significant gains over random search. Two methods are proposed for the reshaping: Recentering (when the distribution of the optimum is known), and Cauchy transformation (when the distribution of the optimum is unknown). The proposed methods are first validated on artificial experiments and simple real-world tests on clustering and Salmon mappings. Then we demonstrate that they bring performance improvement in a wide range of expensive artificial intelligence tasks, namely attend/infer/repeat, video next frame segmentation forecasting and progressive generative adversarial networks.

1. Introduction

One-shot optimization is a critical component of parallel hyperparameter search (Bergstra and Bengio, 2012; Bousquet et al., 2017). It consists in approximating the minimum of a function f by its minimum $\min_{x \in X} f(x)$ over a finite subset $X = \{x_1, \ldots, x_n\}$ of points provided by a *sampler*, i.e., a generator of points sets. A straightforward sampler generates points sets independently and randomly. Despite its simplicity, Bergstra and Bengio (2012) pointed out that such random sampling is robust and outperforms grid sampling. However space-filling designs such as Low Discrepancy Sequences (LDS), Latin Hypercube Sampling (LHS) or Jittered Sampling (JS) are good candidates because they aim at distributing the points more diversely than independent random sampling. In particular, Bousquet et al. (2017) advocated a specific variant of LDS, namely scrambled-Hammersley (Hammersley, 1960; Atanassov, 2004). While LDS performance is well known for numerical integration (Koksma, 1942), their use for one-shot optimization - and a fortiori for hyperparameter tuning - is far less explored. We quantify the benefit of LDS, LHS and JS approaches and, looking for more headroom, propose the concept of *distribution reshaping*, i.e., using a search distribution different from the prior distribution of the optimum.

Space-filling vs reshaping: the two components of one-shot optimization. Let us distinguish two probability distributions: the prior probability distribution P_0 of the optimum and the search probability distribution P_s. When P_0 is known, random sampling boils down to choosing the search distribution $P_s = P_0$ and applying independent sampling. The aforementioned space-filling designs (LDS, LHS, JS), compared to random sampling, relax the assumptions that the samples are independently drawn according to P_s. The state of the art already contains many results in this field: we add some mathematical proofs, and propose a new paradigm, namely *Reshaping*, which goes one step further and relax the condition $P_s = P_0$. This modification originates from Theorem 1 which shows that, surprisingly, *even if the optimum is randomly drawn as a standard normal distribution* in an artificial problem (e.g. we know a priori that the optimum x^* is randomly drawn as a standard Gaussian and the objective function is $x \mapsto \|x - x^*\|^2$), the optimal search distribution *is not that Gaussian distribution*. Hence, inspired by Rahnamayan and Wang (2009); Rahnamayan et al. (2007), we propose reshaped versions using P_s spikier than P_0 around the center: addition of a middle point and, by extension, our Recentering reshaping. On the other hand, the Cauchy reshaping is driven by distinct considerations: it compensates the *corner avoidance property* characterizing some space-filling designs (Owen, 2006; Hartinger et al., 2005) and counterbalances the risk of expert errors on the correct hyperparameter's range, i.e., unknown P_0.

Contributions. We first study the case $P_s = P_0$ in Sec-

[*] Equal contribution [1] ESIEE, Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, F-77454, Marne-la-Vallée, France [2] Facebook AI Research [3] Université du Littoral Côte d'Opale. Correspondence to: M.-L. Cauwet <marie-liesse.cauwet@esiee.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

Table 1: Comparison between non-reshaped sampling methods. Red boxed results are contributions of the paper. The sequences of samples are of size n in dimension d. C_d is a dim-dependent known constant. Results are $O(.)$ unless stated otherwise. Constant factors depend on the d unless d is mentioned.

Sequence	Discrepancy[+]	Incrementality (see SM)	Randomized, PDF>0[†]	Stochastic dispersion[‡]	Stochastic dispersion preserved by projection[*]
LHS	$\Theta(\sqrt{d/n})$ (Doerr et al., 2018)	$\boxed{2n\text{, optimal}}$ (SM)	yes	$\boxed{1/n^{1/d}}$ (Prop. 2)	yes (straightforward)
Grid	$1/n^{1/d}$	no	no	$1/n^{1/d}$ (straightforward)	no (straightforward)
Jittered	$\sqrt{d\log n}/n^{1/2+1/2d}$ (Pausinger et al, 2016)	$\boxed{2^d n\text{, optimal}}$ (SM)	yes	$\boxed{1/n^{1/d}}$ (Prop. 3)	$\boxed{\text{yes}}$ (Prop. 3)
Random	$\log\log(n)^{\frac{1}{2}}/\sqrt{n}$ (Kiefer, 1961)	$+1$	yes	$1/n^{1/d}$ (straightforward)	yes (straightforward)
Halton	$(1+o(1))\times$ $C_d(\log n)^d/n$	$+1$	no	$1/n^{1/d}$ [#]	yes but [#] different constant
Hammersley	$(1+o(1))C_d\times$ $(\log n)^{d-1}/n$	$\boxed{\text{not } n+k\log(n)^{d-\epsilon}}$ (SM) $\epsilon\leq 1, k\leq n-1$	no	$1/n^{1/d}$ [#]	yes but[#] different constant
Scrambled Halton	as Halton, better constant	$+1$	no	$1/n^{1/d}$ [#]	yes but [#] different constant
Scrambled Hammersley	as Hammersley, better constant	$\boxed{\text{not } n+k\log(n)^{d-\epsilon}}$ (SM) $\epsilon\leq 1, k\leq n-1$	no	$1/n^{1/d}$ [#]	yes but [#] different constant
Sobol	as Halton	$+1$	no	$\log(n)/n^{1/d}$	yes but dif. const. [|]
Random -Shift LDS	as original LDS (up to dim-dependent factor)	as original LDS	yes	as original LDS (up to dim-dependent factor)	

[+] The discrepancy of a projection $\Pi(S)$ of a sequence S, when Π is a projection to a subset of indices, is less or equal to the discrepancy of S. Therefore we do not discuss the stability of the sequence in terms of discrepancy of the projection to a subspace, contrarily to what we do for dispersion (for which significant differences can occur, between S and $\Pi(S)$).

[†] "randomized, PDF>0" means that the sampling is randomized with a probability distribution function (averaged over the sample) strictly positive over all the domain.

