Quality and Diversity in Generative Models through the lens of f-divergences

Alexandre Vérine

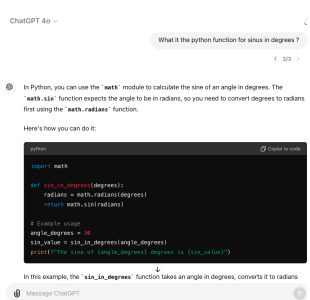
Centre Sciences des Données
DI, École Normale Supérieure PSL

Generative Models

Context

In the past few years, generative models have made significant progress in various domains of application.

Text Generation



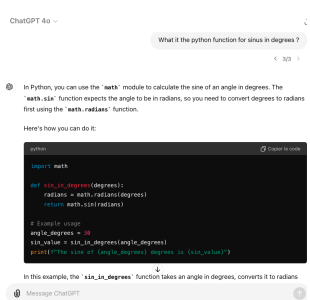
eg: Code Completion

Generative Models

Context

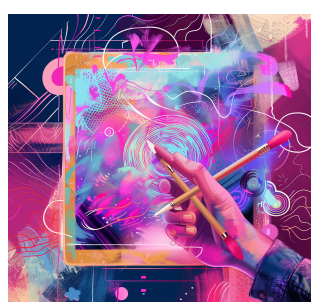
In the past few years, generative models have made significant progress in various domains of application.

Text Generation



eg: Code Completion

Image/Video Generation

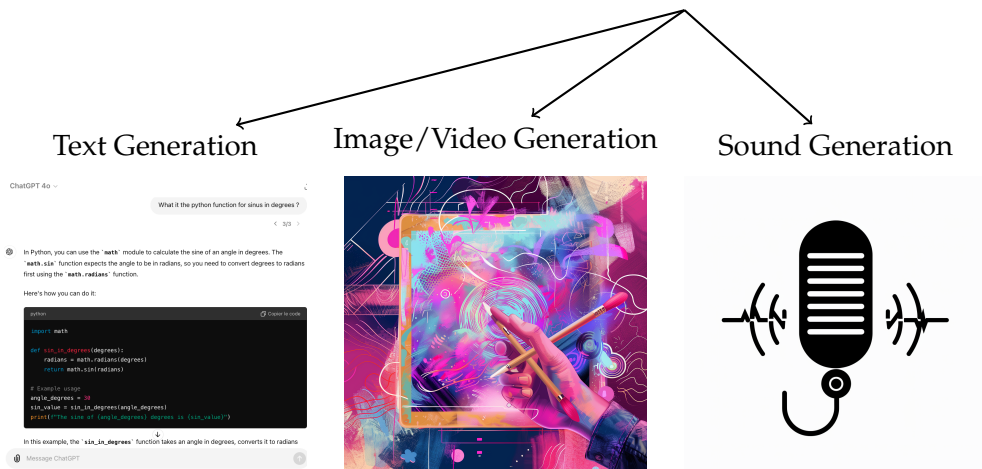


eg: Media Industry

Generative Models

Context

In the past few years, generative models have made significant progress in various domains of application.



eg: Code Completion

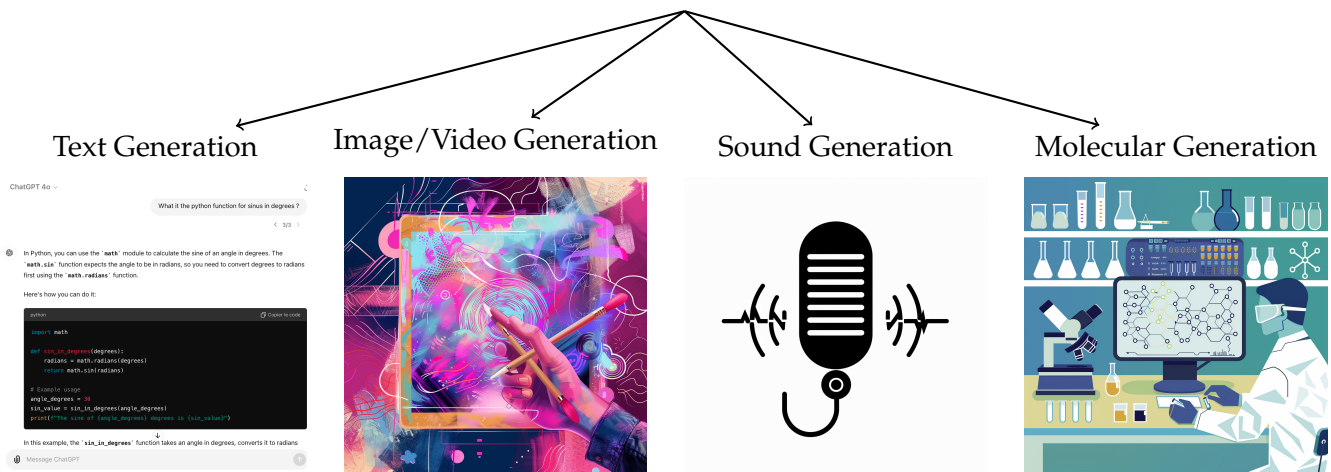
eg: Media Industry

eg: Speech Synthesis

Generative Models

Context

In the past few years, generative models have made significant progress in various domains of application.



eg: Code Completion | eg: Media Industry | eg: Speech Synthesis | eg: Drug Discovery

The various performances of Generative Models

Motivation

As the generation becomes better, the evaluation becomes more challenging.

DALL·E 2 (2023)

?

Midjourney v5 (2023)

Prompt: *A dog playing with a child.*

The various performances of Generative Models

Motivation

As the generation becomes better, the evaluation becomes more challenging.

DALL·E 2 (2023)

$\neq$

Midjourney v5 (2023)

Prompt: *A dog playing with a child.*

Outline

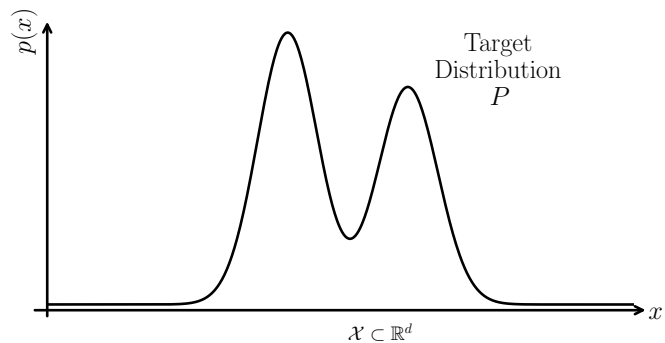
In this presentation, we discuss on evaluating, optimizing and improving quality and diversity of generative models:

1. **Evaluating:** How can we assess quality and diversity independently in Generative Models?
2. **Optimizing:** Can we optimize a specific trade-off between quality and diversity?
3. **Improving:** How can we improve the quality and diversity of a pre-trained generative models?

Evaluating:
How can we assess quality and diversity independtly in Generative Models?

Generative Models

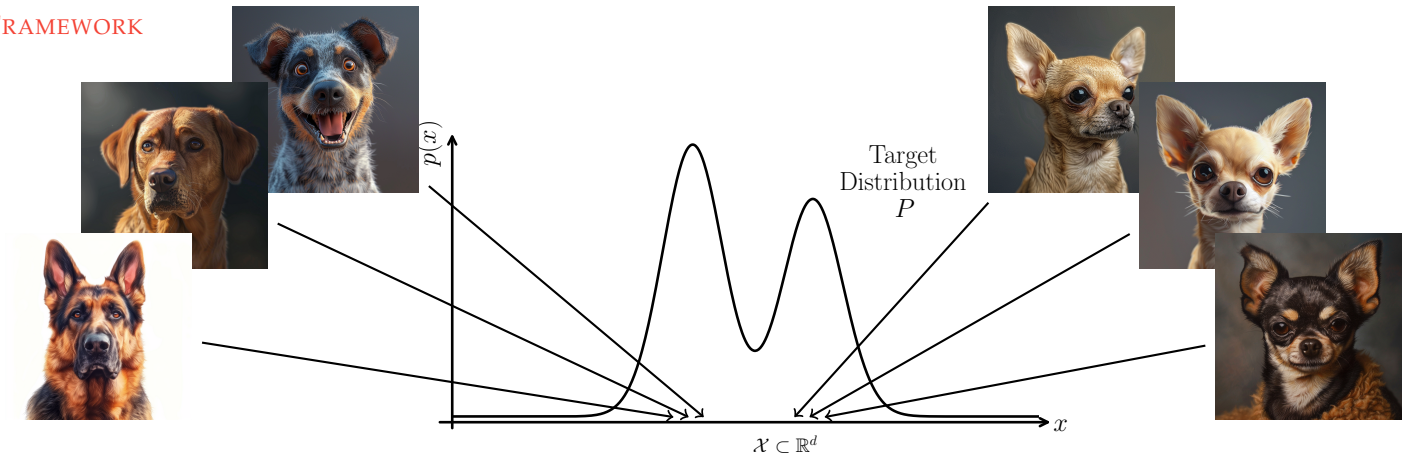
Framework



- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.

Generative Models

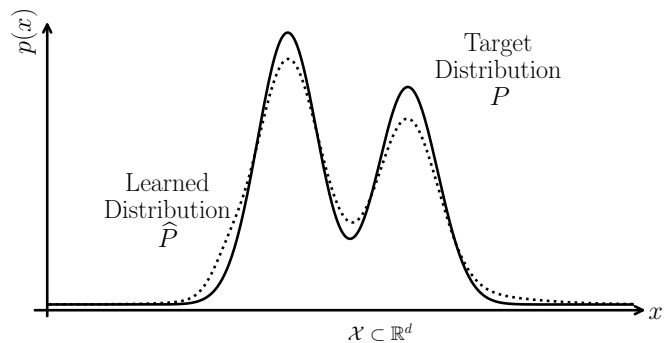
Framework



- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.

Generative Models

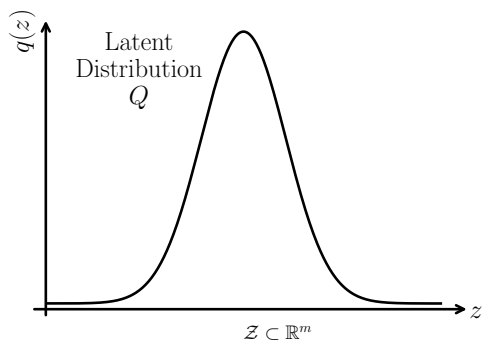
Framework

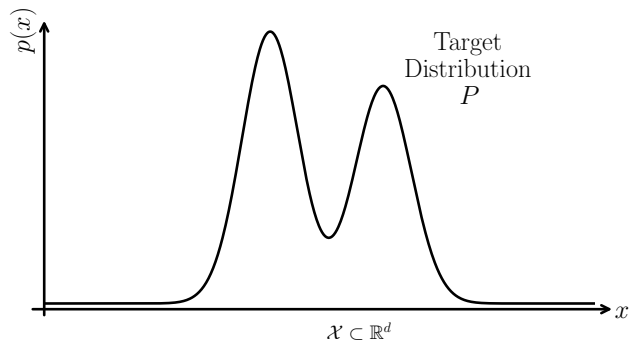


- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.
- Goal: Learn a *parameterized distribution* $\widehat{P}$ that approximate P:

Generative Models

Framework

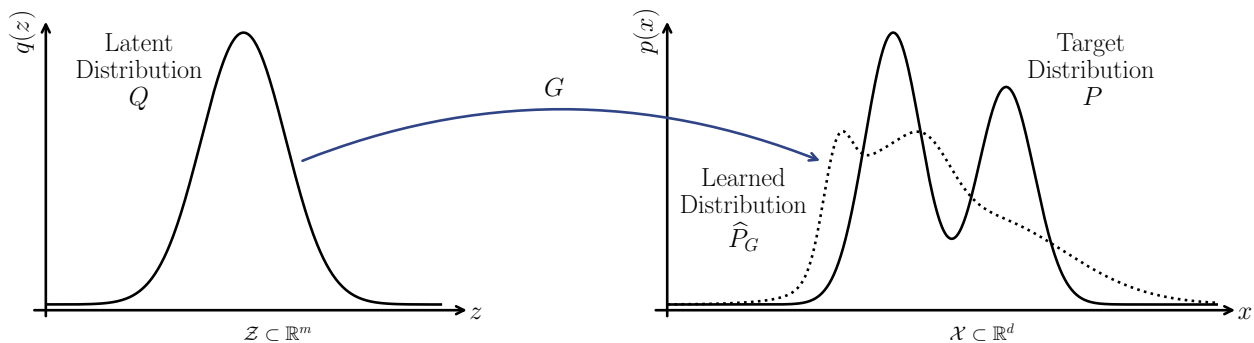




- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.
- Goal: Learn a *parameterized distribution* $\widehat{P}$ that approximate P:
  1. Consider a distribution Q in *a latent space* $\mathcal{X} \subset \mathbb{R}^m$, usually $\mathcal{N}(0, I_m)$.

Generative Models

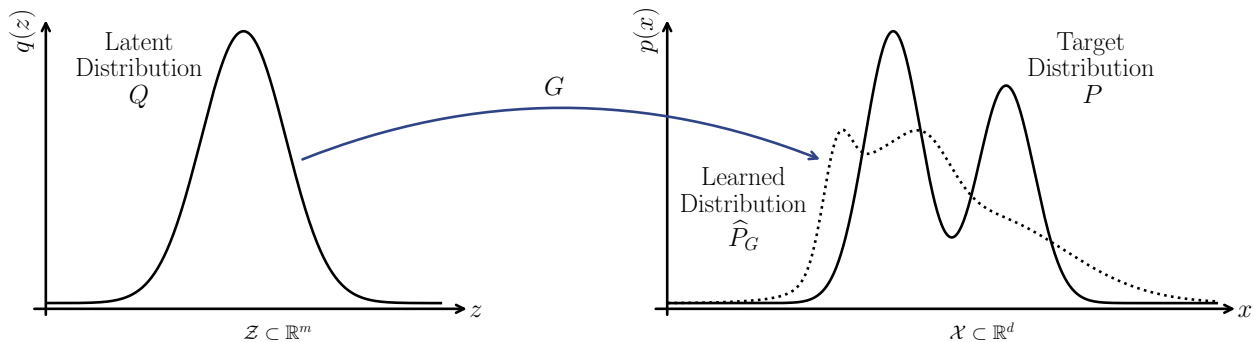
Framework



- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.
- Goal: Learn a *parameterized distribution* $\widehat{P}_G$ that approximate P:
  1. Consider a distribution Q in *a latent space* $\mathcal{X} \subset \mathbb{R}^m$, usually $\mathcal{N}(0, I_m)$.
  2. Take *a generator model* G represented by a neural network. Take $\widehat{P}_G = G\#Q$.

