Optimal Budgeted Rejection Sampling for Generative Models

Alexandre Verine
LAMSADE, CNRS,
Université Paris-Dauphine,
Université PSL,
Paris, France.

Muni Sreenivas Pydi
LAMSADE, CNRS,
Université Paris-Dauphine,
Université PSL,
Paris, France.

Benjamin Negrevergne
LAMSADE, CNRS,
Université Paris-Dauphine,
Université PSL,
Paris, France.

Yann Chevaleyre
LAMSADE, CNRS,
Université Paris-Dauphine,
Université PSL,
Paris, France.

Abstract

Rejection sampling methods have recently been proposed to improve the performance of discriminator-based generative models. However, these methods are only optimal under an unlimited sampling budget, and are usually applied to a generator trained independently of the rejection procedure. We first propose an Optimal Budgeted Rejection Sampling (OBRS) scheme that is provably optimal with respect to *any* f-divergence between the true distribution and the post-rejection distribution, for a given sampling budget. Second, we propose an end-to-end method that incorporates the sampling scheme into the training procedure to further enhance the model's overall performance. Through experiments and supporting theory, we show that the proposed methods are effective in significantly improving the quality and diversity of the samples.

1 INTRODUCTION

Generative Adversarial Networks (GANs) have significantly improved generation of complex, high dimensional data. In the original paper by Goodfellow et al. (2014), GANs are trained to minimize the Jensen-Shannon divergence between true distribution P and a distribution $\widehat{P}$ induced by a generator G. Since P is generally unknown, the divergence between P and $\widehat{P}$ is estimated using a discriminator T, i.e. a function that discriminates available samples from P and samples generated from $\widehat{P}$. In practice T and G are represented using neural networks and trained simultaneously to estimate the divergence and to minimize it. In this paper, we consider the more general framework of f-GAN introduced by Nowozin et al. (2016), which can be used to minimize *any* f-divergence between P and $\widehat{P}$, including the Jensen-Shannon divergence, the Kullback-Leibler divergence or other divergences (See Table 1).

In most settings, the discriminator is not involved in the generation of new samples beyond the training phase (i.e. it is discarded after training). Building on this observation, several methods such as Discriminator Rejection Sampling (DRS) (Azadi et al., 2019) or Metropolis-Hastings GAN (Turner et al., 2019) have demonstrated how to combine G *and* T using rejection sampling, in order to generate better samples than the ones generated using G alone. In the rest of this paper, we call $\widetilde{P}$ the distribution resulting from G enhanced with rejection sampling.

Unfortunately, these methods suffer from several limitations. First they are only provably optimal when the sampling budget is unlimited. In practice, users have to limit the rejection rate to obtain samples in reasonable time through various empirical means (e.g. by capping the number of iterations of the sampling algorithm). This strategy may not yield the best possible sample for the given budget, an observation that leads to the first question that motivated our contribtion.

Question 1: *How to devise a method that generates the best quality sample under a fixed rejection budget?*

Another important limitation is that, since examples are sampled from $\widetilde{P}$ rather than $\widehat{P}$, the objective should be to minimize the divergence between P and $\widetilde{P}$ rather than the divergence between P and $\widehat{P}$. This raises a second research question that we address in this paper:

Question 2: *Can we train a generator G that directly minimizes $\mathcal{D}_f(P\|\widetilde{P})$ instead of $\mathcal{D}_f(P\|\widehat{P})$?*

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

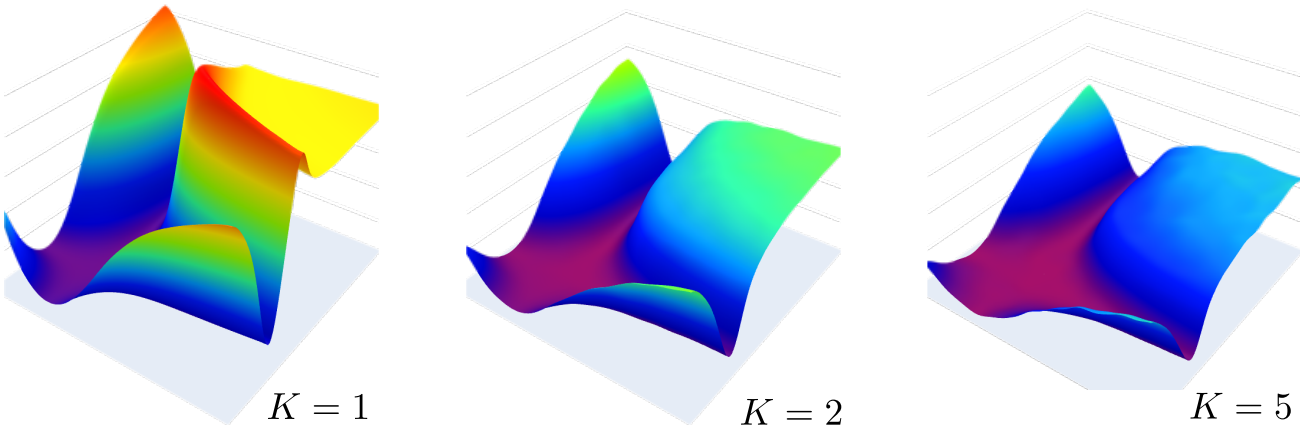


Figure 1: The loss landscape in the parameter domain of a GAN trained on MNIST. The x-axis and y-axis are random directions in the parameter space. The loss is between the target distribution P and the post-rejection distribution. There are three cases: no rejection ($K = 1$), 50% acceptance rate ($K = 2$) and 20% acceptance rate ($K = 5$). OBRS not only reduces loss, but also flattens out the loss landscape and helps avoid local minima.

In this paper, we address Question 1&2, by making following contributions:

- We introduce ORBS, a method that can be used to find an *acceptance function* required to reject/accept samples from $\widehat{P}$ and show in Theorem 3.1 that this function induces the optimal distribution $\widetilde{P}$ under a budget, for *any* f-divergence.
- We characterize the improvement of $\widetilde{P}$ over $\widehat{P}$ in terms of Precision and Recall (Sajjadi et al., 2018) in Theorem 3.3.
- We propose a method to train a generator G to directly minimize an f-divergence between P and $\widetilde{P}$, and we discuss the potential benefits of our method. For example, in Figure 1, we illustrate how OBRS can flatten the loss landscape.

Notation: For the rest of the paper, we use $\mathcal{X} \subseteq \mathbb{R}^d$ to refer to the data space. We use $\mathcal{P}(\mathcal{X})$ to denote the set of probability measures on $\mathcal{X}$ defined on a measure space with the Borel σ-algebra. We use capital letters to denote probability measures (for e.g., $P \in \mathcal{P}(\mathcal{X})$) and small letters to denote their densities (for e.g., $p(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$).

2 BACKGROUND

2.1 f-divergences

The framework of f-divergences can be used to specify a variety of divergences between two probability distributions. An f-divergence is fully characterized by a convex and lower semi-continuous function $f : \mathbb{R}^+ \to \mathbb{R}$ that satisfies $f(1) = 0$. Given f and two probability distributions P and $\widehat{P} \in \mathcal{P}(\mathcal{X})$, the f-divergence between P, $\widehat{P}$ (denoted $\mathcal{D}_f(P\|\widehat{P})$) is defined as follows:

$$\mathcal{D}_f(P\|\widehat{P}) = \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right]. \tag{1}$$

(We assume that P is absolutely continuous with w.r.t. $\widehat{P}$.) Several notable statistical divergences, such as the Kullback-Leibler (KL) divergence ($\mathcal{D}_{\mathrm{KL}}$), the reverse KL divergence ($\mathcal{D}_{\mathrm{rKL}}$), or the Total Variation ($\mathcal{D}_{\mathrm{TV}}$), belong to the class of f-divergences. A overview is provided in Table 1.

A key property of f-divergences is that every f-divergence $\mathcal{D}_f$ admits a dual variational form (Nguyen et al., 2009):

$$\mathcal{D}_f(P\|\widehat{P}) = \sup_{T\in\mathcal{T}} \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f^*(T(\boldsymbol{x}))\right], \tag{2}$$

where $\mathcal{T}$ be the set of all measurable functions $T : \mathcal{X} \to \mathbb{R}$ and $f^*(t) := \sup_{u\in\mathbb{R}}\{tu - f(u)\}$ is the convex conjugate of f. Specifically, the function $T^{\mathrm{opt}} \in \mathcal{T}$ that yields the supremum in (2) can be used to determine the likelihood ratio r^{opt} as follows.

$$r^{\mathrm{opt}}(\boldsymbol{x}) = \nabla f^*\left(T^{\mathrm{opt}}(\boldsymbol{x})\right) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}. \tag{3}$$

2.2 f-GAN, a generalization of GAN

Let $\mathcal{G}$ be the set of all measurable functions $G : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is the latent space and $\mathcal{X}$ is the data space. In the f-GAN framework, the generator $G \in \mathcal{G}$ is used to transform samples from the latent distribution $Q \in \mathcal{P}(\mathcal{Z})$ (typically a multivariate Gaussian) into data samples following the data distribution $\widehat{P}_G \in \mathcal{P}(\mathcal{X})$. G is chosen to minimize the f-divergence $\mathcal{D}_f(P\|\widehat{P}_G)$

Table 1: List of common f-divergences. The generator f is given with its Fenchel conjugate f^*. The optimal discriminator T^{opt} is given to compute the likelihood ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) = \nabla f^*(T^{\text{opt}}(\boldsymbol{x}))$.

Divergence	Notation	$f(u)$	$f^*(t)$	$T^{\text{opt}}(\boldsymbol{x})$
KL	$\mathcal{D}_{\text{KL}}$	$u \log u$	$\exp(t-1)$	$1 + \log p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$
GAN	$\mathcal{D}_{\text{GAN}}$	$u \log u - (u+1)\log(u+1)$	$-\log(1-\exp(t))$	$p(\boldsymbol{x})/\left(p(\boldsymbol{x}) + \widehat{p}(\boldsymbol{x})\right)$
PR	$\mathcal{D}_{\lambda\text{-PR}}$	$\max(\lambda u, 1) - \max(\lambda, 1)$	t/λ	$\lambda \text{sign}\left\{p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) - 1\right\}$

Since P is usually not available, a discriminator $T : \mathcal{X} \to \mathbb{R}$ is used to estimate $\mathcal{D}_f(P \| \widehat{P}_G)$ through the dual variational form in (2), resulting in the following minimax objective (Nowozin et al., 2016).

$$\min_{G \in \mathcal{G}} \max_{T \in \mathcal{T}} \mathbb{E}_{\boldsymbol{x} \sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right]. \tag{4}$$

The optimization procedure is detailed in Algorithm 1. An important special case is that of the original paper of Goodfellow et al. (2014), where $D(\boldsymbol{x}) \coloneqq \exp\left(T(\boldsymbol{x})\right)$, and the minimax objective is as follows:

$$\min_{G} \max_{D} \mathbb{E}_{\boldsymbol{x} \sim P}\left[\log\left(D(\boldsymbol{x})\right)\right] + \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}_G}\left[\log\left(1 - D(\boldsymbol{x})\right)\right]. \tag{5}$$

2.3 Rejection Sampling

Rejection Sampling is a classical method to generate samples from a distribution using samples drawn from a different distribution. In the context of this paper, samples drawn from $\widehat{P}$ are accepted or rejected using an *acceptance function* $a : \mathcal{X} \to [0, 1]$, where $a(\boldsymbol{x})$ is the probability of accepting a sample $\boldsymbol{x}$ from $\widehat{P}$. The distribution induced by the rejection procedure based on a is a new distribution in $\mathcal{P}(\mathcal{X})$ denoted $\widetilde{P}_a$. The density $\widetilde{p}_a(\boldsymbol{x})$ of $\widetilde{P}_a$ has the following form:

$$\widetilde{p}_a(\boldsymbol{x}) = \frac{\widehat{p}(\boldsymbol{x}) a(\boldsymbol{x})}{Z}, \tag{6}$$

where $Z > 0$ is a normalizing constant that ensures that $\int_{\mathcal{X}} \widetilde{p}_a(\boldsymbol{x}) = 1$. The overall acceptance rate is $\mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = Z$. Note that $Z \leq 1$. If $p, \widehat{p}$ are known, and if there are no constraints on the sampling budget (i.e., no lower limit on Z), then a can be set to $a(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x}) M}$ with $M = \sup_{x \in \mathcal{X}} \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$ so that $\widetilde{P}_a$ matches perfectly the target distribution P because $\widetilde{p}_a(\boldsymbol{x}) = \widehat{p}(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x}) Z M} = p(\boldsymbol{x})$ and we have $Z = 1/M$. However in practice for high-dimensional $\mathcal{X}$, M can take high values and set a very low acceptance rate (MacKay, 2005).