[‡] Optimal rate is $O(1/n^{1/d})$ (Sukharev, 1971). We do get the dependency in d, namely $O(\sqrt{d}/n^{1/d})$, for random, LHS and JS.

[*] We consider subspaces parallel to axes, i.e. switching to a subset of d' indices. We request that the dependency in d dimension becomes d'.

[|] The bound on the distance to the optimum is an application of the discrepancy, and low discrepancy is preserved by switching to a subspace, hence this positive result.

[#] Constants depend on which variables are in the subspace - first hyperparameters are "more" uniform (Bousquet et al., 2017).

tion 3. We focus on the measure called *stochastic dispersion* (see Section 2) which was identified as a good indicator of the performance of a space-filling design in the context of one-shot optimization methods (Bousquet et al., 2017). We prove some bounds for LHS and JS in terms of stochastic dispersion and additionally we analyze the case of limited number of *critical variables* (see Section 2). Critical variables appear when some of the variables have a strong impact on the objective function whereas the others have a negligible impact. This is evaluated in terms of *stochastic dispersion preserved by projection*. New results are presented together with the state of the art in a more general framework, including 8 other space-filling strategies and 3 other performance indicators, in Table 1. It turns out that these new bounds are actually close to those of random search.

Given this limited headroom, we propose reshaping in Section 4, i.e., changing the search distribution, in two distinct variants. First, even if the prior probability distribution of the optimum is known, we use a search distribution tightened closer to the center of the domain, in a dimension-dependent and budget-dependent manner (Section 4, Recentering). Second - and possibly simultaneously, in spite of the apparent contradiction - we use Cauchy counterparts, which has a heavier tail, for searching closer to the boundaries (for bounded hyperparameters) or farther from the center (for unbounded hyperparameters). We show that P_s different from P_0 is theoretically better in Section 4. Experiments in which the prior P_0 is known (Section 5.1), and then with an unknown prior (Section 5.2), validate this reshaping approach.

2. Samplers: Definition and Evaluation

For simplicity, we always consider $\mathcal{D}=[0,1]^d$ as the search domain and n the number of points in the sequence.

2.1. Space-filling designs

We consider the following sampling strategies (space-filling designs), aimed at being "more uniform" than independent random search.

Grid picks up the middle of each of k^d hypercubes covering the unit hypercube, with k maximum such that $k^d\leq n$. We then sample $n-k^d$ additional random points uniformly in the domain.

Table 2: Artificial objective functions from Nevergrad (see text) with P_0 known: for each combination (dimension $d \in \{3, \ldots, 600\}$, budget $n \in \{30, \ldots, 30000\}$), we mention the name of the method which had the best frequency of outperforming other methods, for that combination, over all objective functions. In this context, Cauchy distributions do not help much. We emphasize in bold the method that most often, over all cells, performed best. O, QO, Rescale, Recentering, etc: see terminology in Section 5.1.2.

	30	100	300	1000	3000	10000	30000	100000	300000
3	Scr Halton	Scr Halton Plus Middle Point	O Rctg1.2 Scr Halton	O Scr Hammersley	Cauchy Rctg.55 Scr Hammersley	Cauchy Rctg.55 Scr Hammersley	Scr Hammersley	O Rctg.7 Scr Halton	Random
18	Scr Halton Plus Middle Point	Scr Halton Plus Middle Point	O Rctg.7 Scr Halton	Scr Halton	Rctg1.2 Scr Hammersley	O Rctg.4 Scr Hammersley	**Meta Rctg**	Cauchy Rctg.55 Scr Hammersley	O Rctg.7 Scr Halton
25	**Meta Rctg**	Rctg.4 Scr Halton	O Rctg.4 Scr Halton	**Meta Rctg**	Rctg.7 Scr Hammersley	Rctg.7 Scr Hammersley	O Rctg.7 Scr Halton	O Rctg.7 Scr Hammersley	Rctg.7 Scr Halton
100	Q O Rctg.4 Scr Hammersley	**Meta Rctg**	Rctg.4 Scr Halton	**Meta Rctg**	Rctg.4 Scr Hammersley	O Rctg.7 Scr Halton	Rctg.4 Scr Halton	O Rctg.4 Scr Halton	Rctg.4 Scr Hammersley
150	O Rctg.4 Scr Hammersley	Q O Rctg.4 Scr Hammersley	**Meta Rctg**	O Rctg.4 Scr Hammersley	Q O Rctg.7 Scr Hammersley	Rctg.4 Scr Halton	O Rctg.7 Scr Hammersley	**Meta Rctg**	**Meta Rctg**
600	Rctg.4 Scr Halton	Rctg.4 Scr Hammersley	O Rctg.4 Scr Hammersley	**Meta Rctg**	**Meta Rctg**	Rctg.4 Scr Halton	Rctg.4 Scr Hammersley	**Meta Rctg**	Rctg.4 Scr Halton

Latin Hypercube Sampling (LHS) Eglajs and Audze (1977); McKay et al. (1979) defines $\sigma_1, \ldots, \sigma_d$, random independent permutations of $\{0, \ldots, n\}$ and then for $0 \leq i \leq n$, the j^{th} coordinate of the i^{th} point of the sequence is given by $(x_i)_j = (\sigma_j(i) + r_{i,j})/(n+1)$ where the $r_{i,j}$ are independent identically distributed, uniformly in $[0, 1]$.

Jittered sampling consists in splitting $\mathcal{D}$ into $n = k^d$ (assuming that such a k exists) hypercubes of volume $1/k^d$ and drawing one random point uniformly in each of these hypercubes (Pausinger and Steinerberger, 2016). Other forms of jittered sampling exist, e.g., with different numbers of points per axis.

Low Discrepancy Sequences (LDS) are also called *quasi-random sequences*. This is the case for *Halton*, *Hammersley* and *Sobol* sequences. **Halton** (Halton, 1960) defines the j^{th} coordinate of the i^{th} point of the sequence as $(x_i)_j = radixInverse(i, p_j)$ where: (1) $p_0, \ldots, p_{d-1}$ are coprime numbers. Typically, but not necessarily, p_i is the $(i+1)^{th}$ prime number. (2) $radixInverse(k, p) = \sum_{j\geq 0} a_j p^{-j}$ with $(a_j)_{j\geq 0}$ being the writing of k in basis p, i.e., $k = \sum_{j\geq 0} a_j p^j$. **Hammersley's** sequence is given by $(x_i)_j = radixInverse(i, p_{j-1})$ when $j > 0$ and $(x_i)_0 = \frac{i+\frac{1}{2}}{n}$, see (Hammersley, 1960). **Sobol**'s sequence (Sobol, 1967) is another advanced quasi-random sequence.