Generative Models

Framework

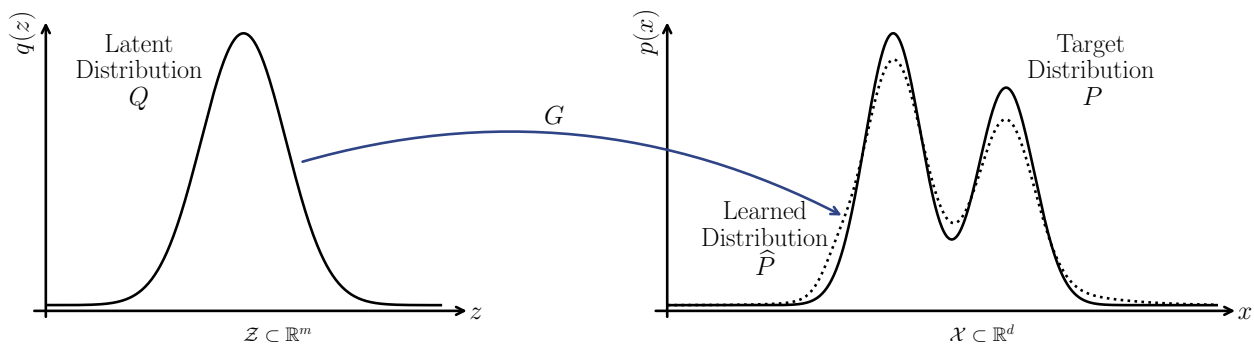


- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.
- Goal: Learn a *parameterized distribution* $\widehat{P}_G$ that approximate P:
  1. Consider a distribution Q in *a latent space* $\mathcal{X} \subset \mathbb{R}^m$, usually $\mathcal{N}(0, I_m)$.
  2. Take *a generator model* G represented by a neural network. Take $\widehat{P}_G = G\#Q$.
  3. Compute G^{opt} that minimize *a dissimilarity measure* D between P and $\widehat{P}_G$:

$$G^{\text{opt}} = \underset{G}{\operatorname{argmin}}\, D(P, \widehat{P}_G)$$

Generative Models

Framework



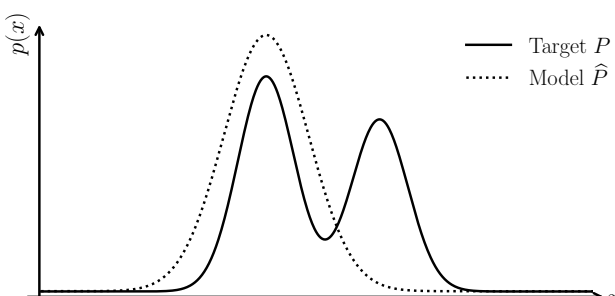
- Assumption: There is an unknown *target distribution* P in $\mathcal{X} \subset \mathbb{R}^d$.
- Goal: Learn a *parameterized distribution* $\widehat{P}$ that approximate P:
  1. Consider a distribution Q in *a latent space* $\mathcal{X} \subset \mathbb{R}^m$, usually $\mathcal{N}(0, I_m)$.
  2. Take *a generator model* G represented by a neural network. Take $\widehat{P} = G\#Q$.
  3. Compute G^{opt} that minimize *a dissimilarity measure* D between P and $\widehat{P}$:

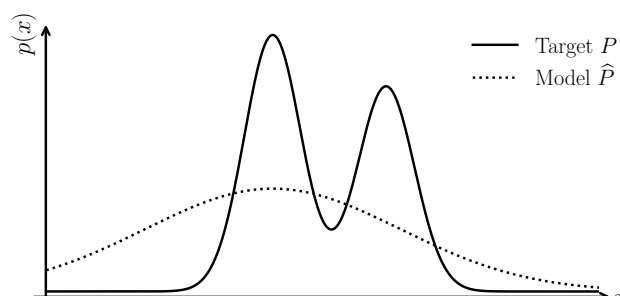
$$G^{\text{opt}} = \underset{G}{\operatorname{argmin}}\, D(P, \widehat{P})$$

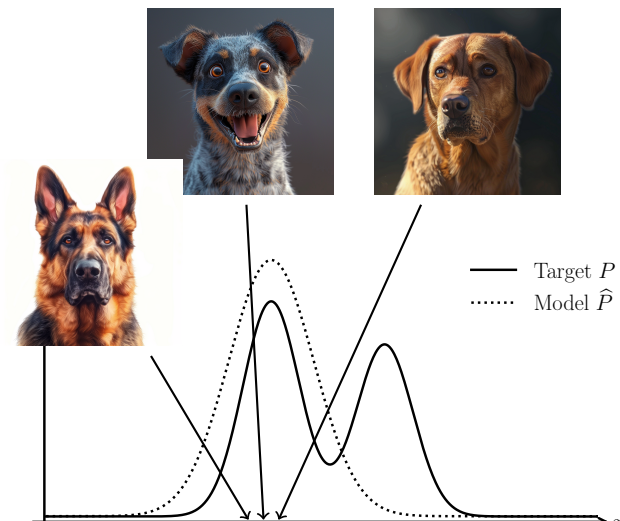
Generative Models

In Practice

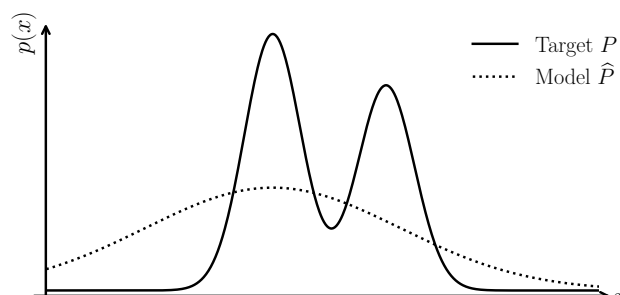
$$\widehat{P} \neq P$$







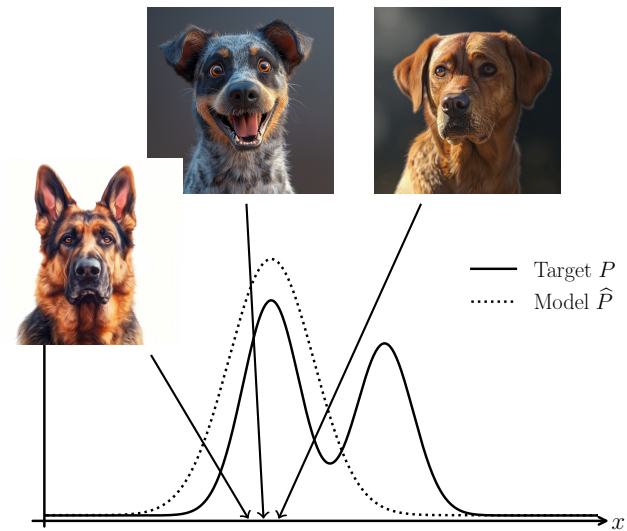
$\widehat{P} \neq P$



Low Diversity

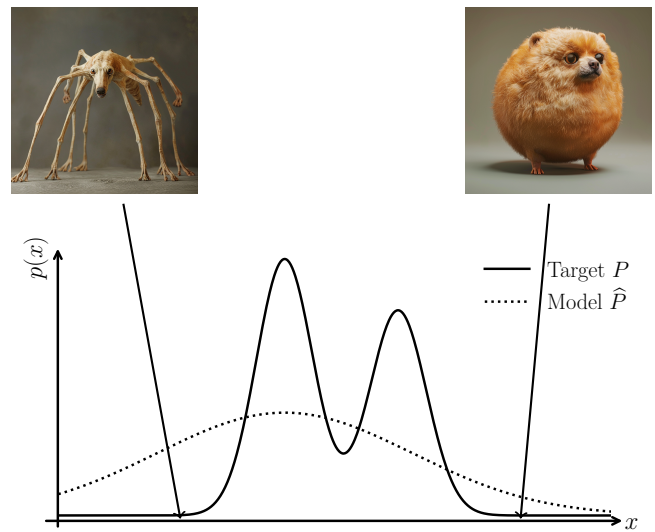
Generative Models

In Practice



Low Diversity

$\widehat{P} \neq P$



Low Quality

Precision and Recall for Generative Models

Metrics to evaluate quality and diversity

To assess models, we use the notion of Precision and Recall, inspired from Information Retrieval:

Quality

Diversity

To assess models, we use the notion of Precision and Recall, inspired from Information Retrieval:

Quality

↓

Precision

=

What proportion of generated samples are realistic?

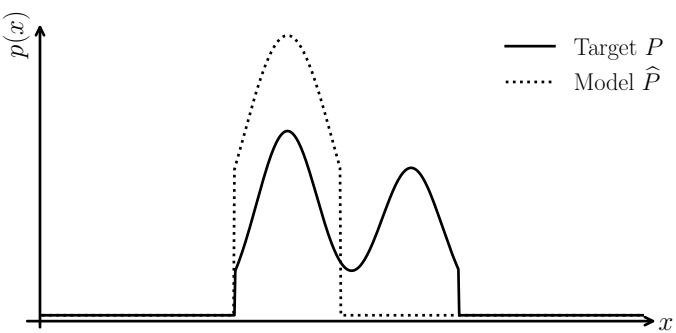
Diversity

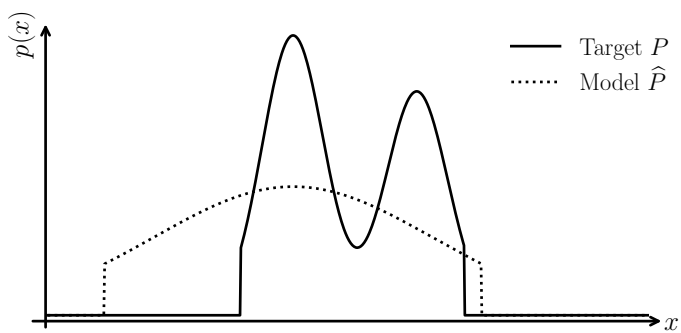
To assess models, we use the notion of Precision and Recall, inspired from Information Retrieval:

Quality	Diversity
↓	↓
Precision	Recall
=	=
What proportion of generated samples are realistic?	What proportion of real samples can be generated?

Precision and Recall for Generative Models

For finite support





Precision and Recall for Generative Models

For finite support

Definition 2.1 (Support-Based Precision and Recall - Kynkäänniemi et al. [8].)

For any distributions $P \in \mathcal{P}(\mathcal{X})$ and $\widehat{P} \in \mathcal{P}(\mathcal{X})$, we say that the distribution P has precision $\bar{\alpha}$ at recall $\bar{\beta}$ with respect to $\widehat{P}$ if

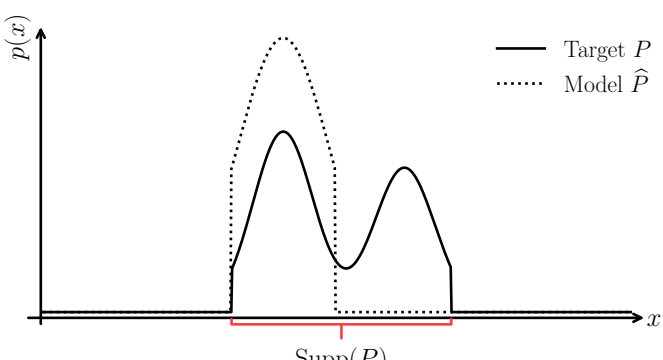
$$\bar{\alpha} := \widehat{P}(\text{Supp}(P)) \quad \textit{and} \quad \bar{\beta} := P(\text{Supp}(\widehat{P})). \tag{1}$$

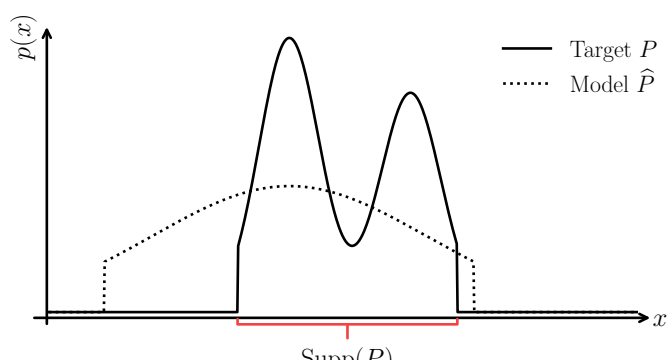
Precision and Recall for Generative Models

For finite support

Precision for finite support is the proportion of generated data that lies on the support of the real data:

$$\bar{\alpha} = \widehat{P}(\text{Supp}(P)).$$



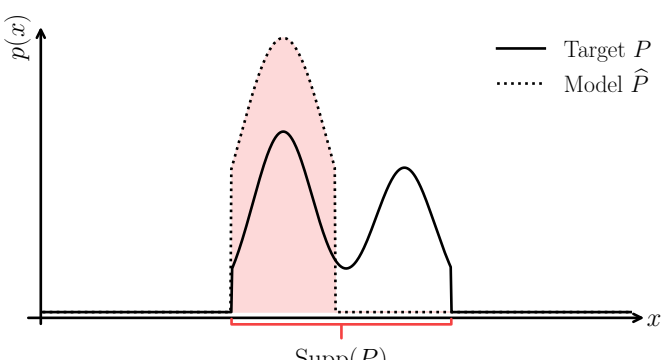


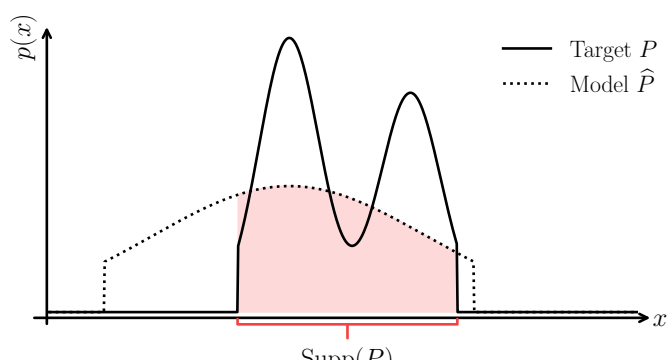
Precision and Recall for Generative Models

For finite support

Precision for finite support is the proportion of generated data that lies on the support of the real data:

$$\bar{\alpha} = \widehat{P}(\text{Supp}(P)).$$



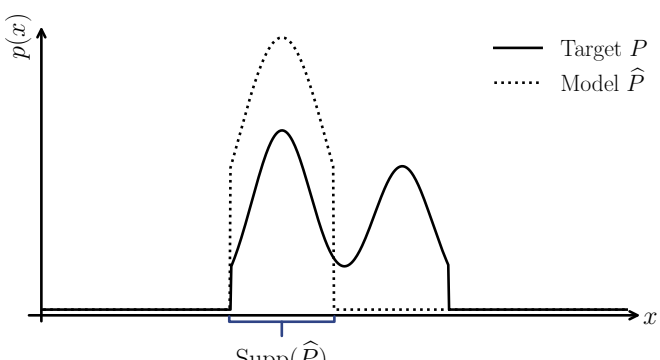


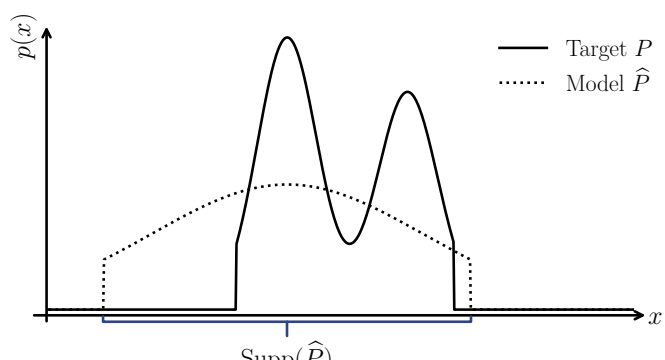
Precision and Recall for Generative Models

For finite support

Recall for finite support is the proportion of the support of the real data that is covered by the generated data:

$$\bar{\beta} = P(\text{Supp}(\widehat{P})).$$



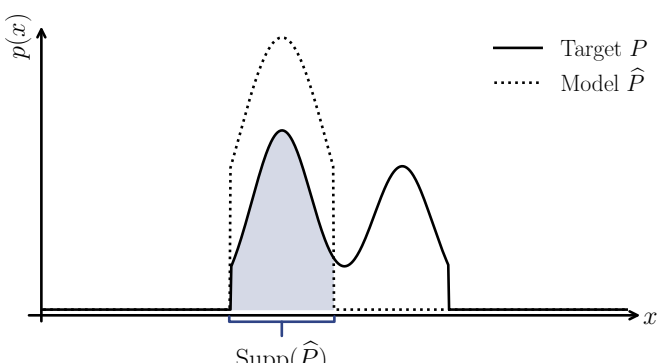


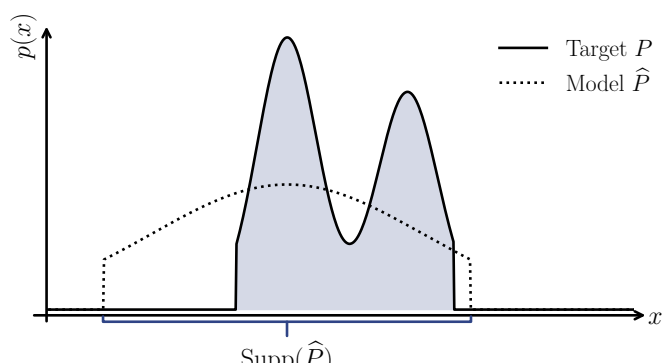
Precision and Recall for Generative Models

For finite support

Recall for finite support is the proportion of the support of the real data that is covered by the generated data:

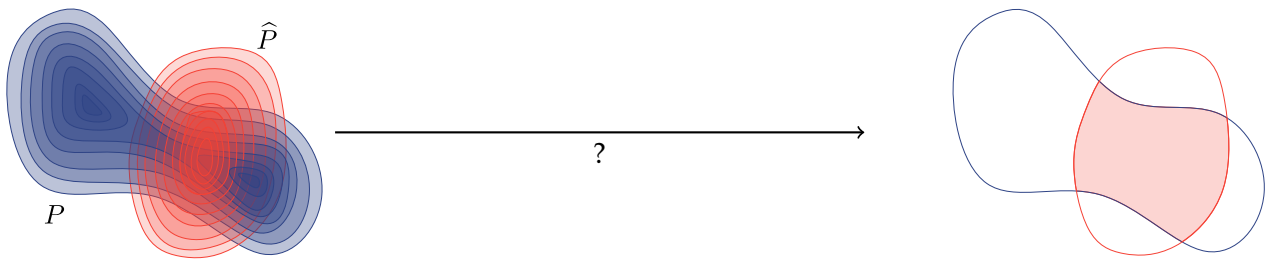
$$\bar{\beta} = P(\text{Supp}(\widehat{P})).$$