Rejection Sampling for GANs: Azadi et al. (2019) propose *Discriminator Rejection Sampling (DRS)* scheme wherein a trained discriminator T is used to approximate the likelihood ratio via the formula,

$$r(\boldsymbol{x}) = \nabla f^*\left(T(\boldsymbol{x})\right), \tag{7}$$

which is an approximation of (3). Thus, the acceptance function of DRS is given by $a_{\text{DRS}}(x) = \frac{r(\boldsymbol{x})}{M}$, where $M = \sup_{\boldsymbol{x}}\{r(\boldsymbol{x})\}$ is estimated using samples $\boldsymbol{x} \sim \widehat{P}$. To account for low acceptance rate, DRS uses a hyper parameter γ to adjust the acceptance rate as,

$$a_{\text{DRS}}(\boldsymbol{x}) = \frac{r(\boldsymbol{x})}{M} e^{-\gamma}. \tag{8}$$

In practice, the discriminator T is calibrated such that $\mathbb{E}_{\widehat{P}}\left[r(\boldsymbol{x})\right] = 1$ which results in an overall acceptance rate of $\mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = \frac{e^{-\gamma}}{M}$. A low value of γ (typically $\gamma < 0$) boosts the acceptance rate.

Related sampling methods: The introduction of DRS has lead to the development of numerous sampling methods that are also applicable to GANs, such as MH-GAN (Turner et al., 2019), DDLS (Che et al., 2021), DOT (Tanaka, 2019), and DG$f$low (Ansari et al., 2021), LatentRS (Issenhuth et al., 2022) and even for Normalizing Flows (Stimper et al., 2022). These methods, relying on gradient ascent or the training of a latent model, have showcased their potential through various applications. However, the sampling is computationally expensive and are not as efficient under a limited time constraint.

Accounting for rejection during training: While the majority of methods employ the rejection sampling scheme post-training, incorporating an *a priori* perspective on the sampling procedure also yields good results empirically. For example, Grover et al. (2018) and Stimper et al. (2022) have embedded latent rejection sampling within their training processes, applying it within a variational inference context and a Normalizing Flow framework, respectively.

3 OPTIMAL BUDGETED REJECTION SAMPLING (OBRS)

Rejection sampling exhibits a well-established efficiency on low-dimensional samples; but the acceptance rate drops when it is applied to higher dimensional samples (MacKay, 2005). In this section, we study the problem of rejection sampling under a limited sampling budget $K \in [1, \infty)$, where K represents the expected number of samples drawn from $\widehat{P}_G$ required to generate a sample from $\widetilde{P}_a$. We start by introducing a method to find the optimal acceptance function under a given budget K (thus addressing research Question 1), then we characterize the improvement provided by this new method using Precision and Recall for generative models (Sajjadi et al., 2018).

3.1 Optimal acceptance function

We recall that P is the true data distribution, $\widehat{P}_G$ (or $\widehat{P}$ for short) is the distribution induced by the generator, and $\widetilde{P}_a$ is the distribution obtained by applying the acceptance function a on samples from $\widehat{P}$. Given a fixed $\widehat{P}$, our goal is to find the acceptance function a that minimizes the divergence between P and $\widetilde{P}_a$ under a budget K, as follows:

$$\min_{a} \quad \mathcal{D}_f(P \| \widetilde{P}_a) \qquad \text{s.t.} \quad \begin{cases} \mathbb{E}_{\widehat{P}}[a(\boldsymbol{x})] \geq 1/K, \\ \forall \boldsymbol{x} \in \mathcal{X},\ 0 \leq a(\boldsymbol{x}) \leq 1. \end{cases} \tag{9}$$

Here the constraint $\mathbb{E}_{\widehat{P}}[a(\boldsymbol{x})] \geq 1/K$ is used to bound the expected acceptance rate. For $K = 1$, the only a satisfying the constraints in (9) is the unit function $a(\boldsymbol{x}) = 1\ \forall\ \boldsymbol{x} \in \mathcal{X}$. This case corresponds to no rejection (or accept w.p. 1) and we have $\widetilde{P}_a = \widehat{P}$ almost everywhere.

Note that the objective $\mathcal{D}_f(P \| \widetilde{P}_a)$ is continuous with respect to a. Since the constraint set for a is closed and bounded, there exists an optimal a for problem (9). In the following theorem, we give an explicit form for the optimal solution a_{O} for finite $\mathcal{X}$ using Lagrangian duality.

Theorem 3.1 (Optimal Acceptance Function)**.** *For a sampling budget $K \geq 1$ and finite $\mathcal{X}$, the solution to problem (9) is,*

$$a_{\mathrm{O}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right), \tag{10}$$

where $c_K \geq 1$ is such that $\mathbb{E}_{\boldsymbol{x}\sim\widehat{p}}[a_{\mathrm{O}}(\boldsymbol{x})] = 1/K$. [1]

[1] This acceptance function was previously introduced by Grover et al. (2018)), with the sole argument that it is a "natural" approximation of the optimal acceptance function. No theoretical argument was provided.

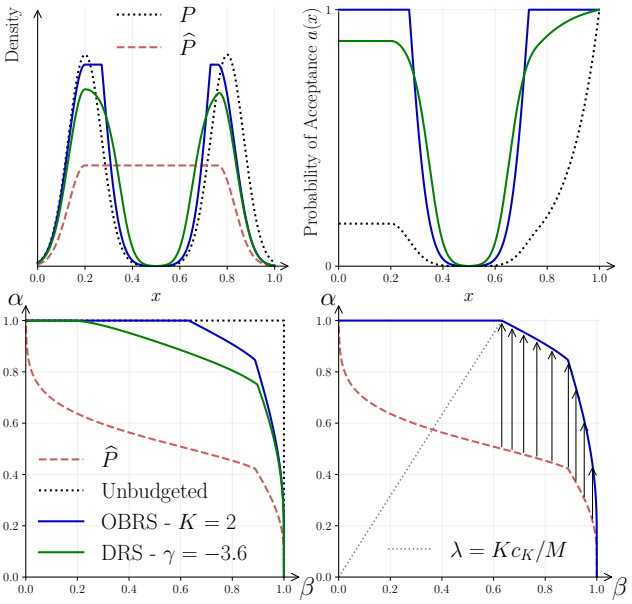


Figure 2: Comparing Unbudgeted, DRS (Azadi et al., 2019) and OBRS (ours) for a one-dimensional example. DRS and OBRS are tuned to reach an acceptance ratio of 50%. TL: The target and learned distributions P and $\widehat{P}$, along the refined distributions. TR: The acceptance functions for the unbudgeted rejection sampling (dotted black), OBRS (blue), and the DRS (green). BL: The PR-Curves of the different models. BR: Visualisation of the improvements by the OBRS. The straight dotted line corresponds to $\lambda = Kc_K/M$.

Few observations should be made on Theorem 3.1:

- The constant c_K is solely determined by K. In practice, we can compute it using a dichotomy algorithm (detailed in Appendix B.2).
- A budget greater than $M = \sup_{x\in\mathcal{X}} \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$ (unbudgeted sampling) implies that $c_K = 1$, and thus $a_{\mathrm{O}}(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{M\widehat{p}(\boldsymbol{x})}$.
- The optimal function a_{O} does not depend on f meaning that OBRS is optimal for various f-divergences including ones that are more sensitive to covering the probability mass or ones that are more sensitive capturing modes. This observation is the base of our analysis on how the OBRS improves Precision and Recall in Section 3.2

Figure 2 illustrates Theorem 3.1 on a one-dimensional example. On the top-left of Figure 2, we draw $\widehat{P}$ and P, where the target distribution P. On the top-right, DRS (green) and OBRS (blue) are compared. We can observe how a_{O} and a_{DRS} lead to different refined distributions $\widetilde{P}_a$.

Finally, we present a theorem showing how much OBRS reduces the f-divergence in general. We show that for any f-divergence, the improvement is linear to K. We also give a tighter version of the bound for the Kullback-Leibler Divergence. Proofs for both results are in Appendix C.

Theorem 3.2. *For any f-divergence, we have*

$$\mathcal{D}_f\left(P\|\widetilde{P}_a\right) \leq \mathcal{D}_f\left(P\|\widehat{P}\right) - \min\left(1, \frac{K-1}{M}\right)\mathcal{D}_f\left(P\|\widehat{P}\right)$$

and for Kullback-Leibler we have for $\gamma = \frac{\log K}{\log M}$

$$\mathcal{D}_{\mathrm{KL}}(P\|\widetilde{P}) \leq (1-\gamma)\left(\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) - \mathcal{D}_\gamma^{\mathrm{R}}(P\|\widehat{P})\right)$$

where $\mathcal{D}_\gamma^{\mathrm{R}}$ is the Rényi divergence with parameter β

3.2 Improvement on the Precision/Recall

A number of recent publications have stressed the importance of measuring the quality of generative models using precision and recall (Kynkäänniemi et al., 2019; Djolonga et al., 2020; Naeem et al., 2020; Cheema and Urner, 2023; Kim et al., 2023b; Verine et al., 2023; Bronnec et al., 2024). In the context of generative modeling, *precision* measures the quality of the generated samples, while *recall* which measures the diversity of the samples. In this section, we introduce Theorem 3.3, that provide a clear characterization of the improvement provided by OBRS in terms of precision and recall.

To model the set of all precision-recall tradeoffs, Simon et al. (2019) introduced the notion of *Precision-Recall Curve* between to distributions P and $\widehat{P}$. This curve, named $\mathrm{PRD}(P, \widehat{P})$, is composed of all coordinate points $(\alpha_\lambda, \beta_\lambda)_{\lambda\in[0,+\infty]} \in [0,1]^2$ defined as follows.

$$\begin{cases} \alpha_\lambda = \mathbb{E}_{\widehat{P}}\left[\min\left\{\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right\}\right] \\ \beta_\lambda = \mathbb{E}_{P}\left[\min\left\{1, \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}\frac{1}{\lambda}\right\}\right] \end{cases} \tag{11}$$

Intuitively, if (α, β) belongs to the Precision-Recall curve, this means that for some fixed recall β, the best achievable precision is α. A more comprehensive definition and explanation of Precision/Recall for generative models is given in Appendix A.

Theorem 3.3 (Precision and Recall Improvement)**.** *Let $K \leq M$ be the budget for the OBRS detailed in Theorem 3.1. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_\mathrm{O}})$ with $\alpha' = \min\{1, K\alpha\}$.*

This theorem shows that for any fixed recall, OBRS consistently improves precision. More precisely, the improved PR-curve is a K-fold vertical scaling of the intial PR-curve capped to 1. The bottom-right part of Figure 2 illustrates this phenomenon. In Appendix B.4, we show a similar theorem for another popular precision-recall measure called the *Information Divergence Frontier* (Djolonga et al., 2020).

4 TRAINING WITH OBRS

In traditional GAN training, the generator G is optimized without considering any *a priori* knowledge regarding the rejection sampling that occurs post-training, potentially leading to suboptimal generative models. This section advocates training with OBRS (Tw/OBRS) for GANs models. First, we introduce the theoretical improvements and the observed effects on the loss function. Then, we introduce an algorithm to incoporate OBRS in the training procedure.

4.1 Principle of Training with OBRS

Let us reformulate Rejection Sampling in the domain of probability measures. Define $B_K(\widehat{P}) = \left\{\widetilde{P} \in \mathcal{P}(\mathcal{X}) | \mathcal{D}_\infty^{\mathrm{R}}(\widetilde{P}\|\widehat{P}) \leq \log K\right\}$, where $\mathcal{D}_\infty^{\mathrm{R}}(\widetilde{P}\|\widehat{P}) = \log(\sup_{\boldsymbol{x}}\{\widetilde{p}(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})\})$ denotes the max-divergence (a limiting case of the α-Rényi Divergence $\mathcal{D}_\alpha^R$ with $\alpha \to \infty$). Note that $B_K(\widehat{P})$ is a convex set. Moreover, the following inclusion holds for any $K_2 \geq K_1 \geq 1$.

$$B_{K_1}(\widehat{P}) \subseteq B_{K_2}(\widehat{P}). \tag{12}$$

The following lemma shows that $B_K(\widehat{P})$ characterizes the set of distributions allowed by a budgeted rejection sampling procedure.