We use various modifiers of the samplers, detailed in supplementary material (SM). **Random shift** consists in adding a random vector in the unit hypercube and applying modulo 1 (Tuffin, 1996). It is applicable to all samplers. **Scrambling** (Owen, 1995; Tuffin, 1998; Atanassov, 2004) affects only Halton and Hammersley sequences. Other modifications are based on reshaping (Section 4).

2.2. Performance measures and proxies.

Given a domain $\mathcal{D}$, the stochastic dispersion (Bousquet et al., 2017) of a random variable $X = (x_1, \ldots, x_n)$ in $\mathcal{D}^n$ is defined as

$$\text{sdisp}(X) = \sup_{x^* \in D} \mathbb{E}_X \inf_{1\leq i\leq n} \|x_i - x^*\|,$$

where $\|\cdot\|$ is the Euclidean norm.

The stochastic dispersion is related to an optimization criterion, namely the simple regret (Bubeck et al., 2009). More precisely, it corresponds to the simple regret, for the worst case over $x^* \in [0, 1]^d$, of the one-shot optimization of a function equal to $x \mapsto \|x - x^*\|$ with a given sampling method.

Critical vs useless variables. We consider the case in which there are d' unknown variables (termed critical variables) with an impact on the objective function whereas $d - d'$ variables (the useless variables) have no impact.

3. Are Space-Filling Designs All Equal ?

In this section, we work within the framework of a search distribution P_s equal to the prior distribution P_0. We show that LHS and JS can only improve results by a constant factor, even in the case of critical variables: this is the key reason for introducing reshaping in Section 4. See the supplementary material for proofs.

Let us first mention Property 1, a known property of random search that enables us to compare the stochastic dispersion of random search with the one of LHS and JS.

Property 1. *Consider x^* an arbitrary point in $[0,1]^d$. Consider $x_1,\dots,x_n$ a sequence generated by random search. Then, for any $\delta \in]0,1[$,*

$$\mathbb{P}\left(\min_i \|x_i - x^*\| \le \epsilon_{d,n,\delta}\right) \ge 1-\delta$$

where $\epsilon_{d,n,\delta} = O\left(\frac{\log(1/\delta)^{1/d}}{(nV_d)^{1/d}}\right)$ and V_d is the volume of $\{x \in \mathbb{R}^d ; \|x\| \le 1\}$.

3.1. Stochastic dispersion of LHS and JS

We now establish a similar bound for LHS and JS in Properties 2 and 3.

Property 2. *Consider x^* an arbitrary point in $[0,1]^d$ and $x_1,\dots,x_n$ a sample generated via LHS. Then for any $\delta \in]0,1[$,*

$$\mathbb{P}\left(\min_i \|x_i - x^*\| \le \epsilon_{d,n,\delta}\right) \ge 1-\delta$$

where $\epsilon_{d,n,\delta} = O\left(\sqrt{d}\log(1/\delta)^{1/d}/n^{1/d}\right)$.

This shows that the stochastic dispersion of LHS is optimal, i.e., it decreases at the optimal rate $O(\sqrt{d}/n^{1/d})$ also guaranteed by Sukharev grids (Sukharev, 1971). This is the same as random search.

Property 3 (Jittered sampling has optimal dispersion after projection on a subset of axes)**.** *Consider x^* an arbitrary point in $[0,1]^{d'}$. Consider jittered sampling with $n = k^d$ points. Consider its projection on the $d' \le d$ first coordinates[1]. Let $x_1,\dots,x_n$ be these projected points. Then, for any $\delta \in]0,1[$,*

$$\mathbb{P}\left(\min_i \|x_i - x^*\| \le 2^{1+1/d'}\frac{\log(1/\delta)^{1/d'}}{(V_{d'}n)^{1/d'}}\right) \ge 1-\delta$$

where V_d is as defined in Property 1.

Hence Jittered Sampling benefits from the same stochastic dispersion bounds as random search and LHS, namely $O\left(\sqrt{d'}\frac{\log(1/\delta)^{1/d'}}{n^{1/d'}}\right)$ - and for all those methods, this is valid for a restriction to d' variables as, contrarily to low-discrepancy sequences, the restriction to a subset does not penalize the stochastic dispersion.

Corollary 1. *Prop. 1, 2 and 3 give the following bounds on the stochastic dispersion of LHS, JS and random search for a restriction to d' variables of a sampling in $[0,1]^d$: $O\left(\sqrt{d'}\frac{\sqrt[d']{\log(1/\delta)}}{n^{1/d'}}\right)$.*

3.2. Space-filling designs: state of the art

These results are to be compared with the state of the art presented in Table 1. In particular, every sampler but one (Sobol) is known optimal in terms of stochastic dispersion. Furthermore, LHS and JS are entirely preserved by projection to any subspace - as opposed to low discrepancy sequences, for which only segments of initial variables keep the same constants in discrepancy or dispersion bounds. We note that Table 1 also present bounds on discrepancy and incrementality results: these performance criteria are not directly related to our experimental framework. Nonetheless, they are suitable in other contexts, so we present them for the sake of exhaustive comparison between these samplers (see SM, Section 2).

4. Theory of Reshaping: Middle Point & Recentering

Interestingly, in the context of the initialization of differential evolution (Storn and Price, 1997), Rahnamayan and Wang (2009) proposed a sampling focusing on the middle. Their method consists in adding, for each point x sampled in the domain $[-1,1]^d$, a point $-r \times x$ for r uniformly drawn in $[0,1]$. This combines **opposite sampling** (which corresponds to the -1 factor, also used for population-based optimization in Teytaud et al. (2006)) and focus to the center (multiplication by a constant < 1): their method is termed **quasiopposite** sampling. Consider dimension d and x^* randomly normally distributed with unit variance in $\mathbb{R}^d$. Consider $x_1,\dots,x_n$, independently randomly normally distributed with unit variance in $\mathbb{R}^d$. The median of $\|x^*\|^2$ is denoted m_0 and the median of $\min_{i\le n}\|x_i - x^*\|^2$ is denoted m_n. We first note Lemmas 1 and 2.