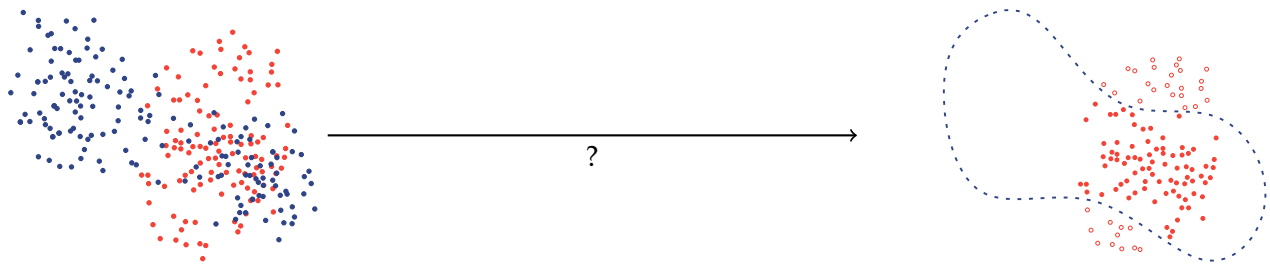
Precision and Recall for Generative Models

Estimating the Precision and Recall



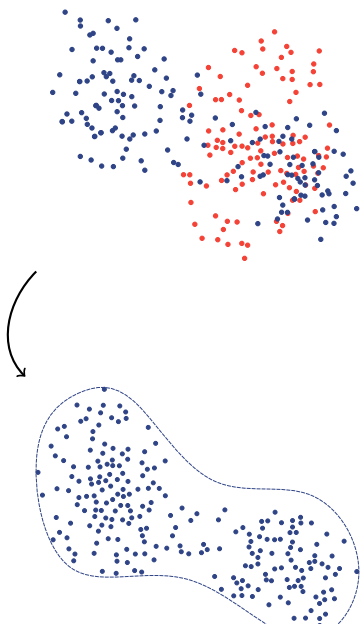
Precision and Recall for Generative Models

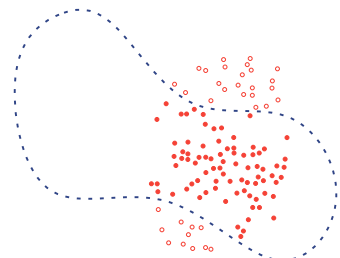
Estimating the Precision and Recall



Precision and Recall for Generative Models

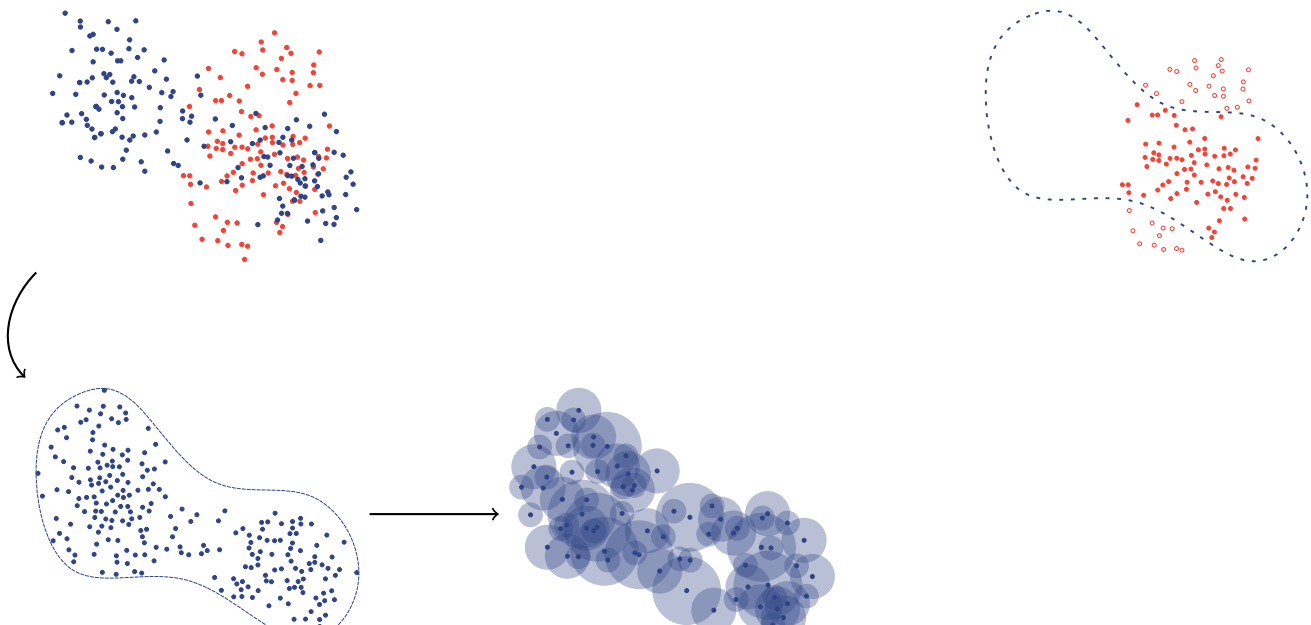
Estimating the Precision and Recall





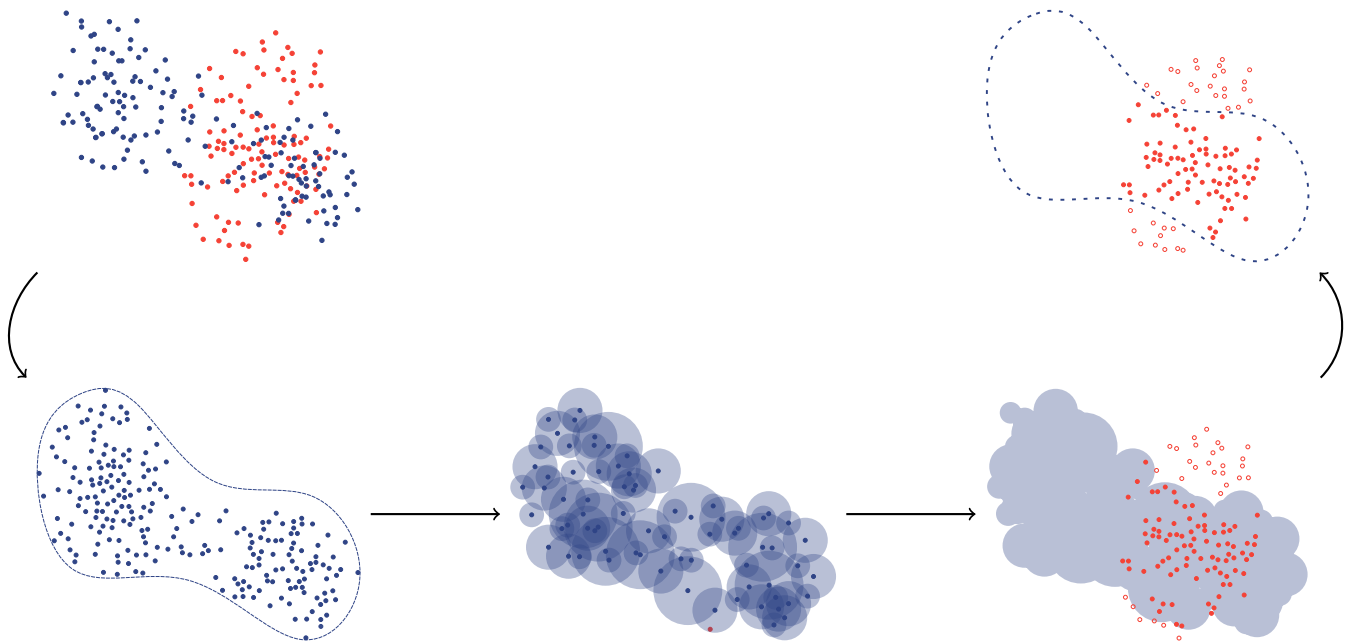
Precision and Recall for Generative Models

Estimating the Precision and Recall



Precision and Recall for Generative Models

Estimating the Precision and Recall



Precision and Recall for Generative Models

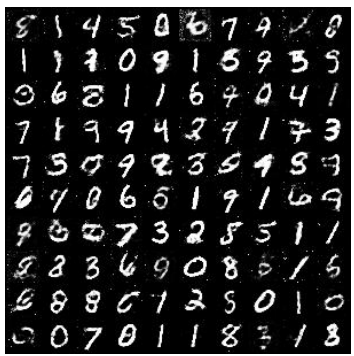
In Practice

MNIST Dataset [14]

Precision and Recall for Generative Models

In Practice

Low Precision High Recall



Precision: 0.54 Recall: 0.91

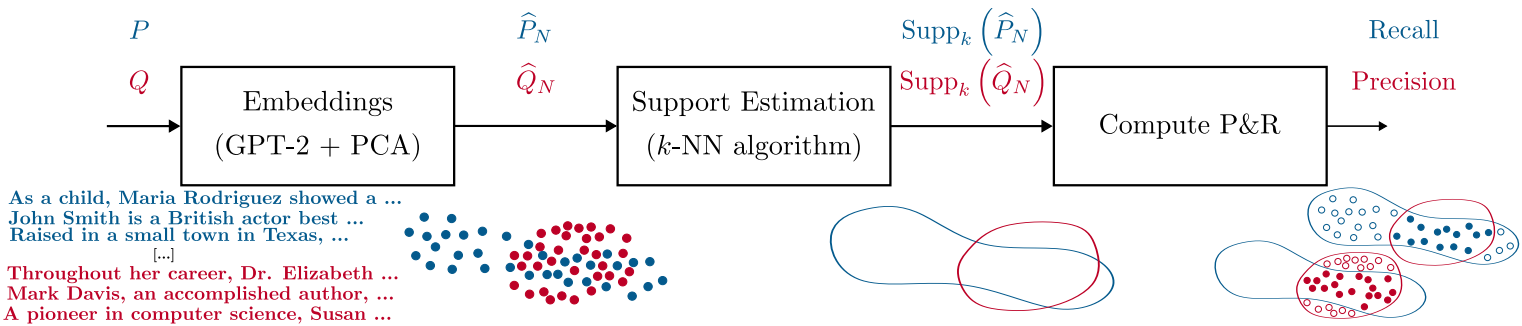
MNIST Dataset [14]

High Precision Low Recall

Precision: 0.80 Recall: 0.70

Precision and Recall for Generative Models

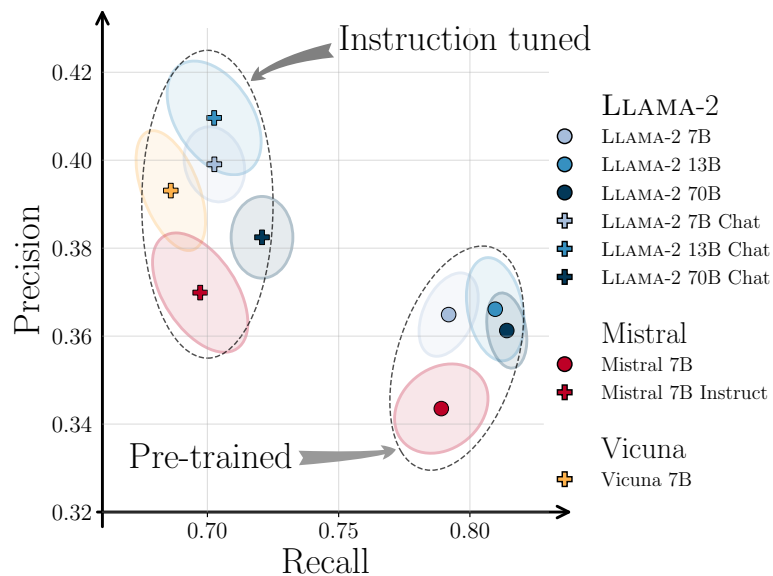
For LLMs



Precision and Recall for Generative Models

For LLMs

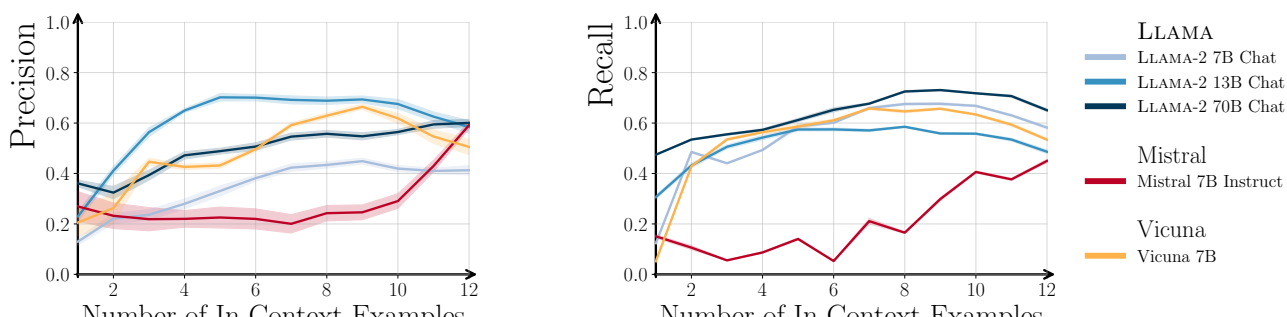
On open-ended generation, the quality and diversity of LLMs can be evaluated using Precision and Recall for instance on Webtext: Bronnec et al. [3]



Precision and Recall for Generative Models

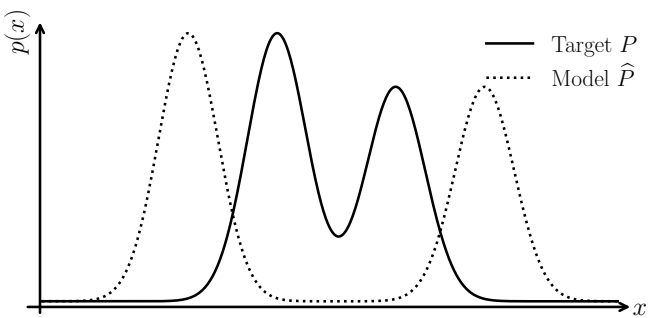
For LLMs

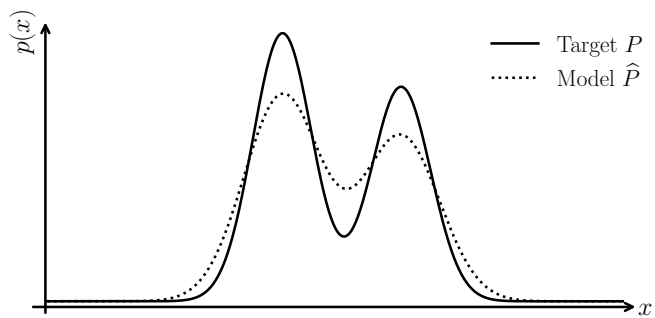
We can also evaluate the quality and diversity of LLMs on Chatbot open-ended generation. We can for instance check the impact of In-Context examples on the quality and diversity of the generated text. For instance on Wikipedia Biographies generation:



Precision and Recall for Generative Models

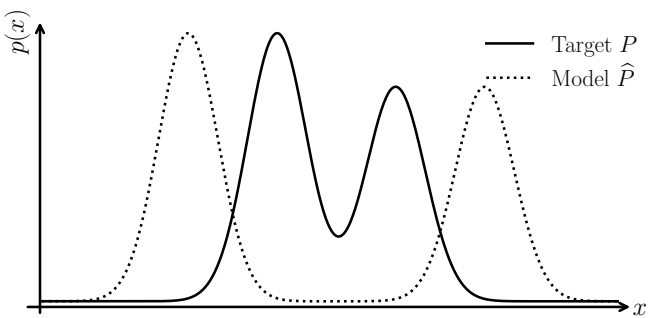
For inifinite support

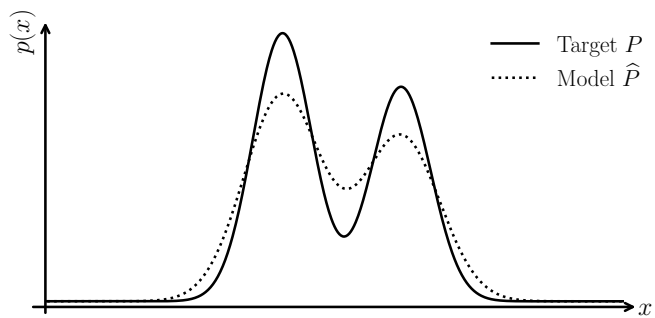




Precision and Recall for Generative Models

For inifinite support





Both distributions have **perfect** Precision *and* Recall.