Lemma 4.1. *$\widetilde{P} \in B_K(\widehat{P})$ if and only if there exist an acceptance function $a : \mathcal{X} \to [0,1]$, and a normalization constant Z such that $\widetilde{p}(\boldsymbol{x}) = \widehat{p}(\boldsymbol{x})a(\boldsymbol{x})/Z$ and the acceptance rate is greater that $1/K$.*

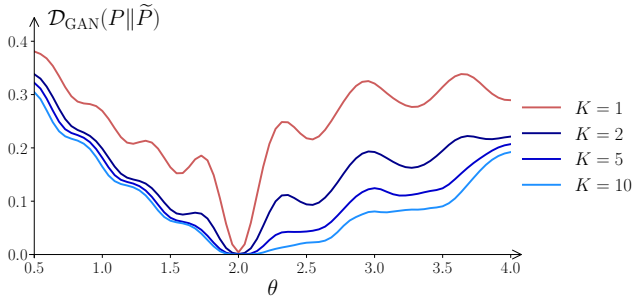
Consider $\widehat{\mathcal{P}} = \left\{\widehat{P} = G_\# Q | G \in \mathcal{G}\right\}$, the set of all distributions $\widehat{P}$ induced by the generator functions from a fixed latent distribution Q. By separating the training process from the rejection sampling process, we are, in effect, solving a two-step minimization problem given below.

$$\text{First solve } \widehat{P}^{\mathrm{opt}} \in \operatorname*{argmin}_{\widehat{P}\in\widehat{\mathcal{P}}} \mathcal{D}_f(P\|\widehat{P}); \tag{13}$$

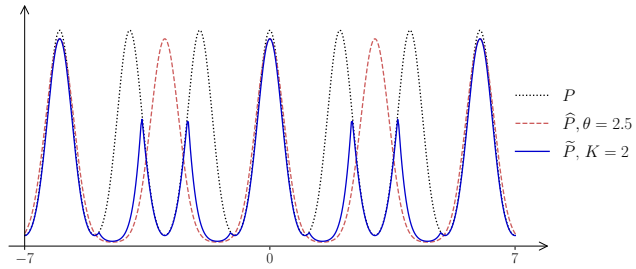
$$\text{Next solve } \widetilde{P}^{\mathrm{opt}} \in \operatorname*{argmin}_{\widetilde{P}\in B_K(\widehat{P}^{\mathrm{opt}})} \mathcal{D}_f(P\|\widetilde{P}). \tag{14}$$

Crucially, $\widehat{P}^{\mathrm{opt}}$ is chosen by the training procedure to optimize (13) whereas the final output distribution $\widetilde{P}^{\mathrm{opt}}$ is assessed via (14), resulting in a mismatched objective. By incorporating the rejection scheme into the training objective, we get:

$$\min_{\widehat{P}\in\widehat{\mathcal{P}}} \min_{\widetilde{P}\in B_K(\widehat{P})} \mathcal{D}_f(P\|\widetilde{P}). \tag{15}$$

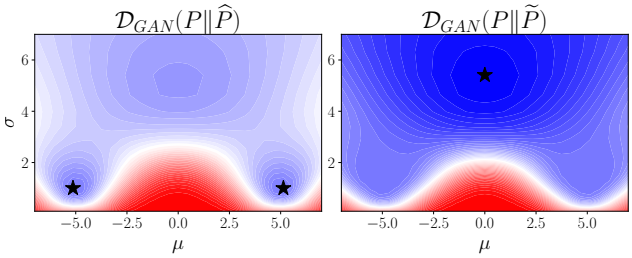


(a) The loss $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$ is calculated for every parameter θ for different budgets K. For $K=1$, it is $\mathcal{D}_{\mathrm{GAN}}(P\|\widehat{P})$.

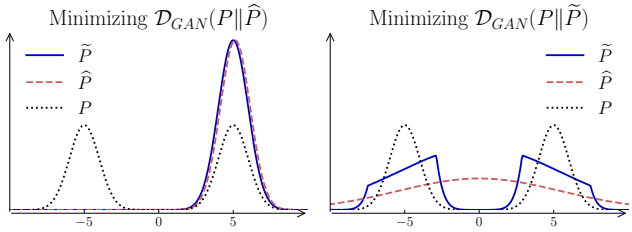


(b) The target distribution P (in dotted black) is a mixture of 10 Gaussians with $\sigma^2 = 0.3$. The approximate distribution is a mixture of 10 Gaussians of $\sigma^2 = 0.4$ separated by θ. $\widetilde{P}$ is computed with OBRS and a budget of $K=2$. Densities are re-scaled and cropped to $[-7, 7]$ for readability.

Figure 3: The loss $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$ is flatten by the OBRS scheme. As the budget K increases, the number of local minima decreases.



(a) $\mathcal{D}_{\mathrm{GAN}}$ is calculated between P and $\widehat{P}$ (left) or $\widetilde{P}$ (right), for all parameters (μ, σ). The stars ($\star$) highlight the minima.



(b) For the target P (in dotted black), the approximation $\widehat{P}$ (in dashed red) corresponds to a minima in Fig. 4a. The post-OBRS distribution $\widetilde{P}$ (in solid blue) is for $K=2$.

Figure 4: One dimensional example of $\mathcal{D}_f$ minimization: P, a mixture of two gaussians is approximated by Gaussian $\widehat{P} = \mathcal{N}(\mu, \sigma^2)$. The distribution $\widehat{P}$ that minimizes $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$ leads to a drastically better approximation $\widetilde{P}$ of P than the post rejection distribution induced by the $\widehat{P}$ that minimizes $\mathcal{D}_{\mathrm{GAN}}(P\|\widehat{P})$.

This re-framing of the objective has the following advantages.

Flattening effect on the parameter landscape: Note that the objective in (15) can be written as,

$$\min_{\widetilde{P}\in\bigcup_{\widehat{P}\in\widehat{\mathcal{P}}} B_K(\widehat{P})} \mathcal{D}_f(P\|\widetilde{P}). \tag{16}$$

Observe that the domain of $\widetilde{P}$ is the dilatation of $\widehat{\mathcal{P}}$ by the convex set B_K, resulting in a smoother set $\bigcup_{\widehat{P}\in\widehat{\mathcal{P}}} B_K(\widehat{P})$. In practice, this results in a flattened loss landscape for optimizing over $\widehat{P}$ as in (15), thus preventing the model from getting stuck in suboptimal local minima. This concept is demonstrated with two examples, showcasing its ability to flatten the parameter landscape. Firstly, Figure 3 shows a one-dimensional example where the loss is flattened by OBRS. Secondly, Figure 1 illustrates a GAN trained to generate MNIST samples. Like in the approach of Li et al. (2018), we present the loss in two arbitrary directions of the parameter space. We observe that OBRS not only reduces the loss but also flattens the landscape, thereby aiding in avoiding local minima. More details are provided in Appendix D.1.

A mass-covering $\widehat{P}$: The optimal $\widehat{P}$ might be different between (13) and (15). Theorem 3.3 explicitly states that OBRS is more efficient on mass-covering models rather than mode-seeking ones, as it improves precision. Taking the rejection sampling into account in the training procedure is pushing the distribution $\widehat{P}$ to be more *suitable* for reject, and thus: more-mode covering. For instance, consider a target distribution P as the Gaussian mixture presented in Figure 4. Assume that the expressivity of $\widehat{P}$ is limited to a single Gaussian $\mathcal{N}(\mu, \sigma)$. If the goal is to naively minimize $\mathcal{D}_{\mathrm{GAN}}$ (defined in Table 1), then, because of the mode-covering property of the divergence, $\widehat{P}$ covers only one mode. In that case, Theorem 3.3 shows that only the precision can be improved, and thus a limited-budget rejection sampling scheme will not reshape $\widehat{P}$, leading to poor coverage. While, if μ and σ are set to directly minimize $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$, then the distribution $\widehat{P}$ changes drastically into a mass covering distribution, allowing the rejection process to match more closely (in terms of $\mathcal{D}_{\mathrm{GAN}}$).

Table 2: Mixture of 25 Gaussiansin 2D. Metrics for the different sampling Methods: Recall (↑) and Precision (↑) as defined in Dumoulin et al. (2017); Calls (↓) of G and D are the number of times the models are called to generate 2500 samples; Time (↓) is the time required to generated 2500 samples. For every metrics, we give the average and standard deviation for 1000 generations of 2500 samples. Best results are emphasized in **bold**.

Model	Recall (%)	Precision (%)	Call of G	Call of D	Time (s)
Baseline G	100.0 ± 0.0	55.80 ± 0.99	2500 ± 0	0 ± 0	0.03 ± 0.01
OBRS (ours) ($K = 2.6$)	100.0 ± 0.0	$\mathbf{92.54 \pm 0.54}$	6262 ± 92	$\mathbf{6262 \pm 92}$	$\mathbf{0.45 \pm 0.01}$
DRS ($\gamma = -0.9$)	100.0 ± 0.0	89.87 ± 0.59	6411 ± 93	6411 ± 93	$\mathbf{0.46 \pm 0.01}$
MH-GAN ($n_{\text{ite}} = 2$)	100.0 ± 0.0	89.98 ± 0.61	6415 ± 45	19292 ± 23	6.38 ± 0.09
DOT ($n_{\text{ite}} = 3$)	100.0 ± 0.0	58.47 ± 1.00	$\mathbf{2500 \pm 0}$	7500 ± 0	0.94 ± 0.14
DG$f$low ($n_{\text{ite}} = 3$)	94.81 ± 2.83	56.00 ± 1.02	7500 ± 0	7500 ± 0	0.67 ± 0.13

4.2 Implementing Tw/OBRS

To implement Tw/OBRS i.e., to solve for the combined objective in (15), we need samples from $\widetilde{P}$ in order to evaluate the final loss $\mathcal{D}_f(P\|\widetilde{P})$. One direct approach is to train a discriminator $\widetilde{T}$ to estimate $D(P\|\widetilde{P})$, and then training a generator to minimize the estimate by minimizing:

$$-\mathbb{E}_{\widetilde{P}}\left[f^*(\widetilde{T}(\boldsymbol{x}))\right] = -\mathbb{E}_{\widehat{P}}\left[Ka_O(\boldsymbol{x})f^*(\widetilde{T}(\boldsymbol{x}))\right]. \quad (17)$$

But, this would require to compute a_{O} which depends on $r(\boldsymbol{x})$ that is obtained by training a discriminator T on $D(P\|\widehat{P})$. In other words, it would require two discriminators T and $\widetilde{T}$. Instead, we propose a method that would require training only a single discriminator T and leverage the primal form of f-divergence give in (1) to estimate $\mathcal{D}_f(P\|\widetilde{P})$ as follows.

$$\begin{aligned}\mathcal{D}_f(P\|\widetilde{P}) &= \mathbb{E}_{\widehat{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \\ &= \mathbb{E}_{\widehat{P}}\left[\frac{\widetilde{p}(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{\widehat{p}(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \\ &= \mathbb{E}_{\widehat{P}}\left[Ka_O(\boldsymbol{x})f\left(\frac{\nabla f^*\left(T(\boldsymbol{x})\right)}{Ka_O(\boldsymbol{x})}\right)\right],\end{aligned}$$

where the last equality follows by plugging in the likelihood ratio estimate of (3). We propose Algorithm 2 that trains a model G to minimize the estimated f-divergence between P and $\widetilde{P}$. This algorithm is, in terms of algorithmic complexity, equivalent to the traditional GAN training procedure detailed in Algorithm 1. We detail in Appendix D.3 how the update of c_K affects the time of the training procedure.

5 EXPERIMENTAL RESULTS

5.1 Sampling methods for 25 Gaussians

We first evaluate our methods using a grid of 5×5 two-dimensional Gaussians following the experimental protocol used by the authors of other GANs sampling methods (Azadi et al., 2019; Turner et al., 2019; Ansari et al., 2021; Che et al., 2021; Tanaka, 2019). Hyperparameters of every methods are set to achieve about 40% acceptance rates ($K = 2.6$) in order to obtain comparable performances. We then measure precision and recall using the methodology proposed by Dumoulin et al. (2017) as well as execution time for every method. Results are presented in Table 2. We observe almost every method achieve 100% recall but that OBRS outperforms all other methods in terms of both precision and sampling time. Detailed experimental settings and a discussion how budget and time affect the performances are available in Appendix D.2.

Algorithm 1 Traditional GAN training procedure

repeat

Update T by ascending the gradient of

$$\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right].$$

Update G by descending the gradient of

$$-\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right].$$

until convergence.

Algorithm 2 GAN Tw/OBRS

repeat

Update T by ascending the gradient of

$$\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right].$$

Update c_K such that $\mathbb{E}_{\widehat{P}_G}\left[a_{\text{O}}(\boldsymbol{x})\right] \leq 1/K$.
(See Alg B.1 in App B.2) for details.)

Update G by descending the gradient of

$$\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[Ka_{\text{O}}(\boldsymbol{x})f\left(\frac{r(\boldsymbol{x})}{Ka_{\text{O}}(\boldsymbol{x})}\right)\right].$$

until convergence.