Lemma 1. *m_0 is equivalent to d as $d \to \infty$.*

Lemma 2. *$P(\|x_1 - x^*\|^2 \le m_n) \ge \frac{1}{2n}$.*

We now note that $\left\|\frac{1}{\sqrt{2}}(x_1 - x^*)\right\|^2$ follows a χ^2 distribution with d degrees of freedom.

Lemma 3. *By Chernoff's bound for the χ^2 distribution, $P\left(\|x_1 - x^*\|^2 \le d(1+o(1))\right) \le \left((1+o(1))\frac{1}{2}\exp(\frac{1}{2})\right)^{d/2}$.*

Theorem 1. *Consider $n > 0$. There exists d_0 such that for all $d > d_0$, if X is a random sample of n independent standard normal points in dimension d, if x^* is a random independent normal point in dimension d, then, unless $n \ge \frac{1}{2}\left((1+o(1))\frac{1}{2}\exp(\frac{1}{2})\right)^{-d/2}$, the median of the minimum distance $\min_{x\in X}\|x - x^*\|$ is greater than the median of the distance $\|x^* - (0,\dots,0)\|$.*

This shows the surprising fact that a single point, in the middle, outperforms the best of n randomly drawn points.

[1] Without loss of generality; we might consider a projection to d' arbitrarily chosen coordinates.

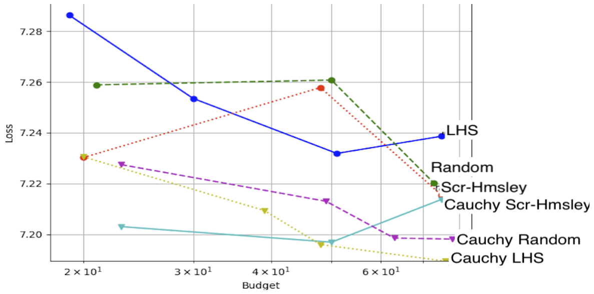

Figure 1: Cauchy vs Gauss on the Attend/Infer/Repeat model. For each setting in {random, LHS, ScrambledHammersley}, we reach better loss values when switching to the Cauchy counterpart. The P-value 0.05 validates Cauchy vs Normal. See SM, Section 3.4, for validation in terms of counting objects.

If we consider a fixed budget n, having n points equal to 0 is pointless. So we can consider $n-1$ standard independent normal points, plus a single middle point: this is our modification "plus middle point". Later, we will also propose another related reshaping, namely tightening the distribution closer to the center: the Recentering method. Then we will see a distinct reshaping, namely switching to the Cauchy distribution.

Recentering (Rctg for short) reshaping. Consider optimization in $[0,1]^d$. Given a sample S, and using g the cumulative distribution function of the standard Gaussian, the Rctg reshaping consists in concentrating the distribution towards $(0.5,\dots,0.5)$: it considers $\{c(s) \mid s \in S\}$ rather than S, with

$$c(s) = g(\lambda \times g^{-1}(s_1)),\dots,g(\lambda \times g^{-1}(s_d)). \quad (1)$$

$\lambda = 0$ sets all points to the middle of the unit hypercube. $\lambda = 1$ means no reshaping. **With MetaRecentering, we consider specific values of λ.** Preliminary experimental results lead to the specification of MetaRctg, using the dimension and the budget for choosing the parameter λ of the Recentering reshaping as follows. MetaRctg uses Scrambled-Hammersley and Rctg reshaping with

$$\lambda = \frac{1+\log(n)}{4\log(d)}. \quad (2)$$

Cauchy. When using the Cauchy distribution, we get $c(s) = g(\lambda \times C^{-1}(s_1)),\dots,g(\lambda \times C^{-1}(s_d))$ with C the Cauchy cumulative distribution function. **Extension to unbounded hyperparameters.** $g(.)$ can be removed from those equations when we consider sampling in $\mathbb{R}^d$ rather than in $[0,1]^d$: we then get $c(s) = \lambda \times g^{-1}(s_1),\dots,\lambda \times g^{-1}(s_d)$ in the normal case. g can also be applied selectively on some variables and not on others when we have both bounded and unbounded hyperparameters.

5. Experiments

See SM for reproduction of all our artificial experiments with one-liners in the Nevergrad platform (Rapin and Teytaud, 2018), for the Nevergrad "oneshot" experiment (SM, Section 4, or the website (Rapin and Teytaud, 2018)) which shows a clear success of MetaRecentering, and for additional deep learning experiments with F_2F_5 (Luc et al., 2018) (SM, Section 3.5). We first test our baselines in an artificial setting with P_0 known and without any reshaping (Section 5.1.1). We then check that Recentering works in such a context of P_0 known (Section 5.1.2). Then we switch to P_0 unknown (Section 5.2). We evaluate the impact of Cauchy (Section 5.2.1). We then combine Recentering and Cauchy (Section 5.2.2 and 5.2.3). We conclude that Cauchy is validated for unknown P_0 and that Recentering works if P_0 is known *or, to some extent, if the underlying data are properly rescaled (Table 3, as opposed to the wildly unscaled problems in Table 4).*

5.1. When the prior P_0 is known

There are real world cases in which P_0 is known, typically when many optimization runs are performed in a row:

- in maximum likelihood maximization for item response theory repeated for estimating the parameters of many questions (Jia et al., 2019),
- in the ELO evaluation of many gamers from their records,
- in repeated hyperparametrization of cloud-based machine learning (Allaire, 2018),
- in repeated optimization of industrial oven parameters (Cavazzuti et al., 2013) for distinct scope statements.

Another case is when the objective function is the worst outcome over a family of scenarios, to be approximated by a finite sample corresponding to ranges of independent exogenous variables (dozens of annual weather parameters and financial parameters), as usually done in network expansion planning (Escobar et al., 2008; Li et al., 2016).