PR-Curve for Generative Models

Definition

Definition 2.2 (PR-Curve for Generative Models - Sajjadi et al. [11], Simon et al. [12])

Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions such that $P, \widehat{P} \ll \mu$. The PR-Curve is the set $\mathrm{PRD}(P, \widehat{P})$ defined as:

$$\mathrm{PRD}(P, \widehat{P}) = \{(\alpha_\lambda, \beta_\lambda) \mid \lambda \in [0, \infty]\} \tag{2}$$

with:

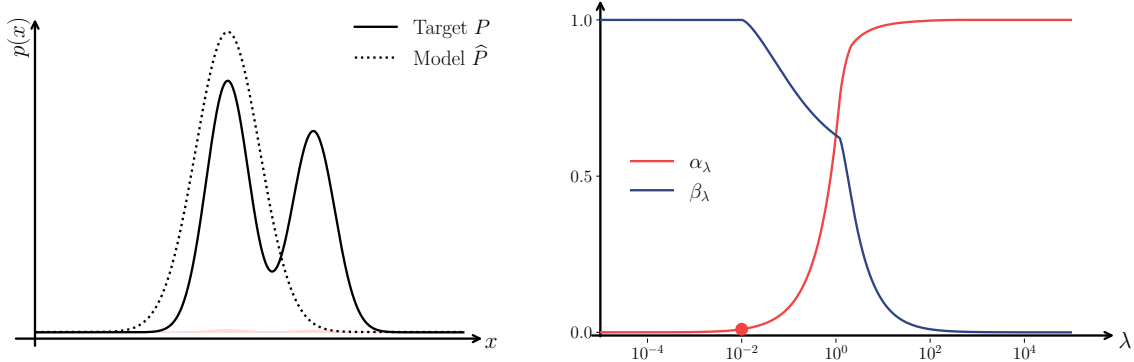
$$\alpha_\lambda = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \quad \textit{and} \quad \beta_\lambda = \int_{\mathcal{X}} \min\left(p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})/\lambda\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{3}$$

PR-Curve for Generative Models

Definition

For the Precision, λp is compared to $\widehat{p}$ for different threshold $\lambda \in [0, +\infty]$:

$$\alpha_\lambda = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{4}$$

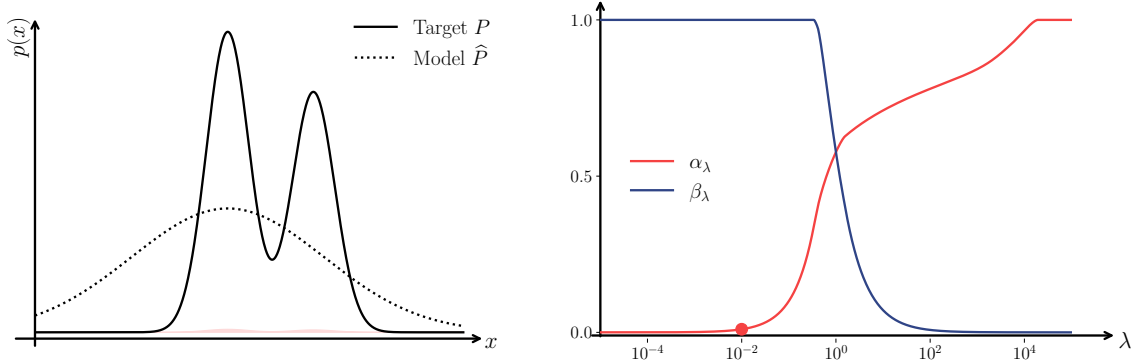


PR-Curve for Generative Models

Definition

For the Precision, λp is compared to $\widehat{p}$ for different threshold $\lambda \in [0, +\infty]$:

$$\alpha_\lambda = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{4}$$

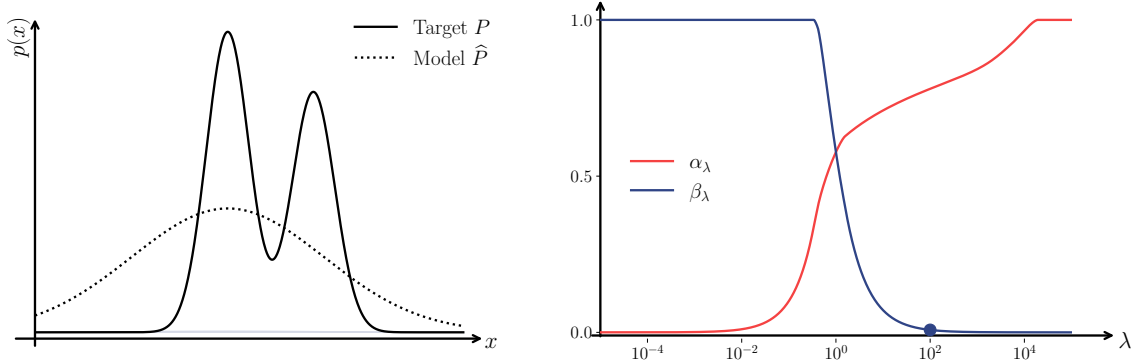


PR-Curve for Generative Models

Definition

For the Recall, p is compared to $\widehat{p}/\lambda$ for different threshold $\lambda \in [0, +\infty]$:

$$\beta_\lambda = \int_{\mathcal{X}} \min\left(p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})/\lambda\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{5}$$

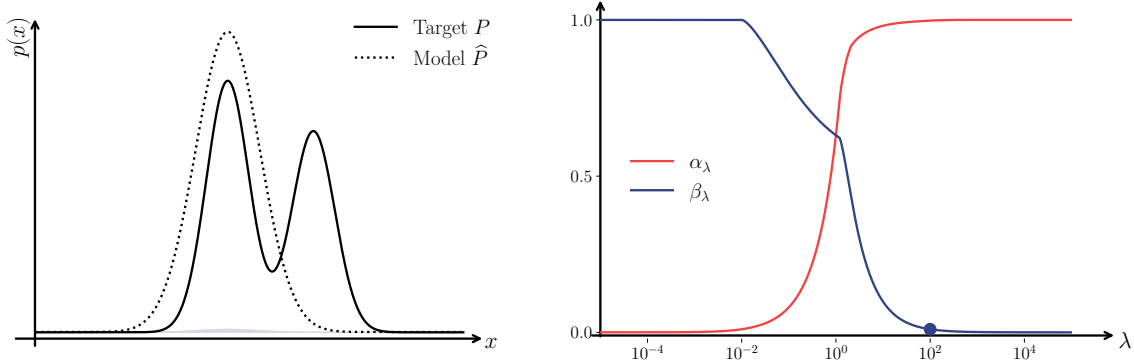


PR-Curve for Generative Models

Definition

For the Recall, p is compared to $\widehat{p}/\lambda$ for different threshold $\lambda \in [0, +\infty]$:

$$\beta_\lambda = \int_{\mathcal{X}} \min\left(p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})/\lambda\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{5}$$



PR-Curve for Generative Models

Examples

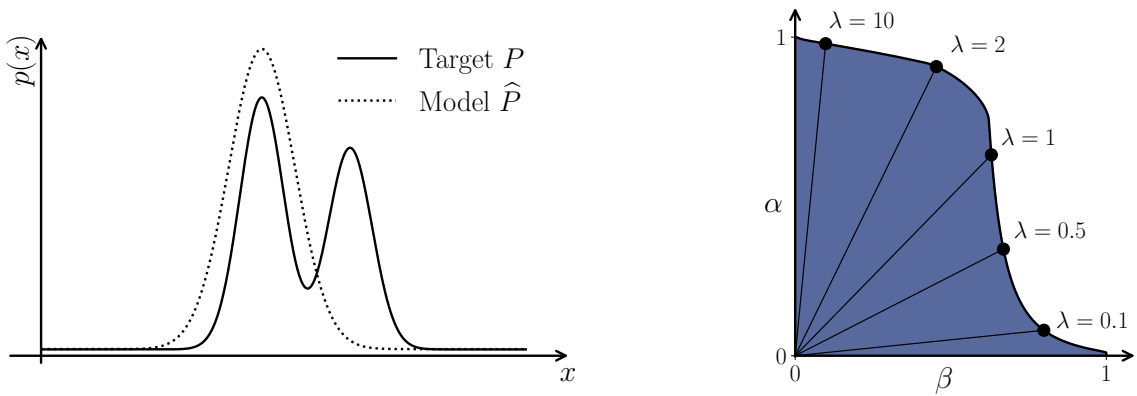


Figure. Learning distribution with low recall and high precision.

PR-Curve for Generative Models

Examples

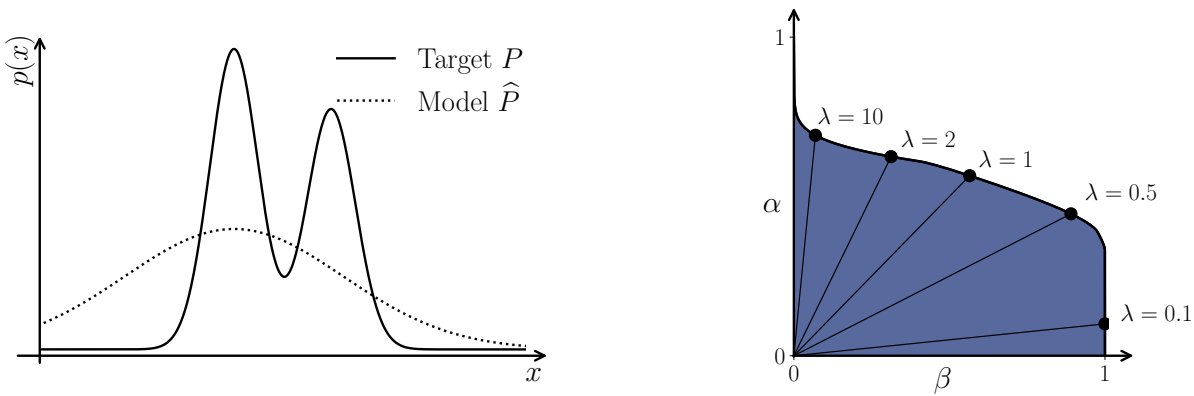


Figure. Learning distribution with high recall and low precision.

PR-Curve for Generative Models

Examples

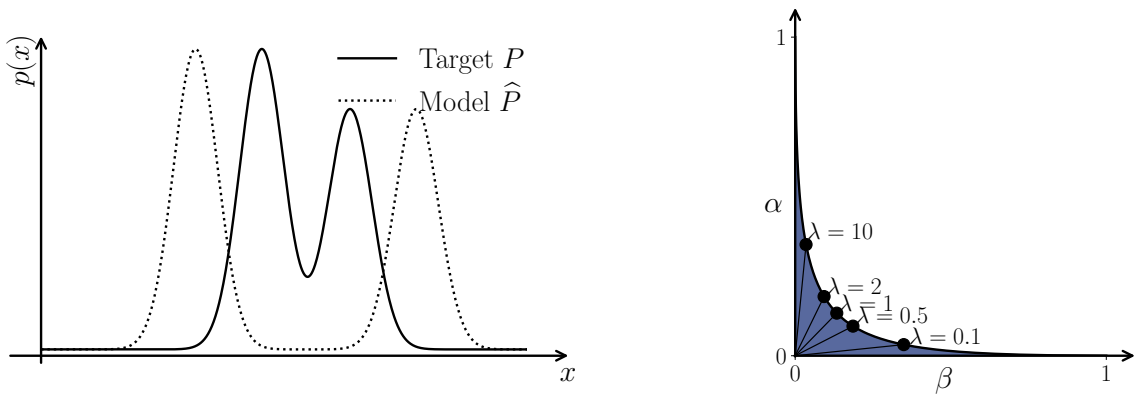


Figure. Learning distribution with low recall and low precision.

PR-Curve for Generative Models

Examples

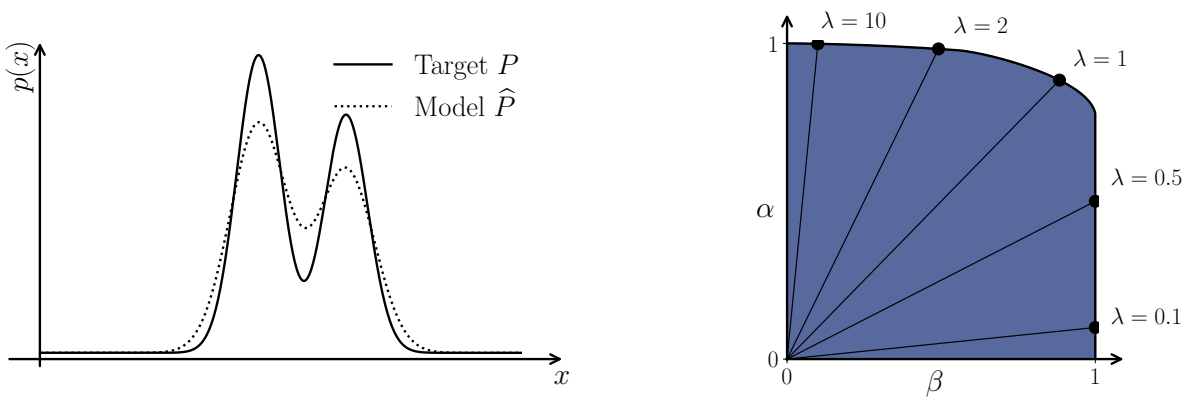


Figure. Learning distribution with high recall and high precision.

The PR-Curve is a generalization of the Precision and Recall for finite support:

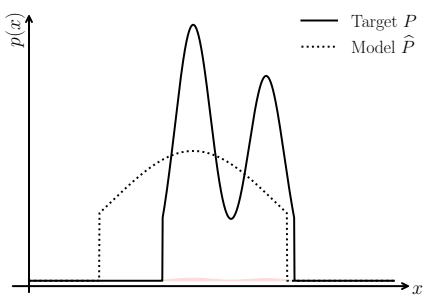
Theorem 2.3 (Support-based and PR-Curves - Siry et al. [13])

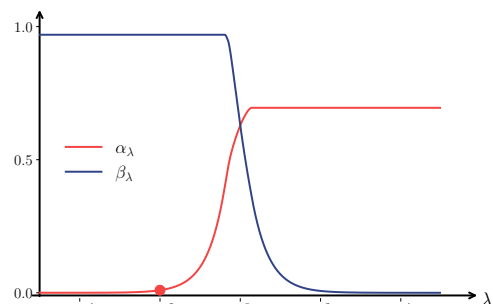
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Then, the support-based Precision and Recall $(\bar{\alpha}, \bar{\beta})$ are related to the PR-Curve values $\mathrm{PRD}(P, \widehat{P})$ *for $\lambda = 0$ and $\lambda = \infty$:*

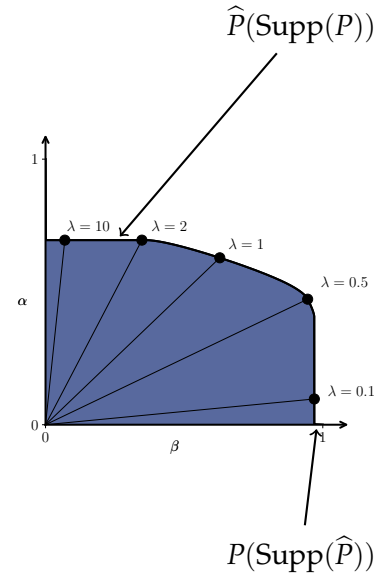
$$\bar{\alpha} = \max_{\lambda} \alpha_\lambda = \alpha_\infty \quad \textit{and} \quad \bar{\beta} = \max_{\lambda} \beta_\lambda = \beta_0. \tag{6}$$