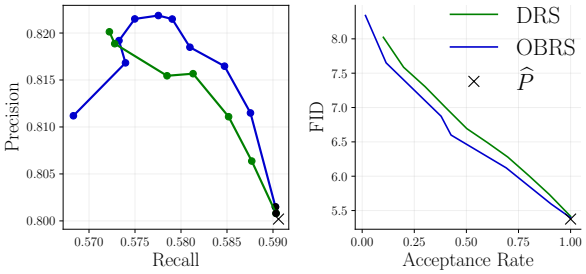


Figure 5: DRS vs. OBRS on a pre-trained BigGAN on CelebA. GAN Baseline model $\widehat{P}_G$, Post-rejection distribution $\widetilde{P}_{a_{\text{DRS}}}$ with DRS, Post-rejection distribution $\widetilde{P}_{a_{\text{O}}}$ with OBRS. (Left) Precision and Recall for different budgets. Lowest budget in black. (Right) FID as a function of the acceptance rate.

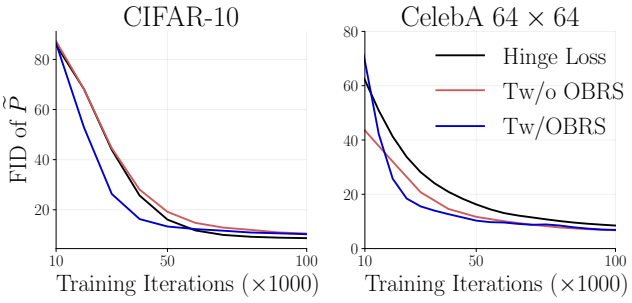
We further demonstrate the impact of the distribution $\widehat{P}$ and particularly the influence of M on the performance disparity between OBRS and DRS. In our experiments, we select hyperparameters to achieve similar acceptance rates. Yet, for varying budgets, the difference of efficiency between OBRS and DRS may increase. Figure D.4 illustrates the behavior of these methods for different values of M.

5.2 OBRS for a pre-trained model

We now investigate how OBRS performs in high dimension. To do so, we use a BigGAN model (Brock et al., 2019) pre-trained on CelebA. Note that the model is originally trained with the hinge loss which is saturating according to Azadi et al. (2019) and leads to a discriminator that is not suitable for density estimation. Thus, following their recommendations, we fine-tune the discriminator to improve density estimation. In Figure 5, we evaluate the resulting model in terms of Precision and Recall (Kynkäänniemi et al., 2019) for 10k samples for $k = 5$ and the FID for 50k samples. When evaluating for multiple budgets between 1 and M, we observe that OBRS outperforms DRS in terms of FID and, for acceptance rates greater than 30%, in terms of precision. We also test the rejection procedure on a diffusion model on CIFAR-10 trained by Karras et al. (2022) with a discriminator trained by Kim et al. (2023a). In Table 3, that the OBRS method improves the FID by a significant margin, while the precision is slightly improved and the recall remains stable.

Table 3: OBRS applied on a Diffusion Model EDM (Karras et al., 2022) with a classifier trained by Kim et al. (2023a). We observe no relevant improvement on the Recall, a slight improvement on the Precision and a significant improvement on the FID.

Acceptance rate	FID	P	R
0.25	1.57	78.48	86.73
0.50	1.58	78.23	86.05
0.75	1.77	77.94	86.54
1	1.97	77.91	86.62



(a) FID during training.

Dataset	Method	FID	P	R
CIFAR-10	Hinge Loss	**8.43**	**84.50**	65.39
32×32	Tw/oOBRS	11.18	83.24	68.44
	Tw/OBRS	8.98	80.09	**69.63**
CelebA	Hinge Loss	9.33	**80.23**	57.78
64×64	Tw/oOBRS	6.33	78.28	**61.02**
	Tw/OBRS	**5.42**	78.01	60.29

(b) Metrics at convergence all between

Figure 6: Training w/OBRS. We use a BigGAN (Brock et al., 2019) trained with hinge loss as a baseline compared to $\mathcal{D}_{\text{GAN}}$ trained without (Tw/oOBRS) and with (Tw/OBRS) OBRS. All metrics are calculated between P and $\widetilde{P}$ with a budget of $K = 2$.

5.3 Training with OBRS

We investigate the Tw/OBRS method discussed in Section 4. We use BigGAN and trained in 3 ways: (1) hinge loss (baseline), (2) $\mathcal{D}_{\text{GAN}}$ loss using the standard method from Algorithm 1 (Tw/oOBRS), and (3) $\mathcal{D}_{\text{GAN}}$ loss using our new method from Algorithm 2 (Tw/OBRS), with $K = 2$. We tested these methods on the CIFAR-10 and CelebA datasets and showed the results in Figure 6. To be fair, we evaluate all 3 models on the refined distribution $\widetilde{P}$ with a budget of $K = 2$. In our experiments, our method demonstrates accelerated convergence and superior performance in terms of FID compared to the alternative approaches. While there is a notable increase in Recall, there is a slight trade-off in Precision.

We also fine-tuned models trained on the hinge loss using our method. We used BigGAN models trained on CelebA and ImageNet in Table 4.

Table 4: Fine-tuning with Tw/OBRS. Pre-trained BigGAN fine-tuned on the $\mathcal{D}_{\text{GAN}}$ with OBRS. We use a BigGAN trained with the hinge loss as a baseline. All metrics are calculated between the target distribution P and the post-distribution with a budget of $K = 2$.

Dataset	Method	FID	P	R
CelebA	Hinge Loss	9.33	**80.23**	57.78
64×64	w/OBRS	**3.74**	74.40	**65.15**
ImageNet	Hinge Loss	12.18	**27.75**	34.33
128×128	w/OBRS	**11.65**	26.84	**46.16**

This set of experiments on training models accounting for rejection shows that intuitions presented in Section 4 are confirmed empirically: the models converge faster and leads to an optimal G more mass-covering.

6 CONCLUSION AND FUTURE WORKS

In this paper, we introduce the concept of budgeted rejection sampling and go a step further by presenting an optimal acceptance function for this sampling method. We use this method to improve discriminator-based models. However, we believe that our Tw/OBRS scheme can be applied to a broader class of generative models. For instance, one could use our approach for Normalizing Flows using the Learned Acceptance/Rejection Sampling method of Stimper et al. (2022). For diffusion models, there is much greater flexibility to refine the distribution through rejection sampling because one can choose to accept a sample at any iteration of the diffusion process. Building on this, one might modify the discriminator refined scored-based sampling of Kim et al. (2023a) to improve diffusion models.

Our work emphasizes the importance of incorporating rejection during the training phase. Practically, this inclusion results in generating distributions with greater recall, ensuring that rejection sampling becomes more effective. In Subsection 4.1, we hypothesize that this improvement may be due to the dilation of the possible set of output distributions $\widehat{\mathcal{P}}$ by a convex set $B_K(\widehat{P})$ during rejection. It would be interesting to further analyze this phenomenon through a theoretical lens.

Acknowledgments

We are grateful for the grant of access to computing resources at the IDRIS Jean Zay cluster under allocations No. AD011011296 and No. AD011014053 made by GENCI.

References

Ansari, A. F., Ang, M. L., and Soh, H. (2021). Refining Deep Generative Models via Discriminator Gradient Flow. arXiv:2012.00780 [cs, stat].

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. (2019). Discriminator Rejection Sampling. arXiv:1810.06758 [cs, stat].

Brock, A., Donahue, J., and Simonyan, K. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv:1809.11096 [cs, stat].

Bronnec, F. L., Verine, A., Negrevergne, B., Chevaleyre, Y., and Allauzen, A. (2024). Exploring Precision and Recall to assess the quality and diversity of LLMs. arXiv:2402.10693 [cs].

Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. (2021). Your GAN is Secretly an Energy-based Model and You Should use Discriminator Driven Latent Sampling. arXiv:2003.06060 [cs, stat].

Cheema, F. and Urner, R. (2023). Precision Recall Cover: A Method For Assessing Generative Models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 6571–6594. PMLR. ISSN: 2640-3498.

Djolonga, J., Lucic, M., Cuturi, M., Bachem, O., Bousquet, O., and Gelly, S. (2020). Precision-Recall Curves Using Information Divergence Frontiers. arXiv:1905.10768 [cs, stat].

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2017). Adversarially Learned Inference. arXiv:1606.00704 [cs, stat].

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. In *27th Conference on Neural Information Processing Systems (NeurIPS 2014)*. arXiv: 1406.2661.

Grover, A., Gummadi, R., Lazaro-Gredilla, M., Schuurmans, D., and Ermon, S. (2018). Variational Rejection Sampling. arXiv:1804.01712 [cs, stat].

Issenhuth, T., Tanielian, U., Picard, D., and Mary, J. (2022). Latent reweighting, an almost free improvement for GANs. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3574–3583, Waikoloa, HI, USA. IEEE.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the Design Space of Diffusion-Based Generative Models. arXiv:2206.00364 [cs, stat].

Kim, D., Kim, Y., Kwon, S. J., Kang, W., and Moon, I.-C. (2023a). Refining Generative Process with Dis-

criminator Guidance in Score-based Diffusion Models. In *Proceedings of the 40 th International Conference on Machine Learning.*, volume 202, Honolulu, Hawaii, USA. JMLR. arXiv:2211.17091 [cs] version: 3.

Kim, P. J., Jang, Y., Kim, J., and Yoo, J. (2023b). TopP&R: Robust Support Estimation Approach for Evaluating Fidelity and Diversity in Generative Models. arXiv:2306.08013 [cs].

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved Precision and Recall Metric for Assessing Generative Models. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.* arXiv: 1904.06991.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the Loss Landscape of Neural Nets. arXiv:1712.09913 [cs, stat].

MacKay, D. J. C. (2005). Information Theory, Inference, and Learning Algorithms.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. (2020). Reliable Fidelity and Diversity Metrics for Generative Models. arXiv:2002.09797 [cs, stat].

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2009). On surrogate loss functions and f-divergences. *The Annals of Statistics*, 37(2). arXiv:math/0510521.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. arXiv:1606.00709 [cs, stat].

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing Generative Models via Precision and Recall. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.* arXiv: 1806.00035.

Simon, L., Webster, R., and Rabin, J. (2019). Revisiting precision recall definition for generative modeling. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5799–5808. PMLR. ISSN: 2640-3498.

Stimper, V., Schölkopf, B., and Hernández-Lobato, J. M. (2022). Resampling Base Distributions of Normalizing Flows. *arXiv:2110.15828 [cs, stat].* arXiv: 2110.15828.

Tanaka, A. (2019). Discriminator optimal transport. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Turner, R., Hung, J., Frank, E., Saatchi, Y., and Yosinski, J. (2019). Metropolis-Hastings Generative Adversarial Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6345–6353. PMLR. ISSN: 2640-3498.

Verine, A., Negrevergne, B., Pydi, M. S., and Chevaleyre, Y. (2023). Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows. arXiv:2305.18910 [cs].

Optimal Budgeted Rejection Sampling for Generative Models Supplementary Materials

A Precision and Recall for Generative Models

According to Sajjadi et al. (2018), *The key intuition is that precision should measure how much of $\widehat{P}$ can be generated by a "part" of P while recall should measure how much of P can be generated by a "part" of $\widehat{P}$.* In this paper, we evaluate how Optimal Budgeted Rejection Sampling affects a given model. To evaluate the improvement theoretically, we need a mathematically grounded method of assessing models and we need this method to assess quality and diversity independently. To do so, we leverage the notion of *PR-Curves* introduced by Sajjadi et al. (2018) and revisited for continuous distributions by Simon et al. (2019).

A.1 From the discrete to the continuous case

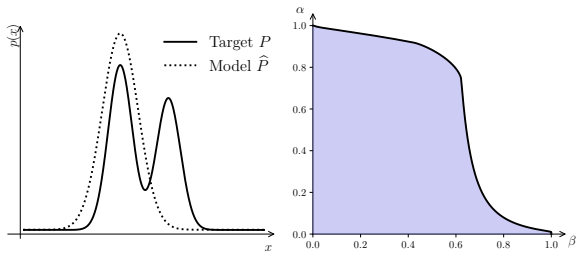
Definition A.1 (Precision and Recall - (Sajjadi et al., 2018))**.** *For $\alpha, \beta \in [0,1]$, the probability distribution $\widehat{P}$ has a precision α at recall β w.r.t. P if there exist distributions μ, ν_P and $\nu_{\widehat{P}}$ such that*

$$P = \beta\mu + (1-\beta)\nu_P \quad \textit{and} \quad \widehat{P} = \alpha\mu + (1-\alpha)\nu_{\widehat{P}}$$

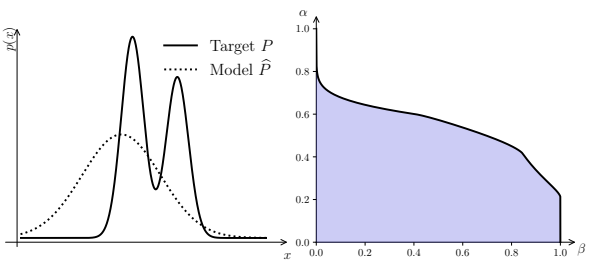
The component ν_P denotes the part of P that is "missed" by $\widehat{P}$. Similarly, $\nu_{\widehat{P}}$ denotes the noise part of $\widehat{P}$.