5.1.1. P_0 KNOWN AND $P_s = P_0$: CONFIRMING THE STATE OF THE ART

While we use mainly real-world experiments in the present paper, we draw the following conclusions from synthetic experiments with controlled P_0 and without reshaping, using classical objective functions from the derivative-free literature, various budgets and all samplers defined above. The detailed setup and results are reported in SM, Section 3.2; they confirm results in Bergstra and Bengio (2012); Bousquet et al. (2017) as follows:

- Many low discrepancy methods (e.g. Halton, Hammersley and their scrambled counterparts) depend on the

Table 3: Experiments on the Nevergrad real-world rescaled testbed with $d \in \{10, \ldots, 120\}$ and $n \in \{25, \ldots, 12800\}$. This experiment termed "oneshotscaledrealworld" corresponds to real-world test cases in which a reasonable effort for rescaling problems according to human expertise has been done; P_0 is not known, but an effort has been made for rescaling problems as far as it is easily possible. Dimension from 10 to 120, budget from 25 to 12800. There is no prior knowledge on the position of the optimum in this setting; MetaRctg did not perform bad overall but Cauchy variants dominate many cases, as well as rescaled versions which sample close to boundaries. Hmsley stands for Hammersley, Cchy for Cauchy, Rctg for Recentering. We emphasize in bold the method performing best overall; variants of CauchyRctg often performed best - most often with a constant < 1, i.e. with a recentering peaked towards the center. All samplers are tested in both variants, normal and Cauchy. This validates both Recentering and Cauchy, but Eq. 2 ("Meta" choice of λ) was moderately confirmed here. O, QO, Rescale, Recentering, etc: see terminology in Section 5.1.2.

	25	50	100	200	400	800	1600	3200	6400	12800
10	Hmsley Plus Middle Point	Scr Hmsley Plus Middle Point	Scr Hmsley	Rctg.7 Scr Halton	Halton	Meta Rctg	Random Plus Middle Point	L H S	Halton	Halton Plus Middle Point
20	Scr Hmsley	Scr Halton Plus Middle Point	Scr Halton	Scr Halton	Halton Plus Middle Point	L H S	O Random	L H S	Hmsley	Hmsley
30	Rescale Scr Hmsley	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	Meta Cchy Rctg	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	Cchy Rctg.4 Scr Hmsley
40	Scr Hmsley	Scr Hmsley Plus Middle Point	Scr Halton Plus Middle Point	Scr Hmsley Plus Middle Point	Scr Halton Plus Middle Point	Random	Scr Hmsley	L H S	Random Plus Middle Point	Scr Hmsley Plus Middle Point
60	Rescale Scr Hmsley	**Cchy Rctg.55 Scr Hmsley**	Cchy Rctg.7 Scr Hmsley	**Cchy Rctg.55 Scr Hmsley**	Cchy Rctg.7 Scr Hmsley	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	Meta Cchy Rctg	**Cchy Rctg.55 Scr Hmsley**	Cchy Rctg.4 Scr Hmsley
120	Rescale Scr Hmsley	Cchy Rctg.7 Scr Hmsley	Cchy Rctg.7 Scr Hmsley	Cchy Rctg.7 Scr Hmsley	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**	**Cchy Rctg.55 Scr Hmsley**

order of hyperparameters - intuitively there is "more" low discrepancy for the first variables: our experiments in optimization confirm that low discrepancy methods are better when variables with greater impact on the objective function are first. With scrambled low-discrepancy methods, results are less penalized (but still penalized) when important variables are last.

- LHS is robust to the order of variables and, therefore, performs well for a small number of randomly positioned critical variables.

Besides confirming the state of the art, these experiments also confirm that adding a single middle point helps, in particular in high dimension. Among methods using $P_s = P_0$, Scrambled-Hammersley plus middle point is one of the best methods in this simplified setting (though one can point out that adding a middle point is not exactly $P_s = P_0$, so this is a first step towards reshaping).

5.1.2. Reshaped space-filling design: Recentering works

We present an experiment on objective functions Sphere, Rastrigin and Cigar, with 100% or 16.67% of critical variables (i.e. in the latter case we add 5 randomly positioned useless variables for each critical variable), with budget in $\{30, 100, 300, 1000, 3000, 10000, 30000, 100000, 300000\}$, with 3, 25 or 100 critical variables. We compare Rctg reshaping with constant $\lambda \in \{0.01, 0.1, 0.4, 0.55, 1.0, 1.2, 2.0\}$, the same Rctg reshaping plus opposite sampling or quasi-opposite sampling, on top of LHS, Scrambled-Halton, Scrambled-Hammersley or pure random sampling; we consider the inverse cumulative distribution functions of Gauss or Cauchy as conversion tools from $[0,1]^d$ to $\mathbb{R}^d$as discussed above. Given the different contexts regarding the number of critical/useless variables previously mentioned, we get dimension 3, 18, 25, 100, 150, 600. The context is still P_0 perfectly known (standard Gaussian). This is reproducible with a one-liner "oneshotcalais" in Nevergrad (see SM). Results are presented in Table 2 and validate Recentering, in particular in its Meta version (Eq. 2), as soon as P_0 is known. We have here 7400 replicas per run. We use several modifiers for the "original" sampling methods (original in the sense before consider $P_s \neq P_0$): (i) O refers to opposite; (ii) QO refers to quasiopposite; (iii) Rescale refers to linearly rescaling to the boundaries: given points $x_1, \ldots, x_i, \ldots, x_n$ in $\mathbb{R}^d$ with coordinates $x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,d}$, we define a Rescale counterpart by $x'_{i,j} = (x'_{i,j} - m_j)/(M_j - m_j)$ with $m_j = \min_i x_{i,j}$ and $M_j = \max_j x_{i,j}$; (iv) Plus Middle Point refers to sampling $n-1$ points instead of n and adding a middle point; (v) Recentering (Rctg) refers to Eq. 1 (Eq. 2 for MetaRecentering). They can all be applied on top of other methods. Recentering has a parameter λ: RctgX refers to $\lambda = X$. For example, Rctg1.2-ScrHammersley refers to Hammersley, equipped with scrambling and recentering with parameter $\lambda = 1.2$. MetaRctg refers to Scrambled

Table 4: Counterpart of Table 3 with the experiment termed "oneshotunscaledrealworld" in Nevergrad, which contains more problems, including many for which no rescaling effort has been made, with $d \in \{10, \dots, 675\}$ and $n \in \{25, \dots, 12800\}$. Rctg does not make sense in this "totally unknown P_0" setting, but we still see a lot of Cauchy counterparts or Rescaled versions, showing that focusing close to boundaries (for bounded hyperparameters) or on large values (for unbounded hyperparameters) makes sense. The contrast with Table 3 in which CauchyMetaRctg and close variants such as CauchyRctg with $\lambda = .55$ perform well suggest the unsurprising fact that scaling parameters and data is a good idea for generic methods, such as MetaRctng, to be effective. In bold, the two methods performing best most often in the present totally unscaled setting.