PR-Curve and Support-Based Precision and Recall

Relation

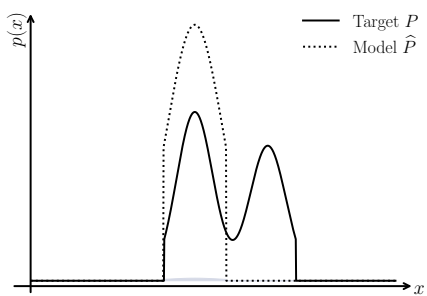


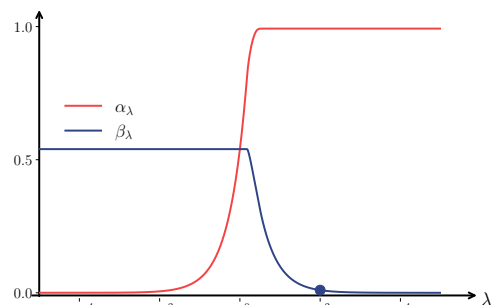


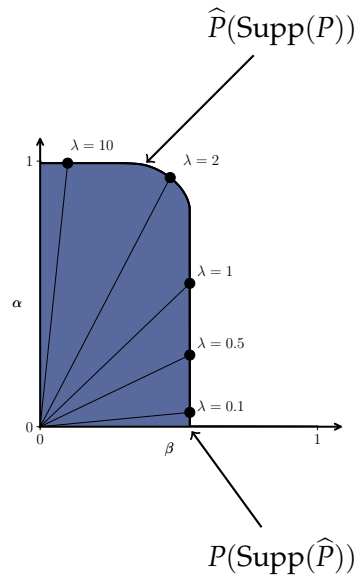


PR-Curve and Support-Based Precision and Recall

Relation



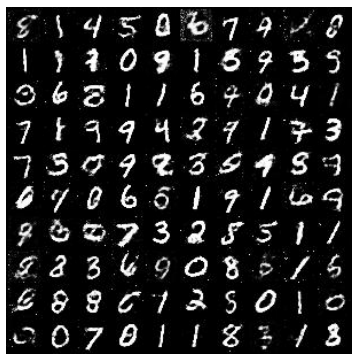




PR-Curve for Generative Models

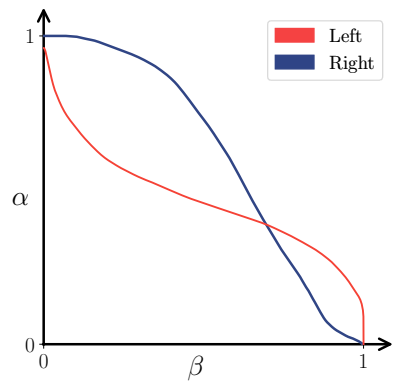
In Practice

Low Precision High Recall



Precision: 0.54 Recall: 0.91

PR-Curves



High Precision Low Recall

Precision: 0.80 Recall: 0.70

PR-Curve for Generative Models

In NLP

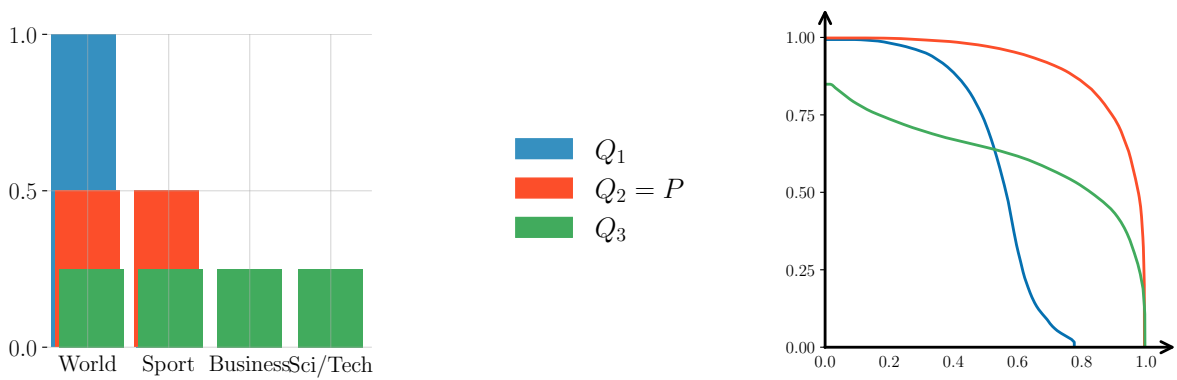


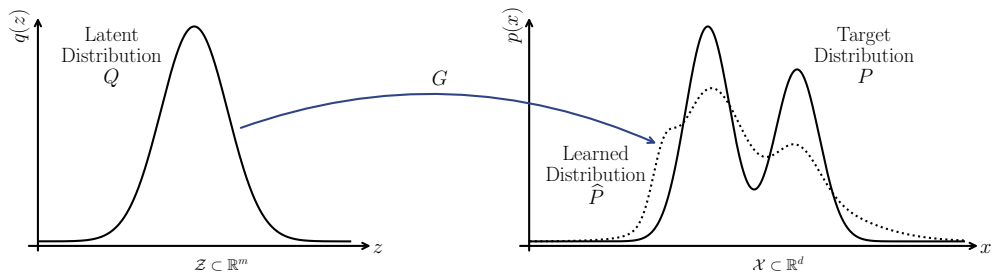
Figure. PR-Curve for distributions journal articles: AG News.

Optimizing:
Can we optimize a specific trade-off between Precision and Recall?

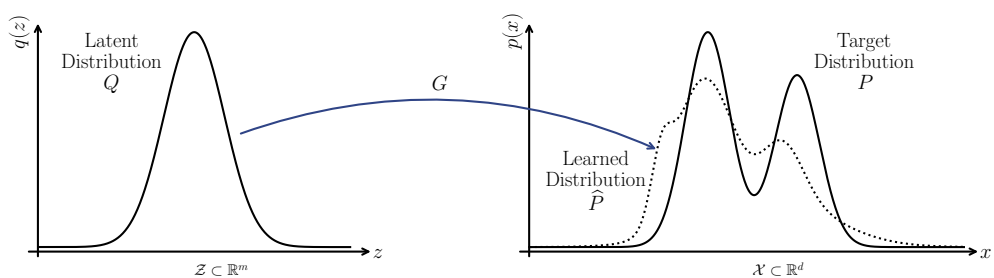
Tuning Precision and Recall in Generative Models

Truncation

Hard Trunctation
Karras et al. [6]



Soft Trunctation
Kingma and Dhariwal [7]



Section 2: Optimizing

Hard Truncation

Figure. From left to right: $\psi = 0.0$, $\psi = 0.3$ $\psi = 0.7$ $\psi = 1.0$.

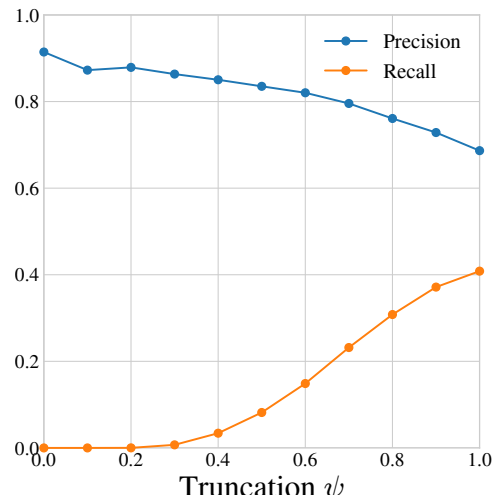


Figure. Source: [8]

(a) $\psi = 0.04$ (b) $\psi = 0.5$ (c) $\psi = 1.0$ (d) $\psi = 2.0$

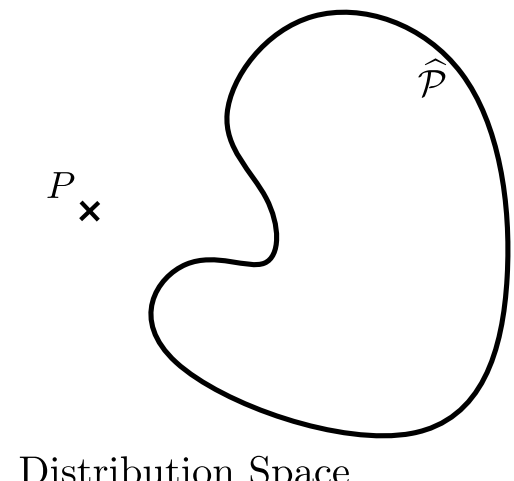
Figure. Soft-Truncation on BigGAN. Source:[2].

Training a Generative Model

in general

Traditionally, the goal is to minimize *a dissimilarity measure* between the target distribution P and the learned distribution $\widehat{P}$:

$$\min_{G} \; D(P, \widehat{P}_G) \tag{6}$$

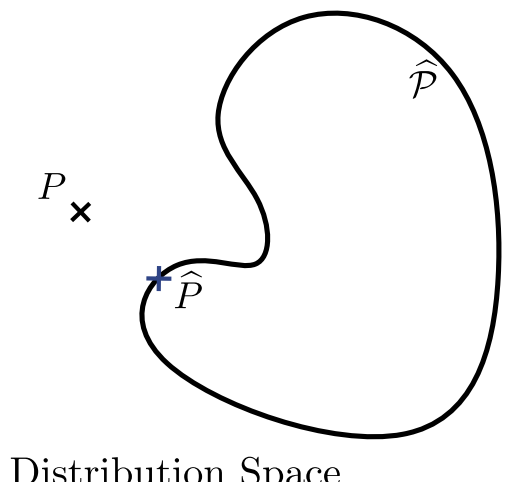


Training a Generative Model

with f-divergences

Traditionally, the goal is to minimize *an f-divergence* between the target distribution P and the learned distribution $\widehat{P}$:

$$\min_G \ \mathcal{D}_f(P \| \widehat{P}_G) \tag{6}$$



Definition 3.1 (f-divergences)

For any two probability distributions P and $\widehat{P}$ in $\mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$. Let p and $\widehat{p}$ be the Radon-Nikodym densities of P and $\widehat{P}$ with respect to μ, respectively. Let f be any convex lower semi-continuous function $f : [0, \infty] \to]-\infty, +\infty]$ such that $f(1) = 0$, the f-divergence between P and $\widehat{P}$ is

$$\mathcal{D}_f(P\|\widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{7}$$

Definition 3.1 (f-divergences)

For any two probability distributions P and $\widehat{P}$ in $\mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$. Let p and $\widehat{p}$ be the Radon-Nikodym densities of P and $\widehat{P}$ with respect to μ, respectively. Let f be any convex lower semi-continuous function $f : [0, \infty] \to]-\infty, +\infty]$ such that $f(1) = 0$, the f-divergence between P and $\widehat{P}$ is

$$\mathcal{D}_f(P\|\widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{7}$$

Usual divergences are f-divergences:

- Kullback-Leibler (KL),
- Reverse Kullback-Leibler (rKL),
- Jensen-Shannon (JS),
- Total Variation (TV),
- $\alpha-$divergences.

Estimating f-divergences

Dual variational form

f-divergences are *hardly tractable* in practice. However, they can be approximated by a dual approximation.

Estimating f-divergences

Dual variational form

f-divergences are *hardly tractable* in practice. However, they can be approximated by a dual approximation.

- $f^*(t) = \sup_{u \in \mathbb{R}} \{tu - f(u)\}$ be *the Fenchel conjugate* of f.
- $\mathcal{T}$ be the set of all measurable functions $\mathcal{X} \to \mathbb{R}$.

Estimating f-divergences

Dual variational form

f-divergences are *hardly tractable* in practice. However, they can be approximated by a dual approximation.

- $f^*(t) = \sup_{u \in \mathbb{R}} \{tu - f(u)\}$ be *the Fenchel conjugate* of f.
- $\mathcal{T}$ be the set of all measurable functions $\mathcal{X} \to \mathbb{R}$.

Theorem 3.2 (Dual variational form of an f-divergence- Nguyen et al. [9])

Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ two distributions such that P is absolutely continuous with respect to $\widehat{P}$ and f a suitable generator function. The f-divergence between P and $\widehat{P}$ admits a dual variational form:

$$\mathcal{D}_f(P \| \widehat{P}) = \sup_{T \in \mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x} \sim P} [T(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}} [f^*(T(\boldsymbol{x}))] \right). \tag{8}$$

We use $T^{\text{opt}} \in \mathcal{T}$ to denote the function that achieves the supremum.

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \underbrace{\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right]}_{\mathcal{D}_{f,T}^{\text{dual}}} \tag{9}$$

Training a Generative Model with f-divergences

Using the dual variational form

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \underbrace{\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right]}_{\mathcal{D}_{f,T}^{\text{dual}}} \tag{9}$$

- The discriminator T is trained *to estimate* the divergence.
- The generator G is trained *to minimize* the divergence.

Training a Generative Model with f-divergences

Using the dual variational form

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \underbrace{\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right]}_{\mathcal{D}_{f,T}^{\text{dual}}} \tag{9}$$

- The discriminator T is trained *to estimate* the divergence.
- The generator G is trained *to minimize* the divergence.

Training a Generative Model with f-divergences

Using the dual variational form

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \; \mathbb{E}_{\boldsymbol{x}\sim P}\left[\log\left(D(\boldsymbol{x})\right)\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*\left(\log(D(\boldsymbol{x}))\right)\right] \tag{9}$$

- The discriminator T is trained *to estimate* the divergence.
- The generator G is trained *to minimize* the divergence.
- With $T(\boldsymbol{x}) = \log(D(\boldsymbol{x}))$ with $D(\boldsymbol{x}) \in [0, 1]$.

Training a Generative Model with f-divergences

Using the dual variational form

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \mathbb{E}_{\boldsymbol{x}\sim P}\left[\log\left(D(\boldsymbol{x})\right)\right] + \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[\log\left(1 - D(\boldsymbol{x})\right)\right] \tag{9}$$

- The discriminator T is trained *to estimate* the divergence.
- The generator G is trained *to minimize* the divergence.
- With $T(\boldsymbol{x}) = \log(D(\boldsymbol{x}))$ with $D(\boldsymbol{x}) \in [0, 1]$.
- $f^*(t) = f^*_{\text{JS}}(t) = -\log(1 - \exp(t))$ for the Jensen-Shannon divergence.

We recover the original GAN framework.

Training a Generative Model with f-divergences

Using the dual variational form

By doing so, we can rewrite the optimization problem as:

$$\min_G \max_T \underbrace{\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right]}_{\mathcal{D}_{f,T}^{\text{dual}}} \tag{9}$$

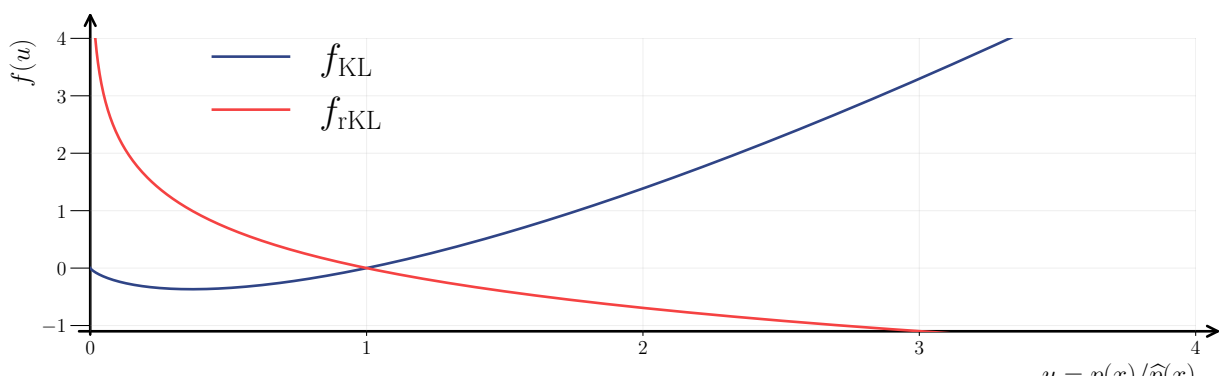
- The discriminator T is trained *to estimate* the divergence.
- The generator G is trained *to minimize* the divergence.

- Generative Adversarial Networks [4] for *the Jensen-Shannon divergence*.
- Extended to other f-divergences by Nowozin et al. [10].
- Extend to other generative models such as Normalizing Flows by Grover et al. [5].

Effect of the f-divergence on the learned distribution

All f-divergences are not equal:

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

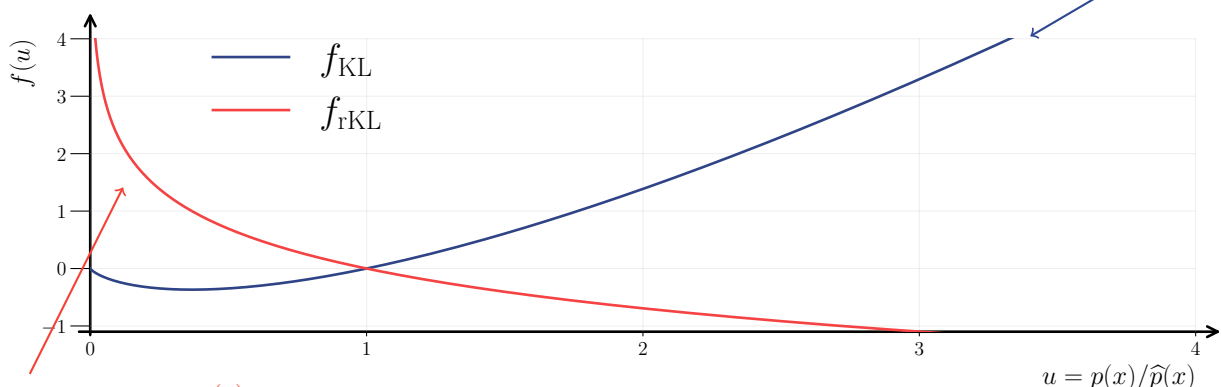


Effect of the f-divergence on the learned distribution

All f-divergences are not equal:

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

Penalizing high values of $\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$



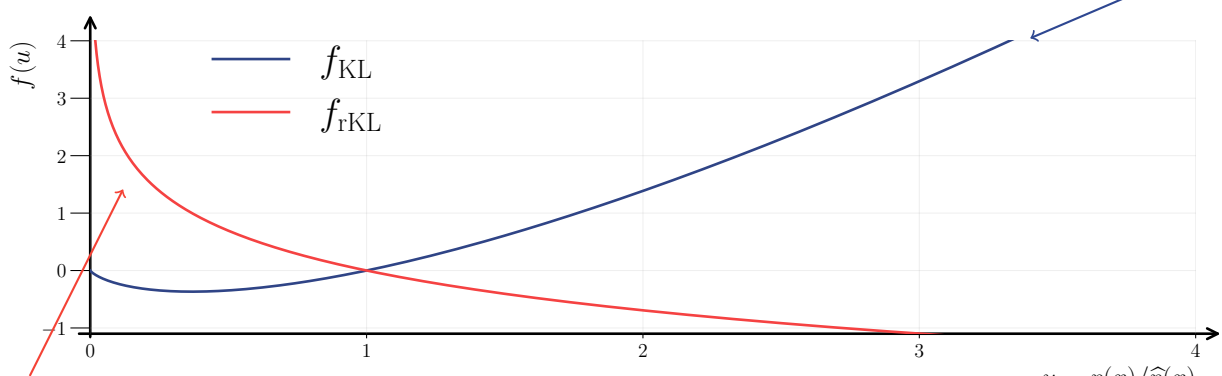
Penalizing low values of $\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$