With this definition, the authors define the set of possible precision-recall pairs: $\mathrm{PR}(P, \widehat{P})$. The frontier of the set of $\mathrm{PR}(P, \widehat{P})$, is the PR-Curve denoted $\mathrm{PRD}(P, \widehat{P})$, parameterized by $\lambda \in [0, \infty]$ and can be computed with the functions:

$$\alpha(\lambda) = \sum_{\boldsymbol{x}_i \in \mathcal{X}} \min\left(\lambda p(\boldsymbol{x}_i), \widehat{p}(\boldsymbol{x}_i)\right) \quad \text{and} \quad \beta(\lambda) = \sum_{\boldsymbol{x}_i \in \mathcal{X}} \min\left(p(\boldsymbol{x}_i), \widehat{p}(\boldsymbol{x}_i)/\lambda\right)$$



(a) High Precision Example



(b) High Recall Example

Figure A.1: Low dimensional examples of distributions with high recall and limited precision and vice versa, with their corresponding PR-Curves. The colored area is the set $\mathrm{PR}(P, \widehat{P})$ and the solid line in black in the frontier $\mathrm{PRD}(P, \widehat{P})$.

This definition has been extended to continuous distributions.

Definition A.2 (Precision and Recall - (Simon et al., 2019)). *For $\alpha, \beta \in [0,1]$, the probability distribution $\widehat{P}$ has a precision α at recall β w.r.t. P if there exists a distribution μ such that*

$$P \geq \beta\mu \quad \textit{and} \quad \widehat{P} \geq \alpha\mu.$$

If also defines a set PR and its frontier is very similar:

$$\alpha(\lambda) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \quad \text{and} \quad \beta(\lambda) = \int_{\mathcal{X}} \min\left(p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})/\lambda\right) \mathrm{d}\boldsymbol{x}.$$

We can reformulate the expressions of the frontier:

$$\begin{cases} \alpha_\lambda = \mathbb{E}_{\widehat{P}}\left[\min\left\{\lambda \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right\}\right] \\ \beta_\lambda = \mathbb{E}_{P}\left[\min\left\{1, \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})} \frac{1}{\lambda}\right\}\right] \end{cases} \tag{18}$$

We can interpret this expression similar to the AUC curve in classification tasks. Consider that the maximum precision and recall are one. Therefore, whenever a point is sampled from $\widehat{P}$ such that $\lambda p(\boldsymbol{x}) < \widehat{p}(\boldsymbol{x})$, the precision decreases further away than 1. In other terms, all the $\boldsymbol{x}$ for which the $\widehat{P}$ overestimate P decrease the precision. On the side, whenever a point is sampled from P such that $\widehat{p}(\boldsymbol{x}) < \lambda p(\boldsymbol{x})$, the recall decreases further away than 1, corresponding to the points where $\widehat{P}$ underestimates P. Let us consider two examples in Figure A.2 and A.4.

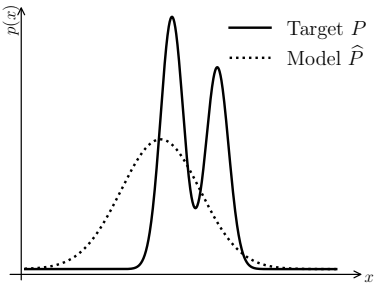
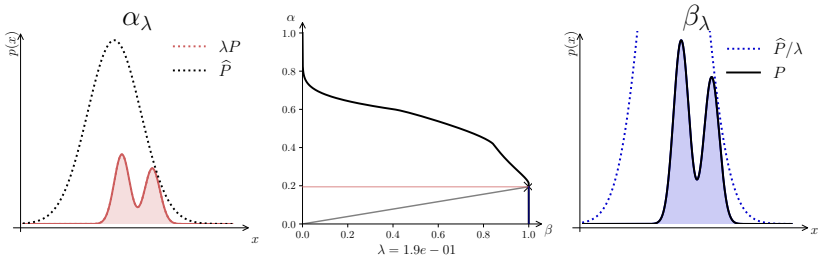
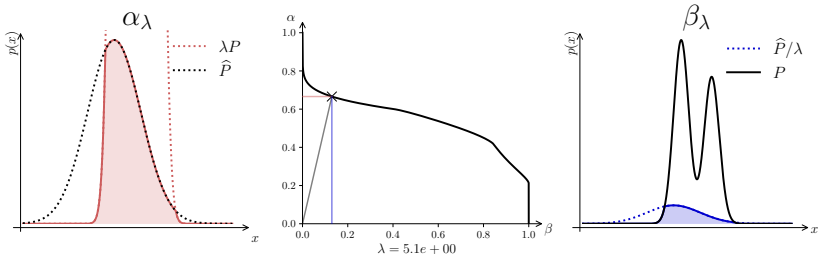


Figure A.2: A target distribution P and the approximated distribution $\widehat{P}$. In this setup, the model is expected to have a decent recall since it covers P but a poor precision since almost half the weight of $\widehat{P}$ does not cover P. In Figures A.3a and A.3b, we show the PR-Cruve and how it is computed.



(a) PR-Curve for the model in Figure A.2, explained for a low λ. The area in red is α_λ and the area in blue is β_λ.



(b) PR-Curve for the model in Figure A.2, explained for a high λ. The area in red in α_λ and the area in blue is β_λ.

Figure A.3: PR-Curves for the model in Figure A.2

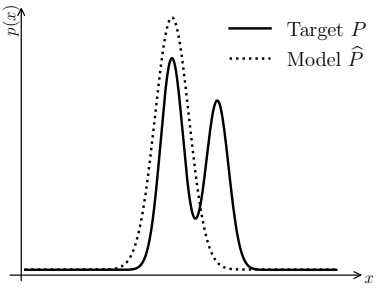
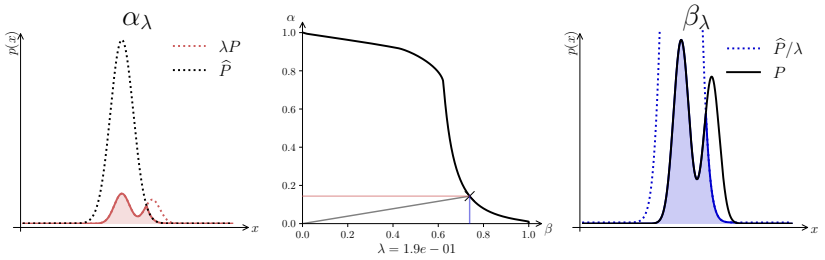
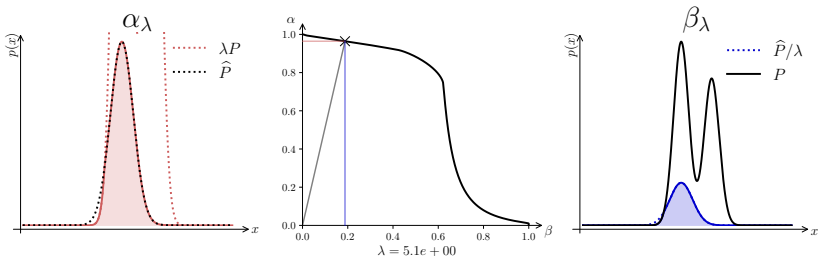


Figure A.4: A target distribution P and the approximated distribution $\widehat{P}$. In this setup, the model is expected to have a poor recall since it covers almost only half the weight of P but a decent precision since the weight covers the contours of P well. In Figures A.5a and A.5b, we represent the PR-Cruve and how it is computed.



(a) PR-Curve for the model in Figure A.4, explained for a low λ. The area in red is α_λ and the area in blue is β_λ.

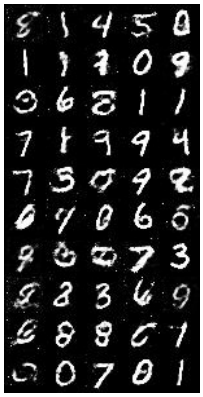


(b) PR-Curve for the model in Figure A.4, explained for a high λ. The area in red in α_λ and the area in blue is β_λ.

Figure A.5: PR-Curves for the model in Figure A.4

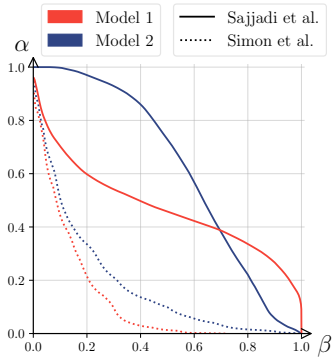
A.2 Precision and Recall in practice

To perfectly compute the set $\mathrm{PRD}(P, \widehat{P})$, one needs the ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. In practice, a variety of heuristics are employed. Sajjadi et al. (2018) use k-NN based algorithm in the Inception latent space to estimated the densities. Simon et al. (2019) use an ensemble of classifiers in Inception's latent space to estimate the likelihood ratio. Verine et al. (2023) use a neural network based discriminator, simlarly to f-GANs, to estimate the likelihood ratio. With these methods, we can compute the PR-Curve for high dimensionnal dataset such as MNIST: see Figure A.6.



(a) Model 1: High Recall
FID: 17.06, IS: 2.69

(b) Model 2: High Precision
FID: 8.80, IS:2.57



(c) PR-Curves for Model 1 and 2.

Figure A.6: Two different models are displayed with very different performances. Model 1 have a great diversity and display all different digits, but contours, backgrounds and shapes are sometimes incoherent. Model 2 is generating coherent samples from only half the classes. Traditional metrics - FID (↓) and IS (↑) - are given for comparison.

B Proof and Supplementary for Section 3

B.1 Proof for Theorem 3.1

The goal is to find an acceptance function $a(\boldsymbol{x})$ that first minimizes the f-divergence between the target distribution P and the distribution after the rejection process $\widetilde{P}_a$. A budget is added to the problem in order to avoid low acceptance rate. We set the budget to be K, the average number of samples to draw before accepting one. With a budget of K, the average acceptance rate is $1/K$. In analogy with the unlimited budget rejection process, the average number of samples to draw in order to keep one is $M = \max_{\mathcal{X}} p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. The function a is the solution of the problem:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(P\|\widetilde{P}_a) \\ \text{s.t.} \quad & \begin{cases} \mathbb{P}\left(\text{acceptance}\right) \geq 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{19}$$

First, we can consider $\mathcal{D}_f(\widetilde{P}_a\|P)$ instead of $\mathcal{D}_f(P\|\widetilde{P}_a)$ without loss of generality: This is because $\mathcal{D}_f(P\|\widetilde{P}_a) = \mathcal{D}_{f'}(\widetilde{P}_a\|P)$ for $f' : x \mapsto xf(1/x)$. Further, the solution to the optimal $a(\boldsymbol{x})$ turns out to be independent of f.

Moreover, we can assume that the budget is always lower that the unlimited budget. In other terms, instead of forcing the acceptance rate to be greater to $1/K$ we can force is to be exactly equal to $1/K$. Then, the probability of acceptance being $\mathbb{P}\left(\text{acceptance}\right) = \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right]$, we can write an equivalent problem as:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(\widetilde{P}_a\|P) \\ \text{s.t.} \quad & \begin{cases} \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{20}$$

Using the definition of the densities in the rejection sampling context, $\widetilde{p}_a(\boldsymbol{x}) = K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x})$, the problem is equivalent to:

$$\begin{aligned} \min_{a} \quad & \mathbb{E}_P\left[f\left(\frac{K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x})}{p(\boldsymbol{x})}\right)\right] \\ \text{s.t.} \quad & \begin{cases} \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{21}$$

Switching to the discrete case, the problem becomes :

$$\begin{aligned} \min_{\boldsymbol{a}\in\mathbb{R}^N} \quad & \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) \\ \text{s.t.} \quad & \begin{cases} \sum_i^N \widehat{p}_i a_i = 1/K \\ \forall i,\, 0 \leq a_i \leq 1 \end{cases} \end{aligned} \tag{22}$$

The Lagrangian function associated with the problem 22 is :

$$\mathcal{L}(\boldsymbol{a}, \mu, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) + \mu\left[\boldsymbol{a}^T\widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T\boldsymbol{\lambda}_1 - \boldsymbol{a}^T\boldsymbol{\lambda}_2 \tag{23}$$

All constraints are affine and the objective function is a convex function, therefore the optimal vector $\boldsymbol{a}^*$ satisfies the KKT conditions:

$$\begin{cases} \nabla_{a_i}\mathcal{L}(\boldsymbol{a}^*, \mu^*\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*) = K\widehat{p}_i\nabla f\left(a_i^* \frac{\widehat{p}_i K}{p_i}\right) + \mu^*\widehat{p}_i + (\lambda_{1i}^* - \lambda_{2i}^*) = 0, \quad \forall i \\ \sum_i a_i^*\widehat{p}_i = 1/K \\ \lambda_{1i}^*(a_i^* - 1) = 0, \quad \forall i \\ \lambda_{2i}^* a_i^* = 0, \quad \forall i \\ \lambda_{1i}^*, \lambda_{2i}^* \geq 0, \forall i \end{cases} \tag{24}$$