	25	50	100	200	400	800	1600	3200	6400	12800
10	Scr Halton Plus Middle Point	**Rescale Scr Hmsly**	Scr Hmsly	**Rescale Scr Hmsly**	Meta Rctg	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	Scr Hmsly	L H S	Scr Hmsly Plus Middle Point
15	Cchy Scr Hmsly	Cchy Random	**Cchy LHS**	**Cchy LHS**	Rctg20 Scr Halton	Cchy Scr Hmsly	Cchy Scr Hmsly	Cchy Rctg12 Scr Hmsly	Cchy Scr Hmsly	Cchy Rctg12 Scr Hmsly
20	L H S	Random	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	Hmsly Plus Middle Point
30	Cchy Random	**Cchy LHS**	**Cchy LHS**	Cchy Random	Cchy Random	**Cchy LHS**	Cchy Random	**Cchy LHS**	Cchy Random	Cchy Random
40	Rctg12 Scr Halton	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	**Rescale Scr Hmsly**	O Random	**Rescale Scr Hmsly**	Random
60	**Cchy LHS**	Cchy Random	**Cchy LHS**	Cchy Rctg7 Scr Hmsly	**Cchy LHS**	Cchy Rctg7 Scr Hmsly	Cchy Random	**Cchy LHS**	**Cchy LHS**	Cchy Random
120	**Cchy LHS**	**Cchy LHS**	**Cchy LHS**	Cchy Random	**Cchy LHS**	Cchy Random	Cchy Random	Cchy Random	Cchy Random	**Cchy LHS**
675	Cchy Rctg12 Scr Hmsly	Cchy Rctg4 Scr Hmsly	Cchy Scr Hmsly	Cchy Random	Cchy Rctg7 Scr Hmsly	Cchy Rctg4 Scr Hmsly	Cchy Rctg4 Scr Hmsly	O Rctg20 Scr Halton	Cchy Scr Hmsly	Meta Cchy Rctg

Hammersley, equipped with recentering with λ chosen by Eq. 2. We note that QO, with the multiplication by a random constant < 1, can be seen as further reducing the constant. Overall the Rctg reshaping outperforms its ancestor the quasiopposite sampling (Rahnamayan and Wang, 2009). MetaRctg turns out to be one of the best methods overall (see also SM, Section 4), performing close to the best for each budget/dimension in the present context of P_0 known. These experiments use the artificial Nevergrad *oneshot* experiment, for which the distribution of the optimum is a standardized multivariate normal distribution. They also show that in such a context (P_0 perfectly known) Cauchy is not useful. The next section will investigate the case of P_0 unknown.

5.2. Cauchy-reshaping with P_0 unknown

5.2.1. No P_0: Cauchy for attend/infer/repeat

Previous results have validated Recentering (our first proposed reshaping) in the case of P_0 known. The second form of reshaping consists in using the Cauchy distribution when P_0 is unknown: we here validate it. As detailed in Section 4, Cauchy makes sense also without recentering and for bounded hyperparameters (in that case, it increases the density close to the boundaries). Fig. 1 presents results on the Attend, Infer, Repeat (AIR) image generation testbed (Eslami et al., 2016). AIR is a special kind of variational autoencoder, which uses a recurrent inference network to infer both the number of objects and a latent representation for each object in the scene. We use a variant that additionally models background and occlusions: details are provided in SM (Section 3.4). We have 12 parameters (initially tuned by a human expert), namely the learning rates of the main network and of the baseline network, the value used for gradient clipping, the number of feature maps, the dimension of each latent representation, the variance of the likelihood, the variance of the prior distribution on the scale of objects, and the initial and final probability parameter of the prior distribution on the number of objects present. The loss function is the Variational Lower Bound, expressed in bits per dimension. The dataset consists in 50000 images from Cifar10 (Krizhevsky et al., 2010) and 50000 object-free patches from COCO (Lin et al., 2014), split into balanced training (80% of the samples) and validation sets. For each space-filling method, the Cauchy counterpart outperforms the original one.

5.2.2. Generative adversarial networks (GANs).

We use Pytorch GAN Zoo (Riviere, 2019) for progressive GANs (Karras et al., 2018), with a short 10 minutes training on a single GPU. We optimize 3 continuous hyperparameters, namely leakiness of Relu units in $[1e - 2, 0.6]$, the discriminator ϵ parameter in $[1e - 5, 1e - 1]$, and the base learning rate in $[1e-5, 1e-1]$. MetaCauchyRctg performed best (Fig. 2), in spite of the fact that it was designed just by adding Cauchy to MetaRctg which was designed on

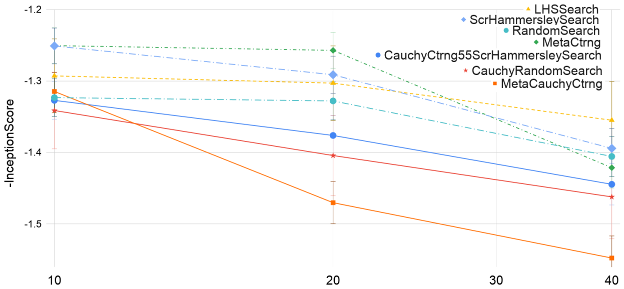

Figure 2: Experiments on progressive GANs(Karras et al., 2018; Riviere, 2019); see a discussion of criteria in Borji (2018). MetaCauchyRctg dominates. All Cauchy variants (full lines) perform well compared to normal ones (dashed). The "Meta" choice of the recentering constant (Eq. 2) seemingly performs well though without Cauchy and low budgets this was not that clear. X-axis = budget. Y-axis = opposite of inception score, the lower the better (we use opposite for consistency with other plots in the present paper, which consider losses to be minimized).

independent distinct artificial experiments in Tab. 2.