Effect of the f-divergence on the learned distribution

All f-divergences are not equal:

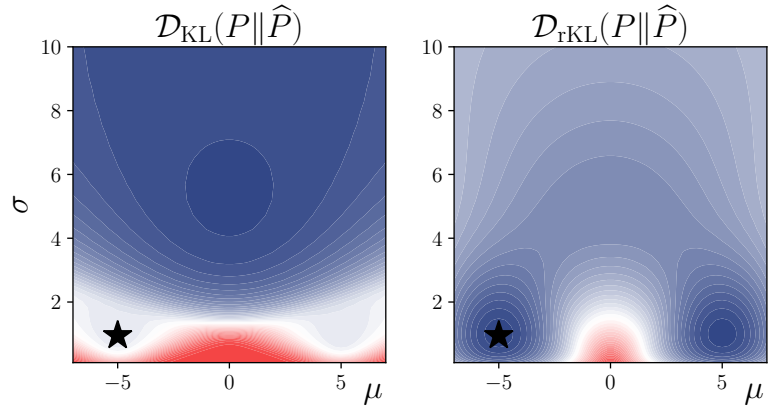
$$\mathcal{D}_f(P\|\widehat{P}) = \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right]$$

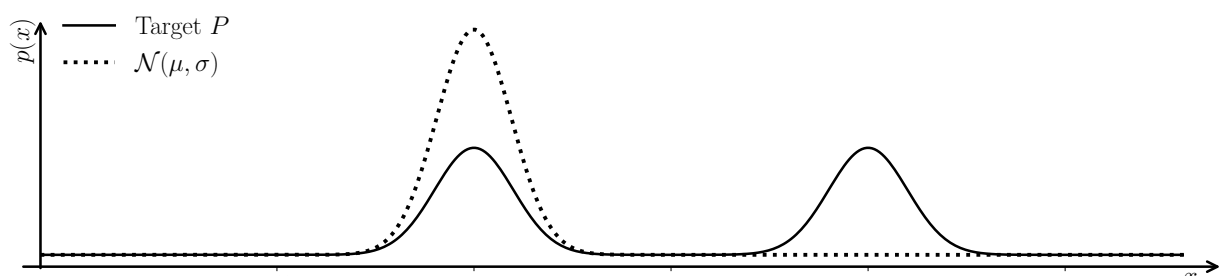
Penalizing high values of $\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$
Favors high recall



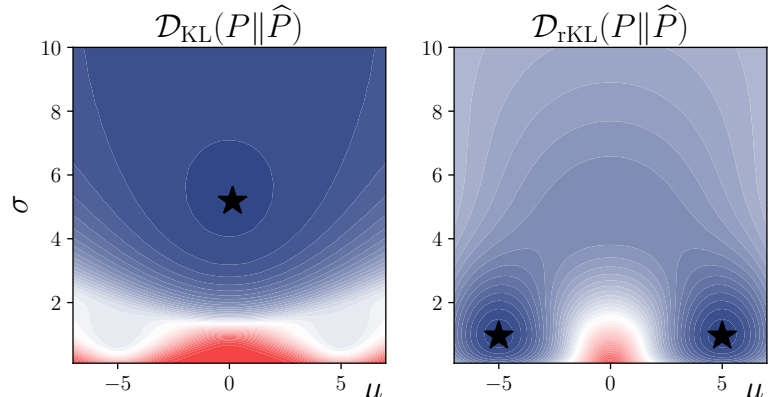
Penalizing low values of $\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$
Favors high precision

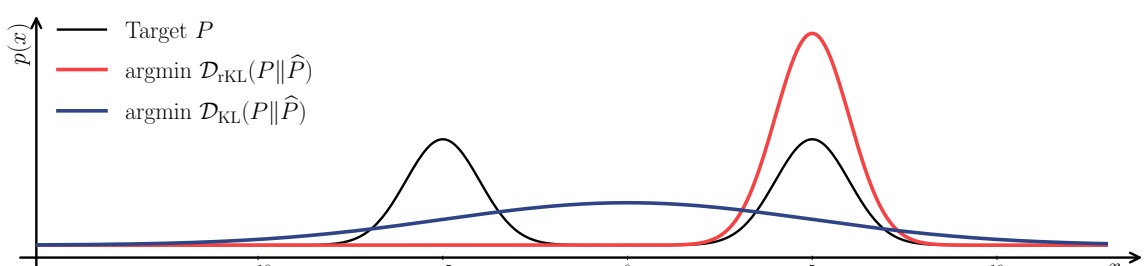
Examples of f-divergence minimization





Examples of f-divergence minimization





Can we optimize a specific trade-off between Precision and Recall?

- What is the relation between the Precision-Recall curve and f-divergences?
- Can we optimize a point on the Precision-Recall curve using f-divergences?

Can we optimize a specific trade-off between Precision and Recall?

- What is the relation between the Precision-Recall curve and f-divergences?
- Can we optimize a point on the Precision-Recall curve using f-divergences?

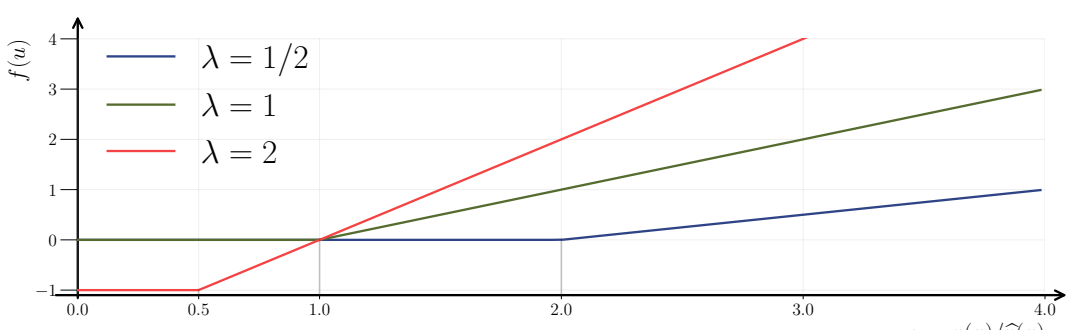
List of contributions:

- We show that the PR-Divergence is an f-divergence.
- We show that any f-divergence can be written as a weighted average PR-Divergences.
- We propose an algorithm to optimize the PR-Divergence.

Definition 3.3 (PR-Divergence generator function f_λ)

Given a trade-off parameter $\lambda \in [0, +\infty]$, we define the generator function $f_\lambda : [0, +\infty] \to]-\infty, +\infty]$ given by

$$f_\lambda(u) = \begin{cases} \max(\lambda u, 1) - \max(\lambda, 1) & \text{for } \lambda \in [0, +\infty[, \\ \mathbb{1}_{\{u=0\}} & \text{for } \lambda = +\infty. \end{cases} \tag{10}$$



Proposition 3.4 (PR-Divergence)

For any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$, then for any $\lambda \in [0, +\infty]$ the PR-Divergence defined as

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda \left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \right) \mathrm{d}\mu(\boldsymbol{x}) \tag{11}$$

belongs to the class of f-divergences.

Theorem 3.5 (PR-Curves as a function of $\mathcal{D}_{\lambda\text{-PR}}$)

Given $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, the PR-Curve ∂PRD is related to the PR-Divergence $\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})$ as follows.

$$\alpha_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}).$$
$$\beta_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(\widehat{P}\|P).$$

Theorem 3.5 (PR-Curves as a function of $\mathcal{D}_{\lambda\text{-PR}}$)

Given $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, the PR-Curve ∂PRD is related to the PR-Divergence $\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})$ as follows.

$$\alpha_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}).$$
$$\beta_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(\widehat{P}\|P).$$

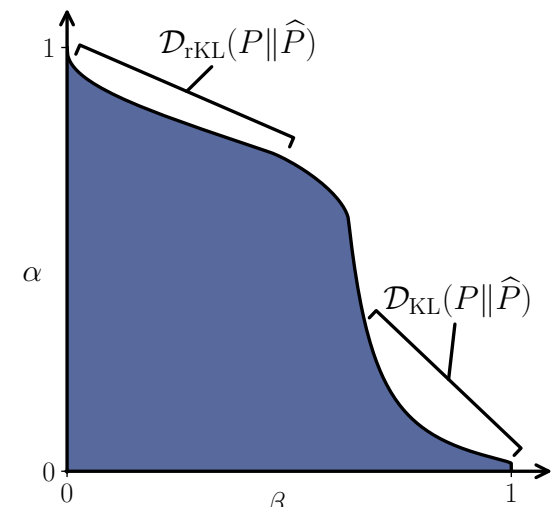
A direct consequence of Theorem 3.5:

$$\operatorname*{argmin}_{\widehat{P}\in\mathcal{P}(\mathcal{X})} \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) = \operatorname*{argmax}_{\widehat{P}\in\mathcal{P}(\mathcal{X})} \alpha_\lambda(P\|\widehat{P}).$$

Theorem 3.6 (f-divergences as a weighted average of PR-Divergences)

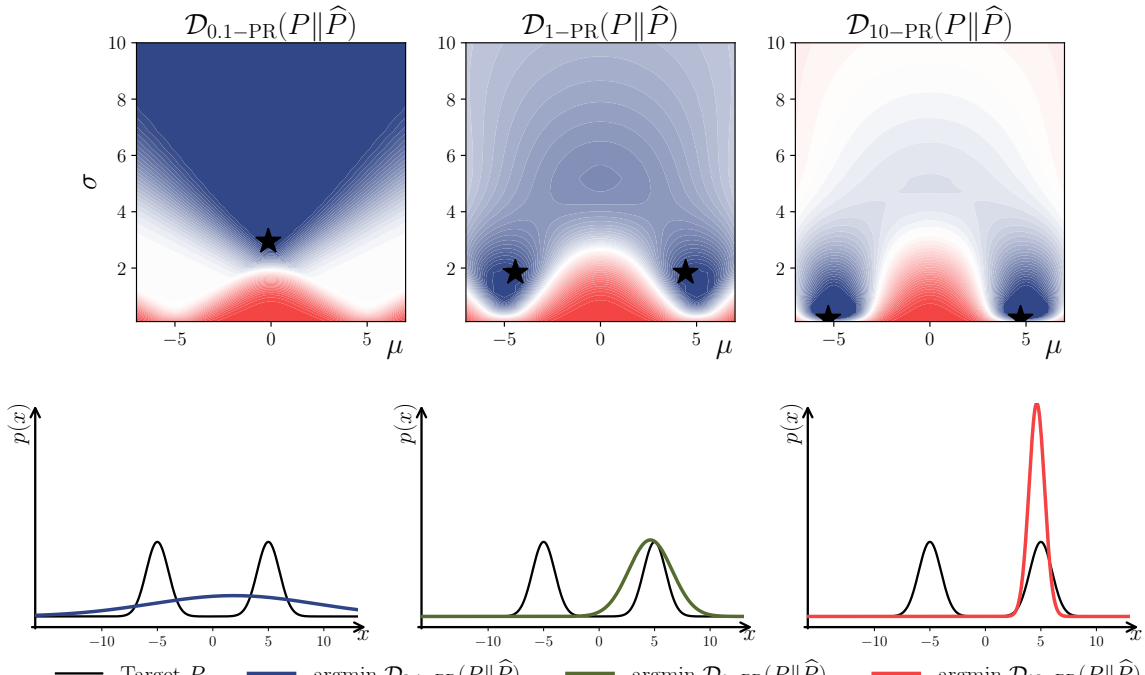
For any $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ supported on all $\mathcal{X}$ and any $\lambda \in [0, +\infty]$, then:

$$\mathcal{D}_f(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) \mathrm{d}\lambda,$$



Optimizing the PR-Divergence

Examples



Optimizing the PR-Divergence

Examples

Optimizing the PR-Divergence

In Practice

If we train a model to minimize the PR-Divergence, we can use the dual variational form:

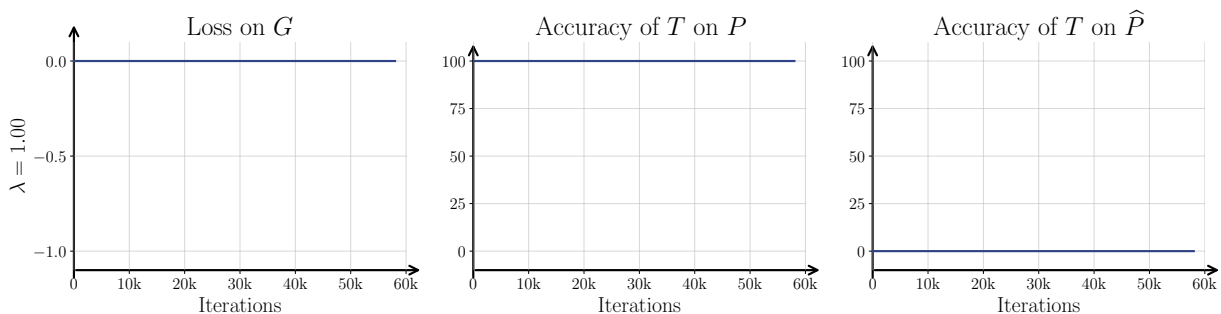
$$\min_G \max_T \; \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f_\lambda^*(T(\boldsymbol{x}))\right]. \tag{12}$$

Optimizing the PR-Divergence

In Practice

If we train a model to minimize the PR-Divergence, we can use the dual variational form:

$$\min_G \max_T \; \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*_\lambda(T(\boldsymbol{x}))\right]. \tag{12}$$



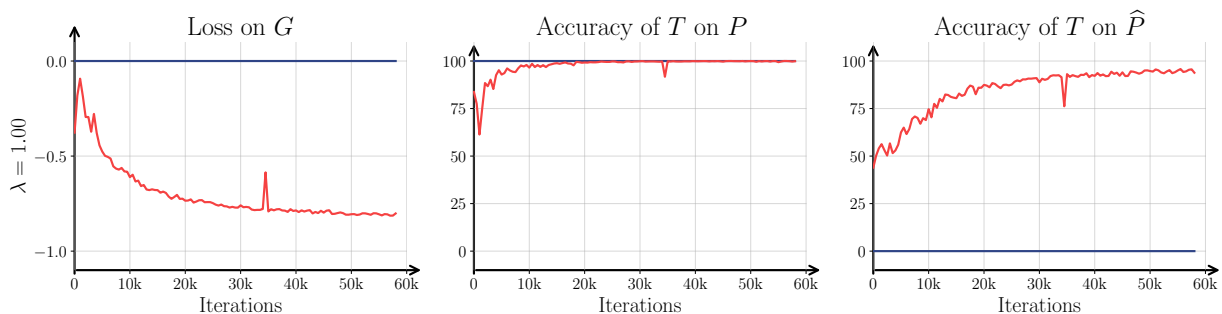
The naive approach *fails* to optimize the PR-Divergence.

Optimizing the PR-Divergence

In Practice

If we train a model to minimize the PR-Divergence, we can use the dual variational form:

$$\min_G \max_T \; \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*_\lambda(T(\boldsymbol{x}))\right]. \tag{12}$$



We propose *a new approach* to optimize the PR-Divergence.