Using the 1st condition:

$$a_i^* = \frac{p_i}{\widehat{p}_i K} \left[\nabla f\right]^{-1} \left(\frac{\lambda_{2i}^* - \lambda_{1i}^*}{K\widehat{p}_i} - \mu/K \right) \tag{25}$$

Since $\left[\nabla f\right]^{-1} = \nabla f^*$:

$$a_i^* = \frac{p_i}{\widehat{p}_i K} \left[\nabla f^*\right] \left(\frac{\lambda_{2i}^* - \lambda_{1i}^*}{\widehat{p}_i K} - \mu/K \right) \tag{26}$$

If the Pearson χ^2 is put aside, all the usual f^* are strictly increasing functions. Therefore, according to Eq 26, all $a_i > 0$. Thus all $\lambda_{2i}^* = 0$. The KKT conditions 24 become :

$$\begin{cases} K\widehat{p}_i \nabla f \left(a_i^* \frac{\widehat{p}_i K}{p_i} \right) + \mu^* \widehat{p}_i + \lambda_{1i}^* = 0, \quad \forall i \\ \sum_i a_i^* \widehat{p}_i = 1/K \\ \lambda_{1i}^* (a_i^* - 1) = 0, \forall i \\ \lambda_{1i}^* \geq 0, \forall i \end{cases} \tag{27}$$

And thus :

$$a_i^* = \frac{p_i}{\widehat{p}_i K} \left[\nabla f^*\right] \left(-\frac{\lambda_{1i}^*}{\widehat{p}_i K} - \mu/K \right) \tag{28}$$

To get the full formula for a_i^*, we need to compute the λ_{1i}s. For this purpose, let us use strong duality to reformulate our problem:

$$\min_{\boldsymbol{a}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} \sum_i^N p_i f \left(a_i \frac{\widehat{p}_i K}{p_i} \right) + \mu \left[\boldsymbol{a}^T \widehat{\boldsymbol{p}} - 1/K \right] + (\boldsymbol{a} - \mathbb{1})^T \boldsymbol{\lambda}_1 \tag{29}$$

$$= \max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} \min_{\boldsymbol{a}} \sum_i^N p_i f \left(a_i \frac{\widehat{p}_i K}{p_i} \right) + \mu \left[\boldsymbol{a}^T \widehat{\boldsymbol{p}} - 1/K \right] + (\boldsymbol{a} - \mathbb{1})^T \boldsymbol{\lambda}_1 \tag{30}$$

Then, we can use the Fenchel Conjugate:

$$\begin{aligned} \min_{\boldsymbol{a}} \sum_i^N p_i^* f \left(a_i \frac{\widehat{p}_i K}{p_i^*} \right) + \mu \left[\boldsymbol{a}^T \widehat{\boldsymbol{p}} - 1/K \right] + (\boldsymbol{a} - \mathbb{1})^T \boldsymbol{\lambda}_1 &= \min_{\boldsymbol{a}} \sum_i^N p_i^* \left[f \left(a_i \frac{\widehat{p}_i K}{p_i^*} \right) - a_i \left(\frac{-\mu \widehat{p}_i - \lambda_{1i}}{p_i} \right) \right] \\ &\quad - \mu/K - \mathbb{1}^T \boldsymbol{\lambda}_1 \\ &= - \sup_{\boldsymbol{a}} \left\{ \sum_i^N p_i^* \left[a_i \left(\frac{-\mu \widehat{p}_i - \lambda_{1i}}{p_i} \right) - f \left(a_i \frac{\widehat{p}_i K}{p_i^*} \right) \right] \right\} \\ &\quad - \mu/K - \mathbb{1}^T \boldsymbol{\lambda}_1 \\ &= - \sum_i^N \left[p_i^* f^* \left(- \frac{p_i^*}{\widehat{p}_i K} \frac{\mu \widehat{p}_i + \lambda_{1i}}{p_i} \right) \right] - \mu/K - \mathbb{1}^T \boldsymbol{\lambda}_1 \\ &= - \sum_i^N \left[p_i^* f^* \left(-\mu/K - \frac{\lambda_{1i}}{\widehat{p}_i K} \right) \right] - \mu/K - \mathbb{1}^T \boldsymbol{\lambda}_1 \end{aligned} \tag{31}$$

Define $u_i = \frac{\lambda_{i1}}{\hat{p}_i}$, assuming $\hat{p}_i > 0$ everywhere. Note that the constraints $\lambda_{i1} \geq 0$ and $u_i \geq 0$ are equivalent. The above equation becomes

$$\sup_{\lambda_1 \geq 0} \mathcal{L} \left(\boldsymbol{a}^*, \mu^*, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2^* \right) = \sup_{\boldsymbol{u} \geq 0} - \sum_i^N p_i^* f^* \left(- \left(\mu^* + u_i \right) / K \right) - \sum_i^N \hat{p}_i u_i - \mu^* / K \tag{32}$$

Let us make another change of variable to make a conjugate form appear. Define $v_i = -\left(\mu^* + v_i\right)$. So $u_i = -\mu^* - v_i$ and the constraint $u_i \geq 0$ becomes $v_i \leq -\mu^*$. Also, define $g(t) = f(Kt)$. Then $g^*(t) = f^*(\frac{t}{K})$. Above equation becomes

$$\sup_{\boldsymbol{\lambda}_1 \geq 0} \mathcal{L}\left(\boldsymbol{a}^*, \mu^*, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2^*\right) = \sup_{\boldsymbol{v} \leq -\mu^*} \sum_i^N \hat{p}_i v_i - \sum_i^N p_i g^*\left(v_i\right) - \mu^*\left(K-1\right) \tag{33}$$

Recall that $\arg\sup_t \langle a, t\rangle - f(t) = \nabla f^*(a)$ and $\arg\sup_t \langle a, t\rangle - f^*(t) = \nabla f(a)$. Thus, given μ^* we can compute the optimal values of v_i one by one as follows:

$$\begin{aligned} v_i^* &= \arg \sup_{v_i \leq -\mu^*} \hat{p}_i v_i - p_i g^*\left(v_i\right) \\ &= \arg \sup_{v_i \leq -\mu^*} \frac{\hat{p}_i}{p_i} v_i - g^*\left(v_i\right) \\ &= \min\left(-\mu^*, \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right) \end{aligned}$$

So $u_i^* = \max\left(0, -\mu^* - \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right)$. This gives us the optimal values of λ_{i1}^*. Note that $\nabla g(t) = K \nabla f\left(Kt\right)$. Replacing $\frac{\lambda_{1i}^*}{\hat{p}_i}$ by u_i^* in the formula of a_i^* gives us:

$$\begin{aligned} a_i^* &= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(-\mu^*/K - \max\left(0, -\mu^* - \nabla g\left(\frac{\hat{p}_i}{p_i}\right)/K\right)\right) \\ &= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(-\mu^*/K + \min\left(0, \mu^* + \nabla g\left(\frac{\hat{p}_i}{p_i}\right)/K\right)\right) \\ &= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(\min\left(-\mu^*, \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right)/K\right) \\ &= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(\min\left(-\mu^*/K, \nabla f\left(\frac{\hat{p}_i K}{p_i}\right)\right)\right) \end{aligned}$$

Note that ∇f^* is strictly increasing, thus:

$$\begin{aligned} a_i^* &= \frac{p_i}{\hat{p}_i K} \min\left(\nabla f^*\left(-\frac{\mu^*}{K}\right), \frac{\hat{p}_i K}{p_i}\right) \\ &= \min\left(\frac{p_i}{\hat{p}_i K} \nabla f^*\left(-K\mu^*\right), 1\right) \end{aligned}$$

Note that $\nabla f^*\left(-\mu^*/K\right)$ is a constant. So the optimal acceptance function under budget looks like $a(\boldsymbol{x}) = \min\left(1, c\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)$ for some constant c defined by K only as:

$$\int_{\mathcal{X}} \min\left(\widehat{p}(\boldsymbol{x}), cp(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} = 1/K. \tag{34}$$

To facilitate the understanding of c, we can set this constant to be equal to c/M instead. Thus,

$$a(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \frac{c}{M}, 1\right) \tag{35}$$

With that notation, $c \geq 1$ and if the optimal unlimited acceptance function is obtained with $c = 1$:

$$a(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \frac{1}{M}, 1\right) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})M} \tag{36}$$

B.2 Algorithm to compute c_K

In Section 3, we show that the optimal acceptance function is

$$a(\boldsymbol{x}, c_K) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right). \tag{37}$$

The constant c_K is determined exclusively by the budget K. In practice, we can draw a set of samples from $\widehat{P}$ and adjust c_K to obtain the correct budget. We use a dichotomy algorithm detailed in Algorithm B.1.

Algorithm B.1 Dichotomy to compute c_K.

Input: N generated samples $\boldsymbol{x}_1^{\text{fake}}, \dots, \boldsymbol{x}_N^{\text{fake}} \sim \widehat{P}$
Parameter: Budget K, Threshold ϵ
Output: Constant c_K

Let $c_{\min} = 1e^{-10}$ and $c_{\max} = 1e^{10}$.
$c_K = (c_{\max} + c_{\min})/2$
Define the loss $\mathcal{L}(c_K) = \sum_{i=1}^{N} a\left(\boldsymbol{x}_i^{\text{fake}}, c_K\right) - \frac{1}{K}$
while $|\mathcal{L}(c_K)| \geq \epsilon$ **do**
    if $\mathcal{L}(c_K) > \epsilon$ **then**
        $c_{\max} = c_K$
    else if $\mathcal{L}(c_K) < -\epsilon$ **then**
        $c_{\min} = c_K$
    end if
    Update: $c_K = (c_{\max} + c_{\min})/2$
    Update: $\mathcal{L}(c_K)$
end while

B.3 Proof for Theorem 3.3

First, with $a(\boldsymbol{x}) = \min\left(1, \frac{c_k}{M}\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\}\right)$, let us recall that

$$\widetilde{p}_a(\boldsymbol{x}) = K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x}) \tag{38}$$

$$= \min\left(K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right). \tag{39}$$

Thus:

$$\alpha_\lambda(P\|\widetilde{P}_a) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widetilde{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \tag{40}$$

$$= \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}. \tag{41}$$

Naturally, the precision can be evaluated for λ lower or greater than Kc_K/M. For $\lambda \leq Kc_K/M$:

$$\alpha_\lambda(P\|\widetilde{P}_a) = \int_{\mathcal{X}} \min\left(K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \tag{42}$$

$$= K\int_{\mathcal{X}} \min\left(\frac{c_K}{M}p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \tag{43}$$

$$= K\mathbb{E}_{\widehat{P}}\left[\min\left(\frac{c_K}{M}\frac{p(\boldsymbol{x})}{p(\boldsymbol{x})}, 1\right)\right] \tag{44}$$

$$= K\frac{1}{K} \quad \text{by definition of} c_K, \tag{45}$$

$$= K\alpha_{c_K/M}(P\|\widehat{P}). \tag{46}$$

Thus, under a given threshold Kc_K/M, the precision is constant and equal to $K\alpha_{c_K/M}(P\|\widehat{P})$. Moreover, we can give a lower bound on this constant value in terms of K. As a matter of fact, α_λ is an increasing function

of λ, therefore:

$$\alpha_{c_K/M}(P\|\widehat{P}) \geq \alpha_{1/M}(P\|\widehat{P}) \tag{47}$$

$$\geq \int_{\mathcal{X}} \min\left(\frac{1}{M}p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}. \tag{48}$$

Finally, by the definition of M, for every $\boldsymbol{x} \in \mathcal{X}$, $\frac{1}{M}p(\boldsymbol{x}) \leq \widehat{p}(\boldsymbol{x})$. Consequently,

$$\alpha_\lambda(P\|\widetilde{P}_a) = K\alpha_{c_K/M}(P\|\widehat{P}) \geq \frac{K}{M}. \tag{49}$$

For $\lambda \leq Kc_K/M$:

$$\alpha_\lambda(P\|\widetilde{P}_a) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), K\widehat{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \tag{50}$$

$$= K\int_{\mathcal{X}} \min\left(\frac{\lambda}{K}p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \tag{51}$$

$$= K\alpha_{\lambda/K}(P\|\widehat{P}). \tag{52}$$

And, since $\lambda/K \geq \lambda/M$:

$$\alpha_\lambda(P\|\widetilde{P}_a) = K\alpha_{\lambda/K}(P\|\widehat{P}) \geq K\alpha_{\lambda/M}(P\|\widehat{P}). \tag{53}$$

Finally, with $\alpha_\lambda = \lambda\beta_\lambda$,

$$\beta_\lambda(P\|\widetilde{P}_a) = \frac{K}{\lambda}\alpha_{\lambda/K}(P\|\widehat{P}) = \frac{K}{(\lambda)}\frac{\lambda}{K}\beta_{\lambda/K}(P\|\widehat{P}) = \beta_{\lambda/K}(P\|\widehat{P}), \tag{54}$$

And, since $\lambda/K \leq M/c_K$, we have:

$$\beta_\lambda(P\|\widetilde{P}_a) \geq \beta_{c_K/M}(P\|\widehat{P}), \tag{55}$$

Therefore we have two regimes:

- For $\lambda \geq \frac{Kc_K}{M}$:

$$\alpha_\lambda\left(P\|\widetilde{P}_{a_\mathrm{O}}\right) = 1 \quad \text{and} \quad \beta_\lambda\left(P\|\widetilde{P}_{a_\mathrm{O}}\right) = 1/\lambda$$

- For $\lambda \leq \frac{Kc_K}{M}$:

$$\begin{cases}\alpha_\lambda(P\|\widetilde{P}_{a_\mathrm{O}}) = K\alpha_{\lambda/K}(P\|\widehat{P}) \\ \beta_\lambda(P\|\widetilde{P}_{a_\mathrm{O}}) = \beta_{\lambda/K}(P\|\widehat{P})\end{cases}$$

  This can be seen as a vertical scaling of the PR-Curve. For a given point (α, β) in $\mathrm{PRD}(P\|\widehat{P})$, then the point with the same β in $\mathrm{PRD}(P\|\widetilde{P})$ has a precision $K\alpha$, up to a certain saturating level ($\alpha < 1$).