5.2.3. Nevergrad real world experiments without control of P_0

Nevergrad provides a real world family of experiments, built on top of the MLDA testbed (Gallagher and Saleem, 2018). This family uses Salmon mappings, clustering, the tuning of game policies, a few traveling salesman problems, some electricity production scheduling problems, and others: these experiments are cheaper and fully reproducible. We run all our samplers with 7400 repetitions. We then check, for each budget/dimension, which method performed best (Tab. 3): in this real-world context, we can not assume a standard deviation of 1 for the uncertainty over the optimum. The Cauchy sampling with Rctg reshaping 0.55 performs well. It is sometimes outperformed by rescaled Scrambled-Hammersley, which takes care of pushing points to the frontier: each variable is rescaled so that the min and the max of each variable, over the sample, hit the boundaries. This means that the two best methods can both sample far from the center: either heavy tail for Cauchy, or sample rescaled for matching the boundaries. The exact optimal constant λ does not always match the MetaRecentering scaling proposed in Eq. 2, but is clearly < 1. This setting does not have a known P_0 (just a rough rescaling of underlying data). Still, we now switch to a more challenging setting in Table 4: we consider the "oneshotunscaledrealworld" experiment in Nevergrad for a case in which no rescaling effort was made; we still get a quite good behavior of Cauchy or rescaled variants, but (consistently with intuition) Recentering does not make sense anymore. As requested by reviewers, we provide results showing the gap between methods (as opposed to only frequencies of method A outperforming method B):

- In Fig. 4, the case $P_s = P_0$, namely budget $\in \{30, 100, 3000, 10000, 30000, 100000\}$, with function in $\{cigar, sphere, rastrigin\}$ and in dimension $\in \{2000, 20000\}$: we see that MetaCentering performs quite well in this context.
- In Fig. 3, the gap between methods in the unscaled real-world cases: we see that Cauchy or Rescale (rescaling to the boundaries) is the key component explaining the success of a method.

Context	Recommendation	XPs
P_0 perfectly known	MetaRctg	Table 2 SM Sec. 4
P_0 approx. known (rescaled data, normalized params)	MetaCchyRctg or CchyRctg.55	Table 3 Fig 2.
P_0 wildly unknown (avoid this!)	Cchy-LHS or Rsc-Scr-Hmsly	Table 4, Fig. 1, 3.

Table 5: Practical recommendations.

6. Conclusion

Theoretical results. We showed that LHS and jittered sampling have, up to a constant factor, the same stochastic dispersion rate as random sampling (Sections 3), including when restricted to a subset of variables. Low-discrepancy sequences have good stochastic dispersion rates as well. However, their performance for a subset of variables depends on the location of those variables - the earlier the better. Typically they are outperformed by LHS when there is a single randomly positioned critical variable. Overall, the benefit of sophisticated space-filling designs turn out to be moderate, compared to random search or LHS, hence the motivation for alternative fully parallel hyperparameter search ideas such as reshaping. We then showed that adding a middle point helps in many high-dimensional cases (Section 4). This element inspired the design of search distributions different from the Gaussian one, such as Cauchy and/or reshaping as in Rctg methods (Section 4) - which provide substantial improvements.

Practical recommendations. Table 5 surveys our practical conclusions. Our experiments suggest to **use MetaRctg (Scrambled-Hammersley + Rctg reshaping with λ as in Eq. 2) when we have a prior on the probability distribution of the optimum** (Tab. 2 and SM, Section 4). Precisely, with a standard normal prior P_0 (use copulas, i.e. multidimensional cumulative distribution functions, for other probability distributions), use

$$x_{i,j} = g\left(\frac{1+\log(n)}{4\log(d)} g^{-1}(ScrH_{i,j})\right)$$

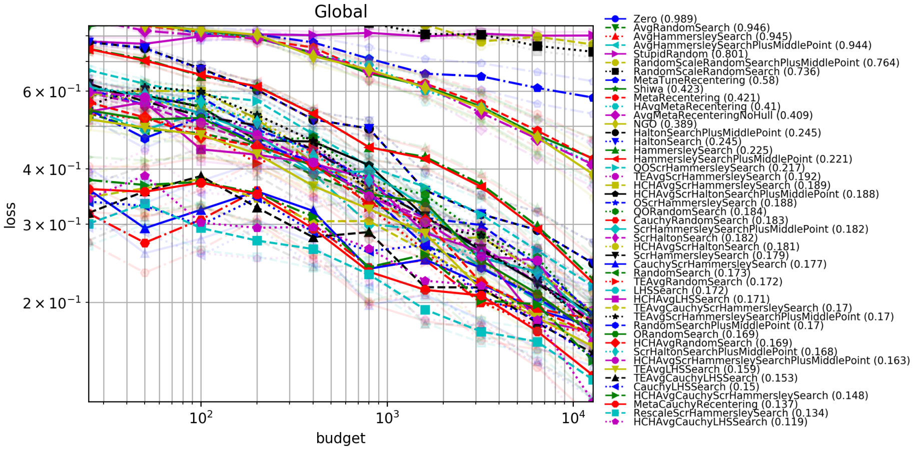

Figure 3: Representation of the Nevergrad experiment "oneshotunscaledrealworld". For each algorithm, each budget, each context (the context includes all information representing the test case, i.e. including dimension and objective function) we consider the average loss. Then, this loss is replaced by its rank (between 0 and 1) among all losses for all optimizers and all budget values of the experiment, for this same context. Then we plot, for each optimization algorithm and each budget, the average rank over all contexts. This is therefore a regret, averaged (after representation by rank) over all test cases. All methods which performed well use either "Rescale" or Cauchy. Methods: all methods here are publicly available in the Nevergrad platform: Zero just returns $(0, \ldots, 0)$, methods with Avg in the name use sophisticated recommendation methods (i.e. they might recommend a point which was not in the sample). Rescale Scrambled Hammersley and MetaCauchyRecentering are the two best overall (though for the maximum budget a method using a recommendation not in the sample $x_1, \ldots, x_n$ performed slightly better).