Optimizing the PR-Divergence

Our approach

We choose an auxiliary function g to train T_g is trained to estimate the f-divergence $\mathcal{D}_g$:

$$\max_T \ \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}}\left[g^*(T(\boldsymbol{x}))\right] \tag{13}$$

Optimizing the PR-Divergence

Our approach

We choose an auxiliary function g to train T_g is trained to estimate the f-divergence $\mathcal{D}_g$:

$$\max_T \ \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}}\left[g^*(T(\boldsymbol{x}))\right] \tag{13}$$

At optimality, we have:

$$\nabla g^*(T_g^{\text{opt}}(\boldsymbol{x})) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}. \tag{14}$$

Optimizing the PR-Divergence

Our approach

We choose an auxiliary function g to train T_g is trained to estimate the f-divergence $\mathcal{D}_g$:

$$\max_T \ \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}}\left[g^*(T(\boldsymbol{x}))\right] \tag{13}$$

At optimality, we have:

$$\nabla g^*(T_g^{\text{opt}}(\boldsymbol{x})) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}. \tag{14}$$

Any f-divergence can be computed using the primal estimation as follows using T_g:

$$\mathcal{D}_f(P\|\widehat{P}) = \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right] = \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f\left(\nabla g^*(T_g^{\text{opt}}(\boldsymbol{x}))\right)\right]. \tag{15}$$

Optimizing the PR-Divergence

Our approach

Theorem 3.7 (Bound on the estimation of an f-divergence using an auxiliary g-divergence)

Let $f, g : \mathbb{R}^+ \to \mathbb{R}$ be such that g is μ-strongly convex, f is σ-Lipschitz, and $\mathcal{D}_f$, $\mathcal{D}_g$ be f-divergences. For any discriminator $T : \mathcal{X} \to \mathrm{dom}(g^*)$, let $r(\boldsymbol{x}) = \nabla g^*(T(\boldsymbol{x}))$. Then:

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}_{g,T}^{\mathrm{dual}} \leq \epsilon \implies \left|\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_{f,T}^{\mathrm{primal}}(P\|\widehat{P})\right| \leq \sigma\sqrt{\frac{2\epsilon}{\mu}}. \tag{16}$$

Theorem 3.7 (Bound on the estimation of an f-divergence using an auxiliary g-divergence)

Let $f, g : \mathbb{R}^+ \to \mathbb{R}$ be such that g is μ-strongly convex, f is σ-Lipschitz, and $\mathcal{D}_f$, $\mathcal{D}_g$ be f-divergences. For any discriminator $T : \mathcal{X} \to \mathrm{dom}(g^*)$, let $r(\boldsymbol{x}) = \nabla g^*(T(\boldsymbol{x}))$. Then:

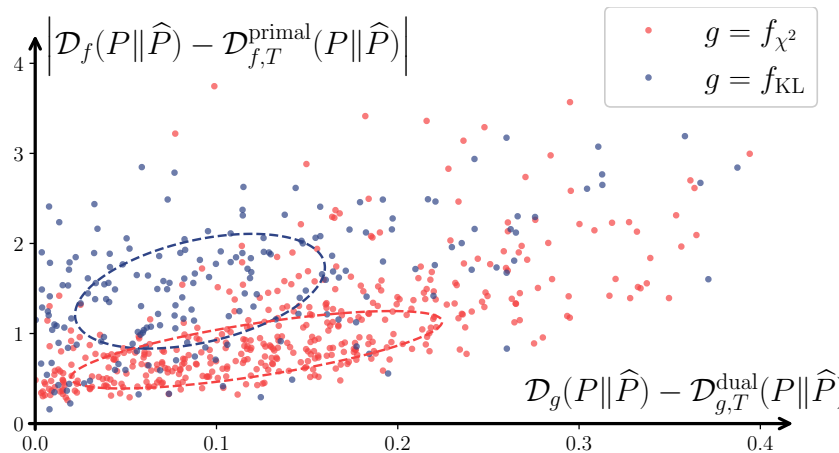
$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}_{g,T}^{\mathrm{dual}} \leq \epsilon \implies \left|\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}_{f,T}^{\mathrm{primal}}(P\|\widehat{P})\right| \leq \sigma\sqrt{\frac{2\epsilon}{\mu}}. \tag{16}$$

The smaller the error on T is,

the smaller the error on estimating $\mathcal{D}_f$ using T is.

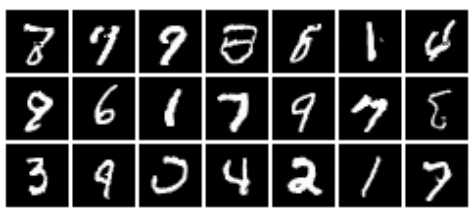
Convergence of the proposed approach

Examples

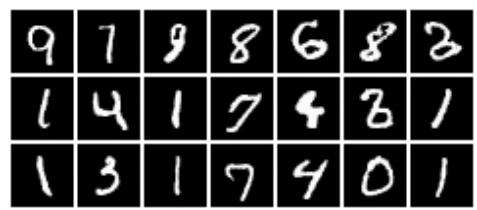


Optimizing the PR-Divergence with our approach

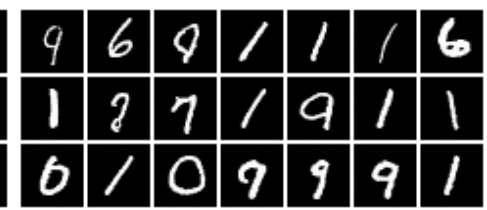
In Practice



(a) $\lambda = 0.1$

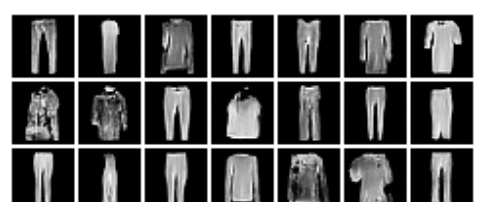


(b) $\lambda = 1$

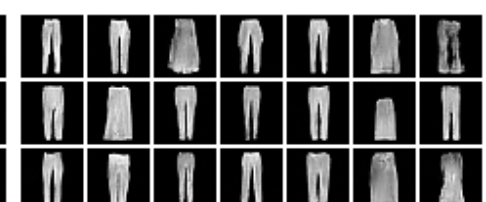


(c) $\lambda = 10$

(d) $\lambda = 0.1$



(e) $\lambda = 1$



(f) $\lambda = 10$

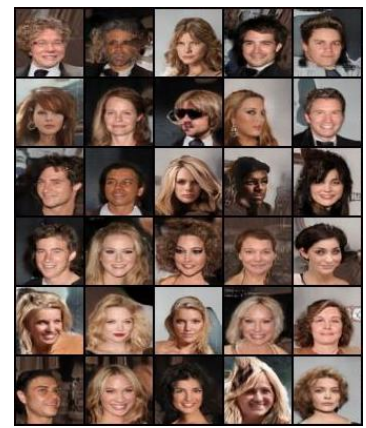
Optimizing the PR-Divergence with our approach

Training GANs

Model	CIFAR-10 32×32			CelebA 64×64		
	FID	P	R	FID	P	R
Baseline Big-GAN	13.37	86.51	65.66	9.16	78.41	**51.42**
$\lambda = 0.05$	13.29	81.10	70.63	-	-	-
$\lambda = 0.1$	**11.62**	81.78	**74.58**	-	-	-
$\lambda = 0.2$	13.36	84.85	65.13	8.79	83.37	44.07
$\lambda = 0.5$	14.50	83.27	68.23	**6.03**	77.60	**55.98**
$\lambda = 1.0$	14.03	83.04	69.35	13.07	81.70	36.85
$\lambda = 2.0$	16.94	84.93	59.79	14.23	82.98	32.87
$\lambda = 5.0$	32.54	83.39	56.94	22.45	**83.96**	25.81
$\lambda = 10.0$	39.69	84.11	39.29	-	-	-
$\lambda = 20.0$	67.03	**90.03**	21.81	-	-	-

When λ increases, $\begin{cases} \text{Precision} \uparrow \\ \text{Recall} \downarrow \end{cases}$

$\lambda = 0.1$



$\lambda = 10$

Optimizing the PR-Divergence with our approach

Fine-Tuning GANs

Model	ImageNet 128 × 128			FFHQ 256 × 256		
	FID	P	R	FID	P	R
Baseline BigGAN	9.83	28.04	41.21	41.41	65.57	**10.17**
Soft $\psi = 0.7$	11.39	23.04	31.13	56.43	76.59	4.87
Soft $\psi = 0.5$	15.49	20.20	19.83	82.05	84.48	1.58
Hard $\psi = 2.0$	**9.69**	25.83	39.89	43.32	68.84	8.66
Hard $\psi = 1.0$	12.12	21.86	35.42	56.19	76.44	4.76
Hard $\psi = 0.5$	15.21	21.13	29.55	71.32	80.99	4.84
$\lambda = 0.2$	9.92	26.69	42.04	35.66	78.70	9.45
$\lambda = 0.5$	10.82	26.83	**42.38**	**35.24**	78.41	9.66
$\lambda = 1.0$	20.42	29.72	28.21	35.91	78.95	8.32
$\lambda = 2.0$	20.21	30.27	30.49	36.33	81.10	8.69
$\lambda = 5.0$	20.76	**30.87**	28.38	38.16	**84.31**	8.52

When λ increases, $\begin{cases} \text{Precision} \uparrow \\ \text{Recall} \downarrow \end{cases}$ with better performances than truncation.

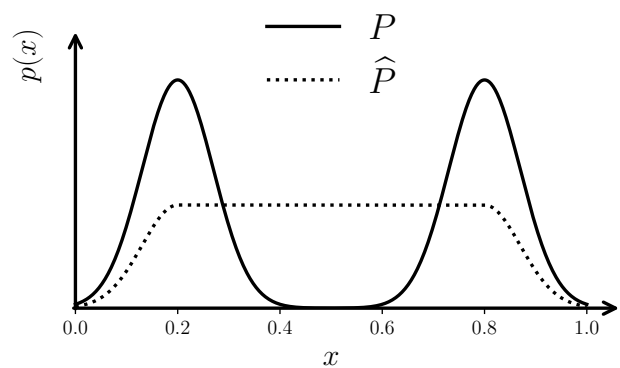
Improving:
How can we improve the quality and diversity of a pre-trained generative models?

Sampling from a generative model

General Setting

To sample a point from the learned distribution $\widehat{P}$:

- Sample $z \sim Q$.
- Compute $x = G(z)$.



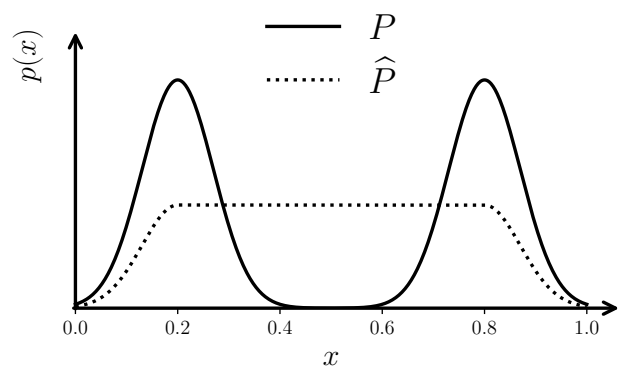
Sampling from a generative model

General Setting

To sample a point from the learned distribution $\widehat{P}$:

- Sample $z \sim Q$.
- Compute $x = G(z)$.

$$P \neq \widehat{P}$$



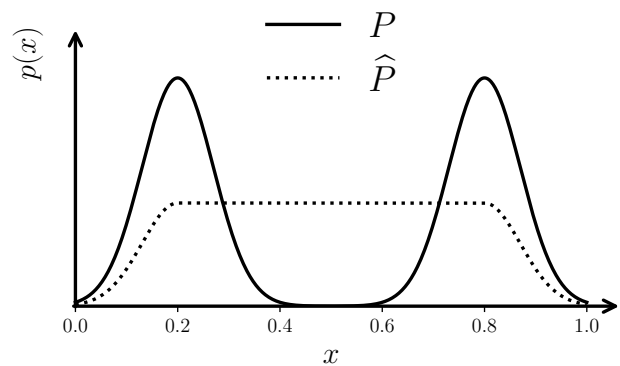
Sampling from a generative model

General Setting

To sample a point from the learned distribution $\widehat{P}$:

- Sample $z \sim Q$.
- Compute $x = G(z)$.

We have an estimation of $\frac{p(x)}{\widehat{p}(x)}$.



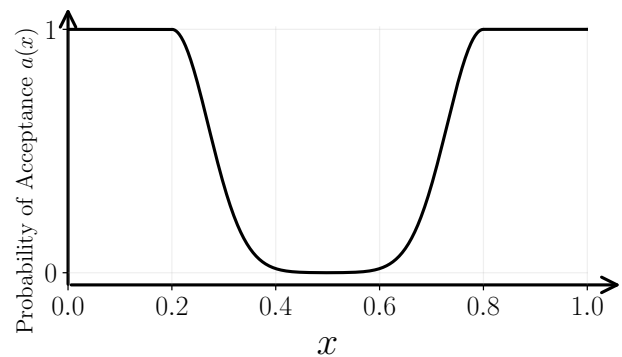
Sampling from a generative model

Rejection Sampling

To sample a point from the refined distribution $\widetilde{P}$:

- Sample $\boldsymbol{z} \sim Q$.
- Compute $\boldsymbol{x} = G(\boldsymbol{z})$.
- Accept $\boldsymbol{x}$ with probability $a(\boldsymbol{x})$.

Using $\frac{p(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}$ in $a(\boldsymbol{x})$ allows sampling from P.



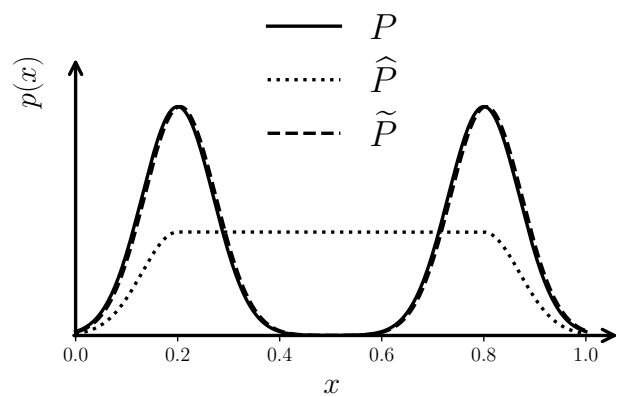
Sampling from a generative model

Rejection Sampling

To sample a point from the refined distribution $\widetilde{P}$:

- Sample $\boldsymbol{z} \sim Q$.
- Compute $\boldsymbol{x} = G(\boldsymbol{z})$.
- Accept $\boldsymbol{x}$ with probability $a(\boldsymbol{x})$.

It defines a new distribution $\widetilde{P}$.



Sampling from a generative model

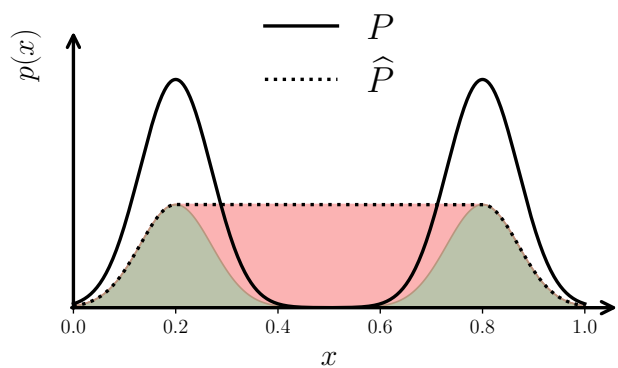
Rejection Sampling

To sample a point from the refined distribution $\widetilde{P}$:

- Sample $z \sim Q$.
- Compute $x = G(z)$.
- Accept x with probability $a(x)$.

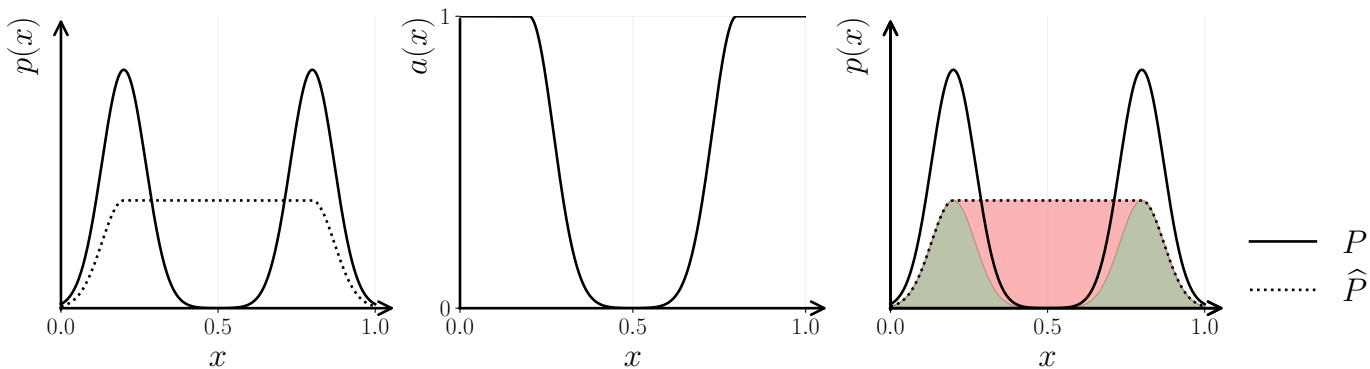
The acceptance rate is :

$$\mathbb{E}_{\widehat{P}}\left[a(x)\right].$$



Sampling from a generative model

Rejection Sampling in high dimension



Acceptance rate: 45.20%

Definition 4.1 (Discriminator Rejection Sampling (DRS) - Azadi et al. [1])

Let $\gamma \in \mathbb{R}$, the acceptance probability is:

$$a_{\text{DRS}}(\boldsymbol{x}) = \frac{r(\boldsymbol{x})}{r(\boldsymbol{x})\,(1 - e^{\gamma}) + Me^{\gamma}}.$$

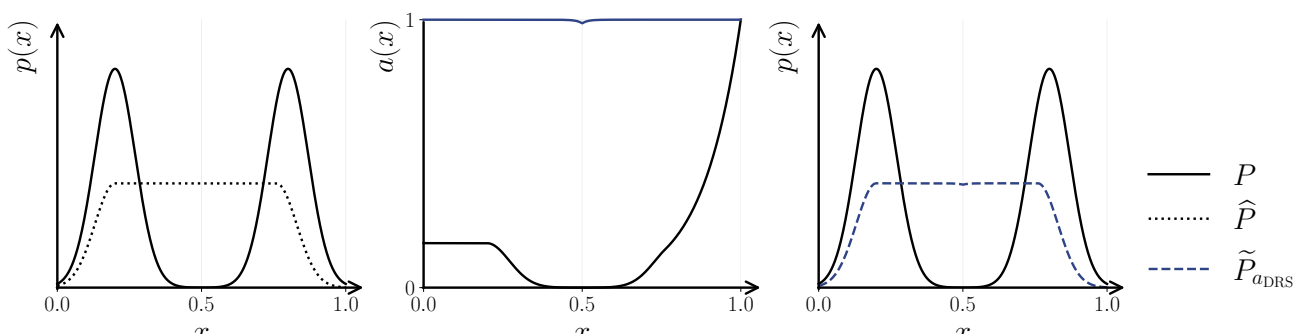
If $\gamma < 0$, then the acceptance rate increases.