B.4 Information Divergence Frontier Improvement

In Djolonga et al. (2020), the authors define another precision-recall curve, named the Information Divergence Frontiers:

$$\mathcal{F}_\beta^\cap(P,Q) = \left\{(\pi,\rho) \in \mathcal{R}_\beta^\cap(P,Q) : \nexists\, (\pi',\rho') \in \mathcal{R}^\cap(P,Q)\ s.t.\ \pi' < \pi, \rho' < \rho\right\}$$

Where $\mathcal{R}_\beta^\cap(P,Q) = \{(\mathcal{D}_\beta(R,Q), \mathcal{D}_\beta(R,P)) : R \in \mathcal{P}(\mathcal{X})\}$ and where $\mathcal{D}_\beta$ is the Renyi divergence parametrized by β.

As an immediate corrolary of the previous theorem and of proposition 6 of Djolonga et al. (2020), we can write the following:

Corollary B.1. *Under the same setting as theorem 3.3, for any $(\pi,\rho) \in \mathcal{F}_\infty^\cap\left(P,\widehat{P}\right)$ we have $(\pi',\rho) \in \mathcal{F}_\infty^\cap\left(P,\widetilde{P}_{a_O}\right)$ with $\pi' = \max(0, \pi - \log K)$.*

C Bounds

Theorem C.1. *Let $M = \sup_{x\in\mathcal{X}} \frac{p(x)}{\widehat{p}(\boldsymbol{x})}$. For any f-divergence, we have*

$$\mathcal{D}_f(P\|\widetilde{P}_a) \le \mathcal{D}_f(P\|\widehat{P}) - \min\left(1, \frac{K-1}{M}\right)\mathcal{D}_f(P\|\widehat{P})$$

and for Kullback-Leibler we have for $\beta = \frac{\log K}{\log M}$

$$\mathcal{D}_{\mathrm{KL}}(P\|\widetilde{P}) \le (1-\beta)\left(\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) - \mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P})\right)$$

where $\mathcal{D}_\beta^{\mathrm{R}}$ is the Renyi divergence with parameter β

Proof. For both bounding the f-divergence and the KL divergence, the strategy will be the same. We want to show that

$$\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_f(P\|\widetilde{P}) \ge \text{some lower bound}$$

Note that for any density p_α such that such that $p_\alpha \le K\widehat{p}$, we have $\mathcal{D}_f(P\|\widetilde{P}) \le \mathcal{D}_f(P, P_\alpha)$ so

$$\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_f(P\|\widetilde{P}) \ge \mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_f(P, P_\alpha)$$

So once we have a suitable p_α, we need to show the lower bound holds:

$$\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_f(P, P_\alpha) \ge \text{some lower bound}$$

For bounding general f-divergences, we will choose $p_\alpha = \widehat{p} + \alpha(p - \widehat{p})$ with $\alpha = \min\left(1, (K-1)\inf_{x\in\mathcal{X}} \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}\right)$

Let us first show that $p_\alpha \le K\widehat{p}$.

$$p_\alpha \le \widehat{p} + (K-1)\inf_x \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}(p - \widehat{p})$$

Note that for any $\boldsymbol{x}' \in \mathcal{X}$,

$$\inf_x \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}(p(\boldsymbol{x}') - \widehat{p}(\boldsymbol{x}')) \le \widehat{p}(\boldsymbol{x}')$$

So

$$p_\alpha(\boldsymbol{x}) \leq \widehat{p} + (K-1)\widehat{p} \leq K\widehat{p}$$

Next, let us show the lower bound. Recall that $\mathcal{D}_f(p,\cdot)$ is convex in its second argument. Thus, convexity implies:

$\mathcal{D}_f(P\|P_\alpha) \leq (1-\alpha)\,\mathcal{D}_f(P\|\widehat{P}) + \alpha\mathcal{D}_f(P\|P) \leq (1-\alpha)\,\mathcal{D}_f(P\|\widehat{P})$

Now to apply the same type of idea to bound the KL, let us define $p_\beta(\boldsymbol{x}) = \frac{1}{Z}\widehat{p}(\boldsymbol{x})^{1-\beta}p(\boldsymbol{x})^\beta$, where $Z = \int \widehat{p}(\boldsymbol{x})^{1-\beta}p(\boldsymbol{x})^\beta d\mu(\boldsymbol{x}) = e^{(\beta-1)\mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P})}$ where $\mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P}) = \frac{1}{\beta-1}\log\int p^\beta \widehat{p}^{1-\beta}d\mu$ is the Renyi divergence of parameter β and μ is the reference measure.

First, let us choose $\beta' = \frac{\log K - (1-\beta')R_{\beta'}(P\|\widehat{P})}{\log M}$ and let us show as before that $p_{\beta'} \leq K\widehat{p}$. More precisely, let us show that $\log\frac{p_{\beta'}(\boldsymbol{x})}{K\widehat{p}(\boldsymbol{x})} \leq 0$

For any x, we have

$$\begin{aligned}
\log\frac{p_{\beta'}(\boldsymbol{x})}{K\widehat{p}(\boldsymbol{x})} &= (1-\beta')\log\widehat{p}(\boldsymbol{x}) + \beta'\log p(\boldsymbol{x}) - \log Z - \log K\widehat{p}(\boldsymbol{x}) \\
&= \log\widehat{p}(\boldsymbol{x}) + \beta'\log\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - (\beta'-1)R_{\beta'}(P\|\widehat{P}) - \log K\widehat{p}(\boldsymbol{x}) \\
&= \beta'\log\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - (\beta'-1)R_{\beta'}(P\|\widehat{P}) - \log K \\
&\leq \frac{\log K - (1-\beta')R_{\beta'}(P\|\widehat{P})}{\log M}\log\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - (\beta'-1)\mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P}) - \log K \\
&\leq \log K - (1-\beta')R_{\beta'}(P\|\widehat{P}) - (\beta'-1)R_{\beta'}(P\|\widehat{P}) - \log K \\
&\leq 0
\end{aligned}$$

More generally it is easy to see that for all $\beta \in [0,\beta']$, we have $p_\beta \leq K\widehat{p}$. For convenience, we will choose $\beta = \frac{\log K}{\log M}$. Clearly, $\beta \leq \beta'$ so $p_\beta \leq K\widehat{p}$. Finally, let us compute $\mathcal{D}_{\mathrm{KL}}(P\|P_\beta)$

$$\begin{aligned}
\mathcal{D}_{\mathrm{KL}}(P\|P_\beta) &= \int p(\boldsymbol{x})\log\frac{p(\boldsymbol{x}).Z}{\widehat{p}(\boldsymbol{x})^{1-\beta}p(\boldsymbol{x})^\beta}d\mu(\boldsymbol{x}) \\
&= \int p(\boldsymbol{x})\log\left(\frac{p(\boldsymbol{x})^{1-\beta}}{\widehat{p}(\boldsymbol{x})^{1-\beta}}.Z\right)d\mu(\boldsymbol{x}) \\
&= (1-\beta)\int p(\boldsymbol{x})\log\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)d\mu(\boldsymbol{x}) + \log Z \\
&= (1-\beta)\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) - (1-\beta)\mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P}) \\
&= (1-\beta)\left(\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) - \mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P})\right)
\end{aligned}$$

Thus the result holds: $\mathcal{D}_{\mathrm{KL}}(P\|\widetilde{P}) \leq \mathcal{D}_{\mathrm{KL}}(P\|P_\beta) \leq (1-\beta)\left(\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) - \mathcal{D}_\beta^{\mathrm{R}}(P\|\widehat{P})\right)$ □

D Additional Experiments

In this section, we provide more details on the different experiments. First, in Section D.1, we explain how the loss landscape is produced. Then, in Section D.2, we provide more details on how the budget affects the results on the 25 Gaussians experiments. Finally, in Section D.3, we compare the traditional GAN training procedure and our approach in terms of time complexity.

D.1 Smoothing the lanscape parameters for MNIST

Similarly to Li et al. (2018), the goal is to observe a two dimensional projection of the parameters domain of a neural network, and compute the loss on this domain.

To do so, we train a simple GAN on MNIST. Both the generator and the discriminator are based on 3 linear layers with Leaky Relu. The models are trained using the tradition approach described in Algorithm 1. Let us define θ_0, the parameter vector of the generator G_{θ_0}. We randomly draw two directions θ_1 and θ_2 in the parameter domain: defining an hyperspace of generators defined as $G_{\theta_0+x\theta_1+y\theta_2}$ with $(x, y) \in \mathbb{R}^2$. Then, given any parameters θ, we train a new discriminator T_2 based on samples of P and $\widehat{P}_{G_\theta}$ to determine the baseline loss landscape ($K = 1$). For the OBRS loss landscape, we fine-tune the initial model T in order to perform optimal budgeted rejection sampling. Finally, similar to the baseline, a new discriminator T_2 is trained to estimate the loss, but based on samples P and $\widetilde{P}$.

In Figure D.1, we plot the loss surface. In addition to Figure 1, we represent a batch of samples drawn from G_{θ_0} (lower left) and from the G_θ given the worst loss (upper right). When OBRS is applied, we show in red the rejected samples and in green the accepted samples.

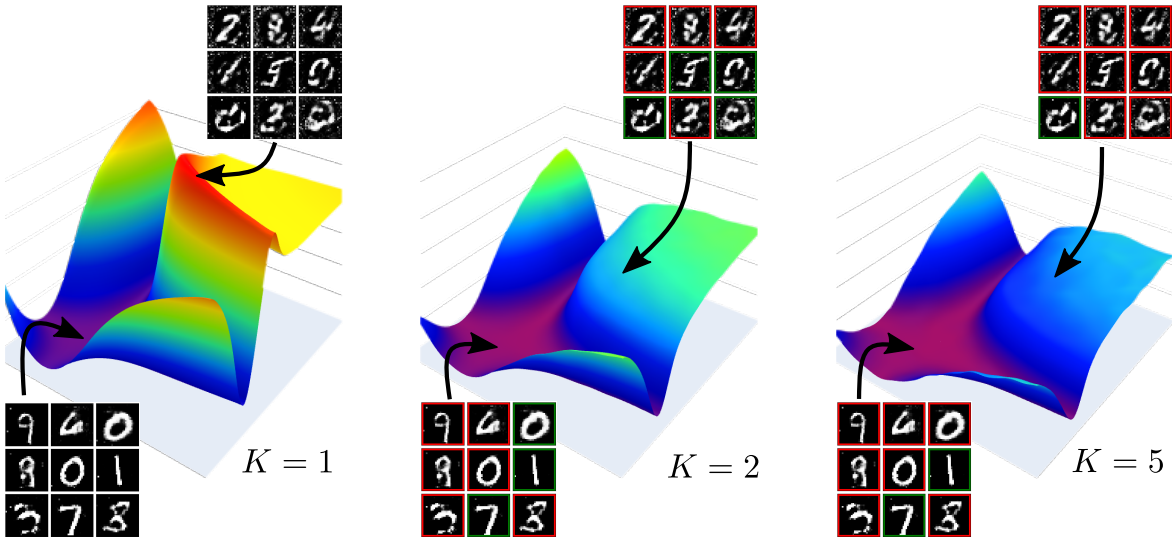


Figure D.1: The Loss surface in the parameters domain of a DCGAN trained on MNIST randomly projected in 2D, observed for different rejection sampling budgets.