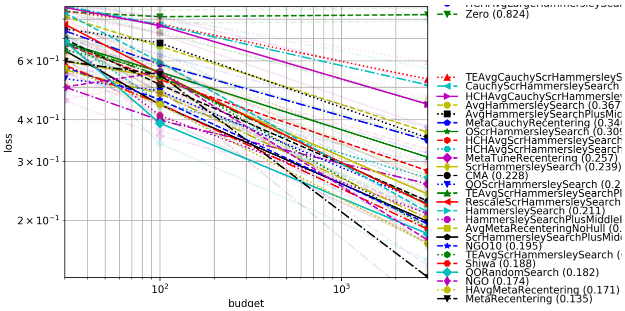

Figure 4: Representation by rank / averaging of the DOE experiment in Nevergrad (same method of quantiles/ranks and averaging as in Fig. 3): in the present context with known P_0, our MetaRecentering performed best for the maximum budget and among the best for various levels of the budget. We acknowledge that the mathematically derived MetaTuneRecentering, similar to MetaRecentering but with better parametrization, does perform better in particular for large budgets and moderate dimension(Meunier et al., 2020).

for searching in $[0, 1]^d$ with budget n, where $ScrH_{i,j}$ is the j^{th} coordinate of the i^{th} point in the Scrambled Hammersley sequence and g the Gaussian cumulative distribution function. In many cases, however, we do not have such a prior on the position of the optimum: experts provide a correct range of values for most variables, but miss a few ones. For those missed ones, the best $x_{i,j}$ values are extreme. Unfortunately, low-discrepancy methods are weak, by a dimension-dependent factor, for searching close to boundaries - this is known as the corner avoidance effect (Owen, 2006; Hartinger et al., 2005). Therefore low-discrepancy methods can then perform worse than random. In such a case, Cauchy helps. A second suggestion is therefore **to use Cauchy in real world cases**: we got in Tabs 3 and 4 or for AIR (Fig. 1) or Pytorch-GAN-zoo (Fig. 2), i.e. all our real-world experiments, better performances with Cauchy. More specifically, **for real-world problems, rescaling data and parameters and using CauchyMetaRctg looks like the best solution** - with a minimum of standardization, we got good results for CauchyMetaRctg in Fig. 2 (just using human expertise for rescaling) and Tab. 3 (just based on standardizing underlying data, so no real known prior P_0). When a proper scaling is impossible, we still get good performance for Cauchy variables but λ is impossible to guess so that the best method varies from one case to another (Table 4) - Cauchy-LHS and Rescale-Scr-Hammersley being the most stable.

References

Allaire, J. (2018). Tensorflow for R: R interface to Google CloudML.

Atanassov, E. I. (2004). On the discrepancy of the Halton sequences. *Math. Balkanica (NS)*, 18(1-2):15–32.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Borji, A. (2018). Pros and cons of gan evaluation measures.

Bousquet, O., Gelly, S., Karol, K., Teytaud, O., and Vincent, D. (2017). Critical hyper-parameters: No random, no cry. *CoRR*, abs/1706.03200.

Bubeck, S., Munos, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer.

Cavazzuti, M., Corticelli, M., Nuccio, A., and Zauli, B. (2013). Cfd analysis of a syngas-fired burner for ceramic industrial roller kiln. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227:2600–2609.

Doerr, B., Doerr, C., and Gnewuch, M. (2018). *Probabilistic Lower Bounds for the Discrepancy of Latin Hypercube Samples*, pages 339–350. Springer International Publishing, Cham.

Eglajs, V. and Audze, P. (1977). New approach to the design of multifactor experiments. *Problems of Dynamics and Strengths*, 35:104–107.

Escobar, A. H., Romero, R. A., and Gallego, R. A. (2008). Transmission network expansion planning considering multiple generation scenarios. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, pages 1–6.

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3233–3241, USA.

Gallagher, M. and Saleem, S. (2018). Exploratory landscape analysis of the mlda problem set. In *PPSN'18 workshop*.

Halton, J. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90.

Hammersley, J. M. (1960). Monte Carlo methods for solving multivariate problems. *Annals of the New York Academy of Sciences*, 86(3):844–874.

Hartinger, J., Kainhofer, R., and Ziegler, V. (2005). On the corner avoidance properties of various low-discrepancy sequences. *INTEGERS: Electronic Journal of Combinatorial Number Theory*, pages 1–16.

Jia, B., Zhang, X., and Zhu, Z. (2019). A short note on aberrant responses bias in item response theory. *Frontiers in Psychology*, 10:43.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of ICLR*.

Kiefer, J. (1961). On large deviations of the empiric d. f. of vector chance variables and a law of the iterated logarithm. *Pacific J. Math.*, 11(2):649–660.

Koksma, J. F. (1942). Een algemeene stelling inuit de theorie der gelijkmatige verdeeling modulo 1. *Mathematica (Zutphen)*, 11:7–11.

Krizhevsky, A., Nair, V., and Hinton, G. (2010). Cifar-10 (canadian institute for advanced research).

Li, J., Ye, L., Zeng, Y., and Wei, H. (2016). A scenario-based robust transmission network expansion planning method for consideration of wind power uncertainties. *CSEE Journal of Power and Energy Systems*, 2(1):11–18.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 740–755.

Luc, P., Couprie, C., LeCun, Y., and Verbeek, J. (2018). Predicting future instance segmentation by forecasting convolutional features. *Proc. of European Conference on Computer Vision (ECCV)*, pages 593–608.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Meunier, L., Doerr, C., Rapin, J., and Teytaud, O. (2020). Variance reduction for better sampling in continuous domains.

Owen, A. B. (1995). Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and quasi-Monte Carlo methods in scientific computing*, pages 299–317. Springer.

Owen, A. B. (2006). Halton sequences avoid the origin. *SIAM Review*, 48:487–503.

Pausinger, F. and Steinerberger, S. (2016). On the discrepancy of jittered sampling. *Journal of Complexity*, 33:199–216.

Rahnamayan, S., Tizhoosh, H. R., and Salama, M. M. A. (2007). Quasi-oppositional differential evolution. In *2007 IEEE Congress on Evolutionary Computation*, pages 2229–2236.

Rahnamayan, S. and Wang, G. G. (2009). Center-based sampling for population-based algorithms. In *2009 IEEE Congress on Evolutionary Computation*, pages 933–938.

Rapin, J. and Teytaud, O. (2018). Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`.

Riviere, M. (2019). Pytorch GAN Zoo. `https://GitHub.com/FacebookResearch/pytorch_GAN_zoo`.

Sobol, I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112.

Storn, R. and Price, K. (1997). Differential evolution &ndash; a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359.

Sukharev, A. G. (1971). Optimal strategies of the search for an extremum. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 11(4).

Teytaud, O., Gelly, S., and Mary, J. (2006). On the ultimate convergence rates for isotropic algorithms and the best choices among various forms of isotropy. In *Proceedings of PPSN*, pages 32–41.

Tuffin, B. (1996). On the use of low discrepancy sequences in monte carlo methods. *Monte Carlo Methods and Applications*, 2(4):295–320.

Tuffin, B. (1998). A new permutation choice in halton sequences. In *Monte Carlo and Quasi-Monte Carlo Methods*, volume 127, pages 427–435. Springer, New York, NY.