D.2 Additional Results on the 2D 25-Gaussians Dataset

In this section, we provide more details on the GAN trained on the 25 Gaussians. The goal of this experiment is to compare OBRS with other rejection sampling methods such as DRS (Azadi et al., 2019) or MH-GAN (Turner et al., 2019), but also with other sampling techniques that involve gradient descent, such as DOT (Tanaka, 2019) and DG$f$low (Ansari et al., 2021). We train a simple GAN on 25 two-dimensional Gaussians and apply each method. We tune (when possible) the method to obtain around 6500 inferences from the generator to generate 2500 samples. To be more precise, both ORBS and DRS are easily tunable; however, the rejection rate of MH-GAN highly depends on the number of iterations of the algorithm. Therefore, we set the number of iterations to 2 to obtain 40% and then tune γ and K to achieve a similar acceptance rate. We obtain the results plotted in Figure D.2.

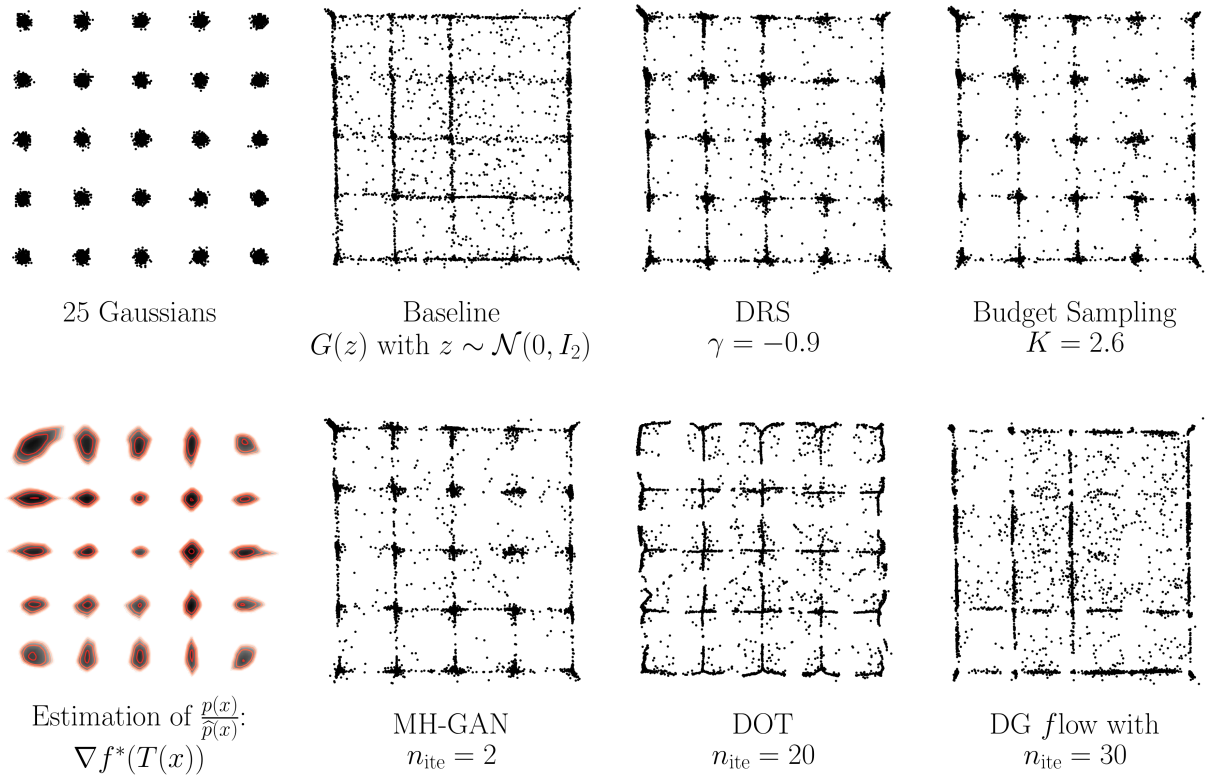


Figure D.2: Visual Representation of the different sampling methods.

In the previous experiment; we arbitrarily set up the acceptance rate (or the sampling time for the *non* rejection sampling methods). We also compare the methods for different sampling time. Since for most methods, the recall was equivalent, we compare the precision denoted as *high quality samples* in Dumoulin et al. (2017). We observe that for any given precision under 93%, the fastest method is the OBRS. However, OBRS, like MH-GAN and DRS, appear to be capped.

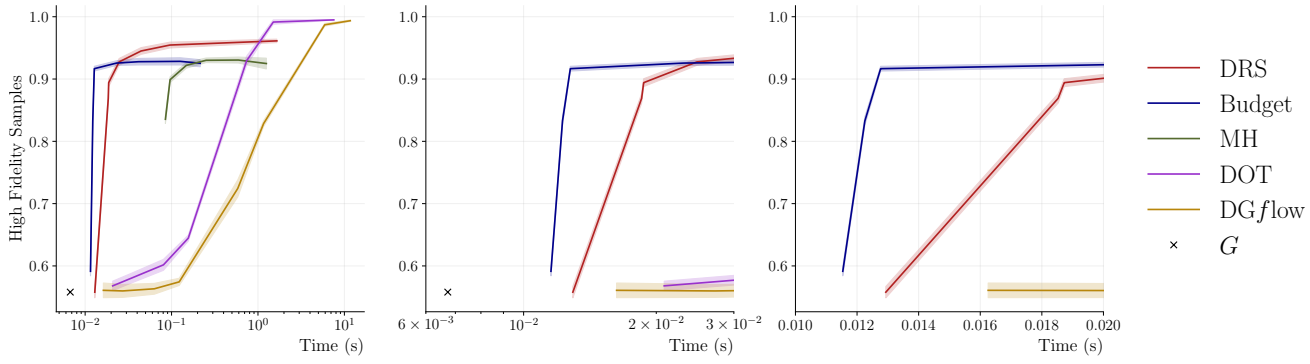


Figure D.3: How the different methods behave with regard to time: to achieve similar results, DOT and DGflow need 100 more times. MH only 10 times more. And for similar time (similar budget), Budgeted Reject is better than DRS. (blue above red). MH, DRS and OBRS (Budget) are caped. They only use the discriminator to refine samples, while the DOT and DGflow sample data point from the latent space and refines the samples directly using Gradient ascent.

As the distribution $\widehat{P}$ highly impacts how the rejection sampling methods behave, we also compare the OBRS and the DRS methods for different budgets and different $\widehat{P}$. In Figure D.4, we observe that the precision of the

OBRS is systematically better than the DRS. The distribution and the budget set in the experiment illustrated in Figures D.2 are set compare the methods for similar acceptance rates.

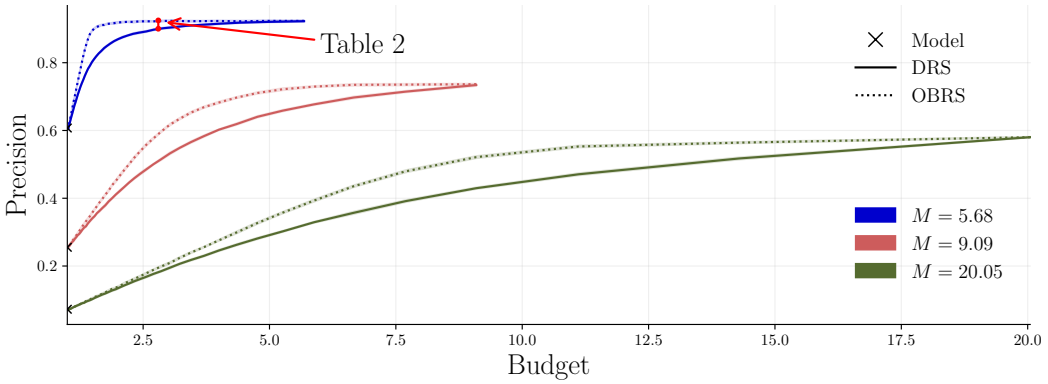


Figure D.4: Precision for different budget in various 2D datasets.

D.3 Complexity of Algorithm 2

In Algoritm 2, between every update of T and G, the parameter c_K is updated. In practice, the parameter c_K is not update every iterations. In this section, we investigate our the frequency of update affect the training procedure both in convergence speed and in terms of time.

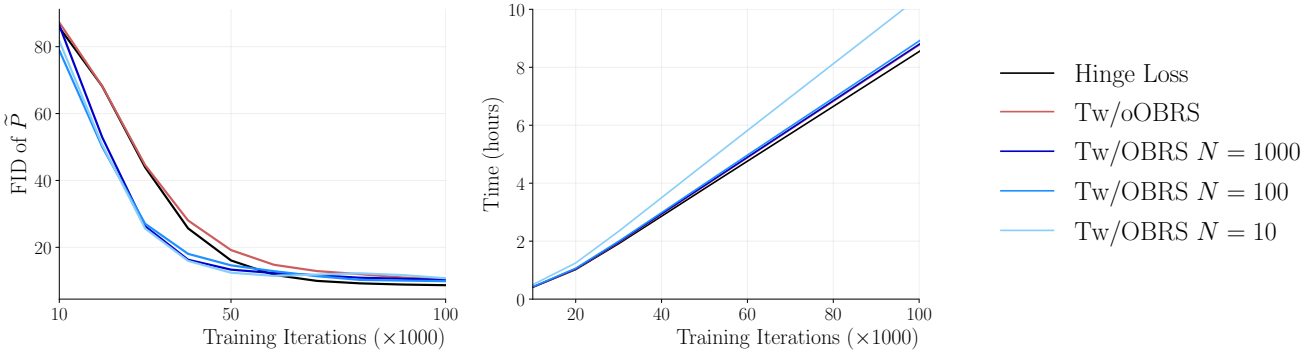


Figure D.5: Tw/OBRS: Training BigGAN models on CIFAR-10 with the hinge loss, and $\mathcal{D}_{\text{GAN}}$ without OBRS (Tw/oOBRS) and with OBRS (Tw/OBRS). For the models trained with OBRS, the parameter c_K is updated every N iterations.

We train different BigGAN models on CIFAR-10 with different frequency of updates: every 10 iterations, every 100 iterations and every 1000 iterations. We plot in Figure D.5, the FID during training and the time during training, both as a function of the number of iterations. We observe that the frequency of updates does not affect the speed of convergence. Furthermore, we observe that update c_K every 10 operation takes on average 19% longer to train than $\mathcal{D}_{\text{GAN}}$ without OBRS, while updating every 100 and 1000 iterations are only 1.69% and 0.03% longer.

E Experimental Procedure

In this paper, OBRS have been investigated in two different contexts.

- Using OBRS to improve a pre-trained model, with different budget.
- Training and fine-tuning a model accounting for OBRS with a budget of $K = 5$.

In every experiment, we have used BigGAN models Brock et al. (2019). For every dataset we have used hyper-parameters as close as possible to the original ones. In the original paper, the hinge loss is used and, according to Azadi et al. (2019), the fact that the loss is saturating decreases the performance of the estimation of the density ratio. In the first context, we take a pretrained model, typically trained with hinge loss, and fine-tune the discriminator only, based on $\mathcal{D}_{\text{GAN}}$. And thus we can perform density estimation for the rejection sampling procedure. In the DRS method, they retrain the discriminator on 10k samples. We opt for training the discriminator on the entire data set with a learning rate of 10^{-10} with the same hyperparameters as the one proposed by Brock et al. (2019).

Then for the second context: Tw/OBRS, we need to compare the speed of convergence for three different losses: hinge loss (since it is the original one), $\mathcal{D}_{\text{GAN}}(P\|\widehat{P})$ (Tw/oOBRS) and $\mathcal{D}_{\text{GAN}}(P\|\widetilde{P})$ (Tw/OBRS). However, we evaluate the models based their rejected distribution in terms of FID to analyze the speed of convergence. Therefore, two tails for the discriminator were built: one was trained with any loss and the other systematically with $\mathcal{D}_{\text{GAN}}$. Therefore, we can train the model G based on the given loss and still evaluate the model with OBRS. For the training with OBRS we used this set of hyperparameters. Every model has been trained on a 4xV100 clusters.

Dataset	Task	Tch	Gch	lr T	lr G	Batch Size
CIFAR-10	Training	64	64	2.10^{-5}	2.10^{-5}	50
CelebA64	Training	32	32	1.10^{-4}	4.10^{-4}	128
CelebA64	Fine Tuning	32	32	1.10^{-6}	1.10^{-6}	128
ImageNet128	Fine Tuning	96	96	1.10^{-5}	1.10^{-5}	2048

Table E.1: Hyper-parameters used for the different BigGAN configurations. Tch and Gch stands for the number of channels in each model. T lr and G lr stands for the learning rate of each models.