Definition 4.1 (Discriminator Rejection Sampling (DRS) - Azadi et al. [1])

Let $\gamma \in \mathbb{R}$, the acceptance probability is:

$$a_{\text{DRS}}(\boldsymbol{x}) = \frac{r(\boldsymbol{x})}{r(\boldsymbol{x})\,(1 - e^{\gamma}) + Me^{\gamma}}.$$

If $\gamma < 0$, then the acceptance rate increases.



Acceptance rate: 100.00%

How can we improve the quality and diversity of a pre-trained generative models?

- How can we apply Rejection Sampling with a limited budget?
- How does it improve Precision and Recall?

How can we improve the quality and diversity of a pre-trained generative models?

- How can we apply Rejection Sampling with a limited budget?
- How does it improve Precision and Recall?

List of contributions:

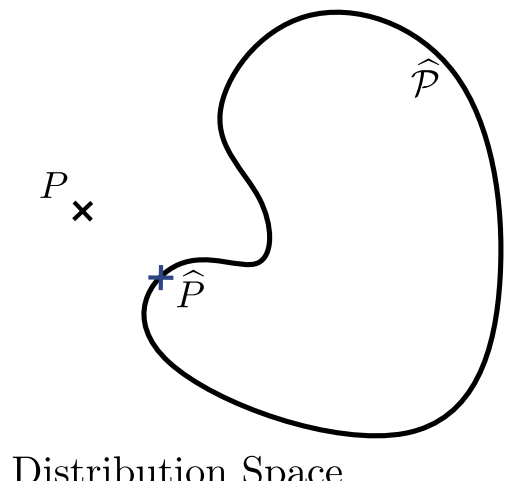
- We introduce the Optimal Budgeted Rejection Sampling (OBRS).
- We show how OBRS improves the Precision the learned distribution.
- We show that training a generative model with OBRS improves the Recall.

Optimal Budgeted Rejection Sampling

Definition

Traditionally, the goal is:

$$\min_{G} \quad \mathcal{D}_f(P \| \widehat{P}_G)$$

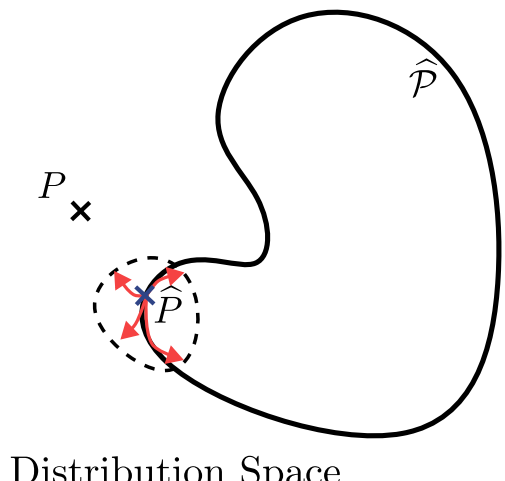


Optimal Budgeted Rejection Sampling

Definition

With a given $\widehat{P}_G$, our goal is:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(P \| \widetilde{P}_a) \\ \text{s.t.} \quad & \text{the acceptance rate is greater than } 1/K. \end{aligned}$$

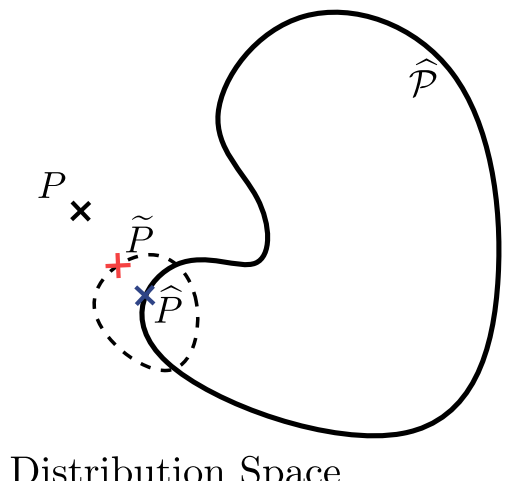


Optimal Budgeted Rejection Sampling

Definition

With a given $\widehat{P}_G$, our goal is:

$$\min_{a} \quad \mathcal{D}_f(P \| \widetilde{P}_a) \qquad \text{s.t.} \begin{cases} \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] \geq 1/K, \\ \forall \boldsymbol{x} \in \mathcal{X},\ 0 \leq a(\boldsymbol{x}) \leq 1. \end{cases} \tag{17}$$



Theorem 4.2 (Optimal Acceptance Function)

For a sampling budget $K \geq 1$ and finite $\mathcal{X}$, the solution is,

$$a_{\text{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right), \tag{18}$$

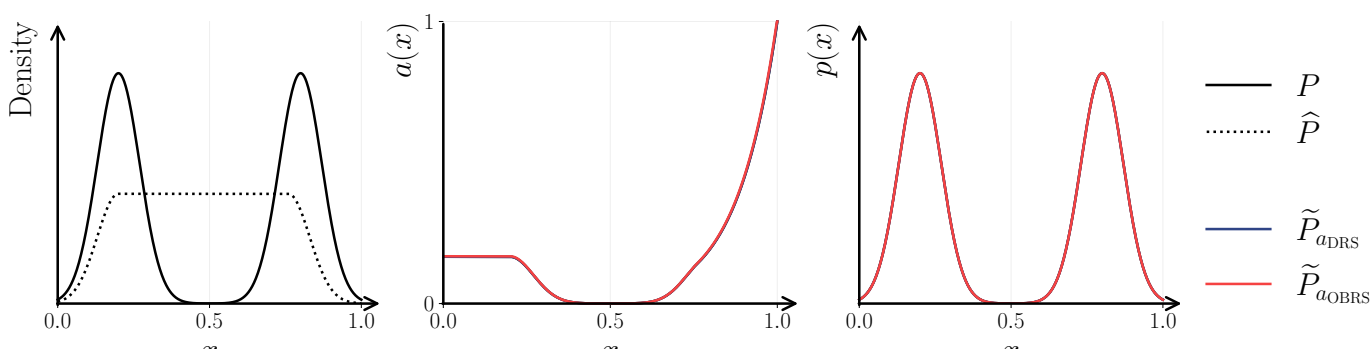
where $c_K \geq 1$ is such that $\mathbb{E}_{\boldsymbol{x}\sim\widehat{p}}[a_{\text{OBRS}}(\boldsymbol{x})] = 1/K$.

Theorem 4.2 (Optimal Acceptance Function)

For a sampling budget $K \geq 1$ and finite $\mathcal{X}$, the solution is,

$$a_{\text{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right), \tag{18}$$

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



Acceptance rate: 7.91%

Theorem 4.2 (Optimal Acceptance Function)

For a sampling budget $K \geq 1$ and finite $\mathcal{X}$, the solution is,

$$a_{\text{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right) \tag{18}$$

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

Does not depend on f.

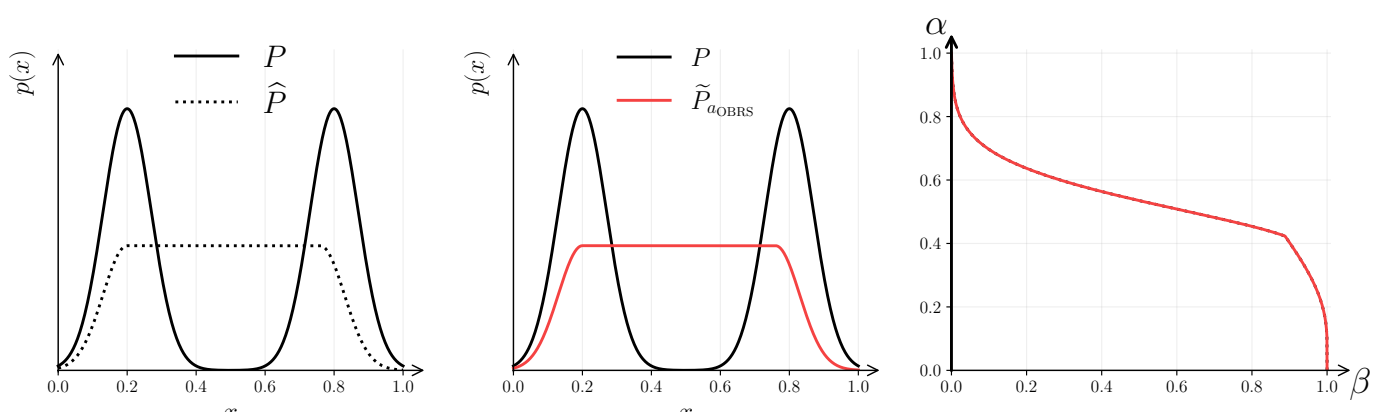
Same acceptance function to improve Precision and Recall.

Proposition 4.3 (Precision and Recall Improvement)

Let $K \leq M$ be the budget for the OBRS. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.

Proposition 4.3 (Precision and Recall Improvement)

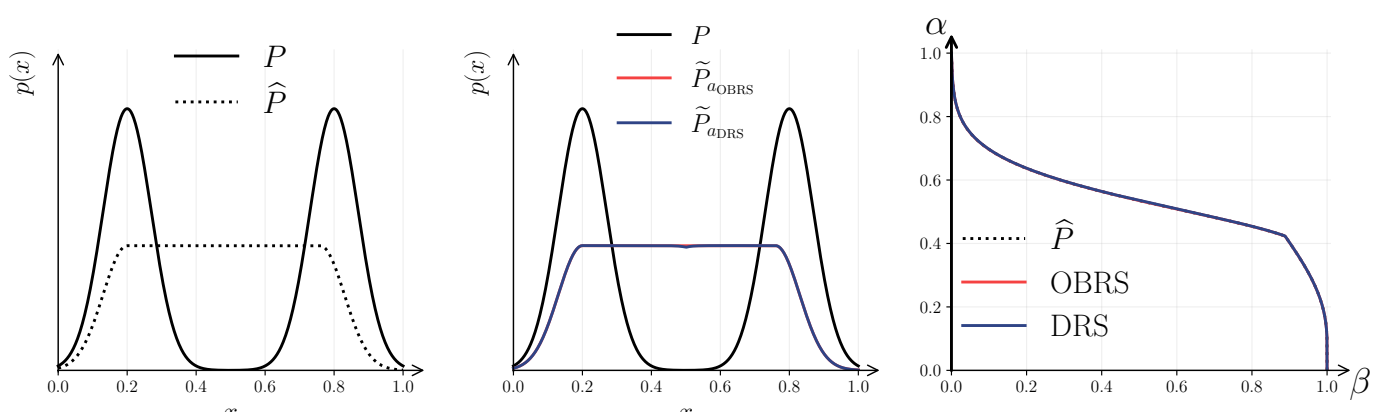
Let $K \leq M$ be the budget for the OBRS. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.



Acceptance rate: 100.00%

Proposition 4.3 (Precision and Recall Improvement)

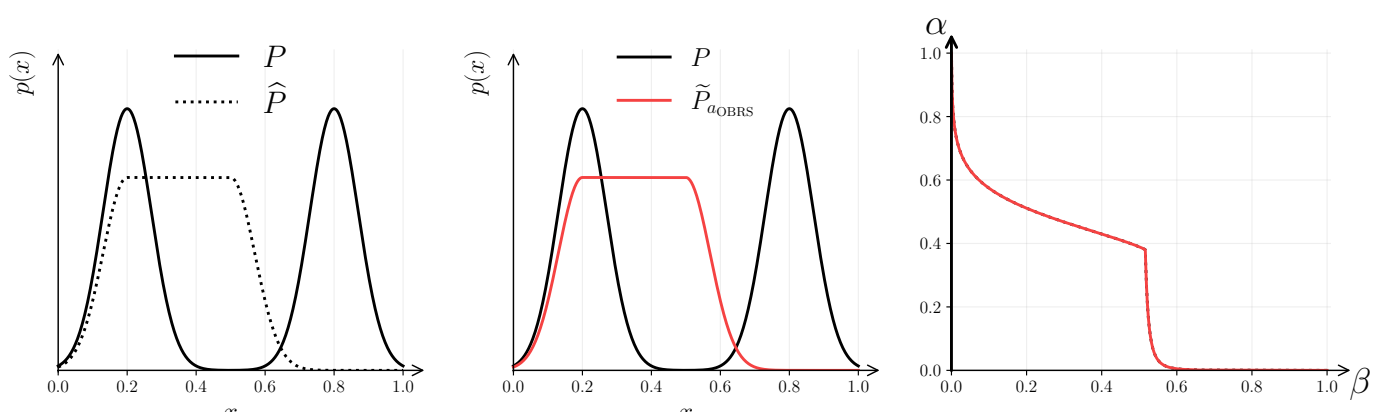
Let $K \leq M$ be the budget for the OBRS. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.



Acceptance rate: 100.00%

Proposition 4.3 (Precision and Recall Improvement)

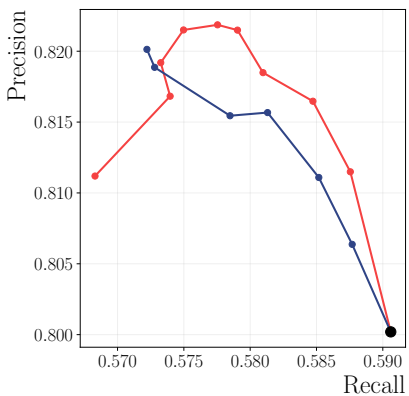
Let $K \leq M$ be the budget for the OBRS. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.

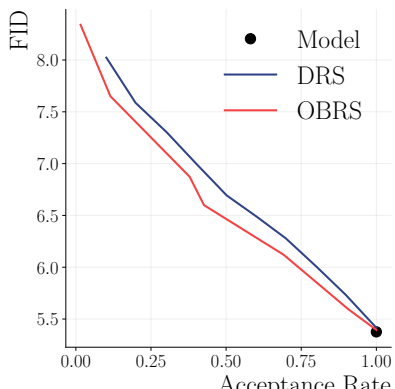


Acceptance rate: 100.00%

Improving Precision and Recall

In Practice





GAN on CelebA

1/K	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

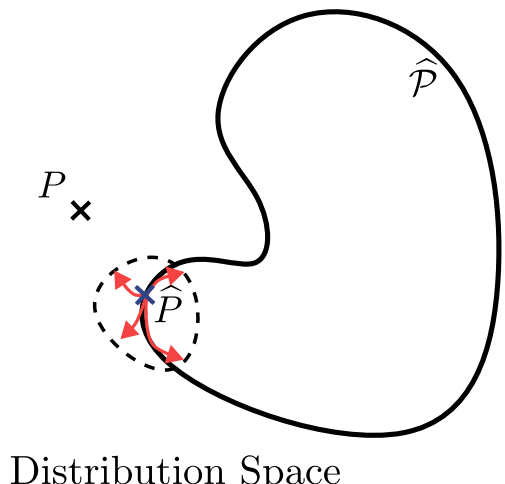
Diffusion Model on CIFAR-10

Training generative models with OBRS

Objective

With a given $\widehat{P}_G$, our goal is:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(P \| \widetilde{P}_a) \\ \text{s.t.} \quad & \text{the acceptance rate is greater than } 1/K. \end{aligned}$$



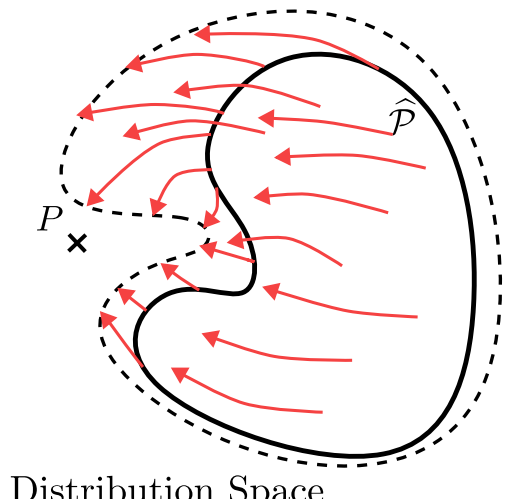
Training generative models with OBRS

Objective

We can train G to optimize:

$$\min_G \min_a \quad \mathcal{D}_f(P \| \widetilde{P}_{a,G})$$

$$\text{s.t.} \quad \text{the acceptance rate is greater than } 1/K.$$



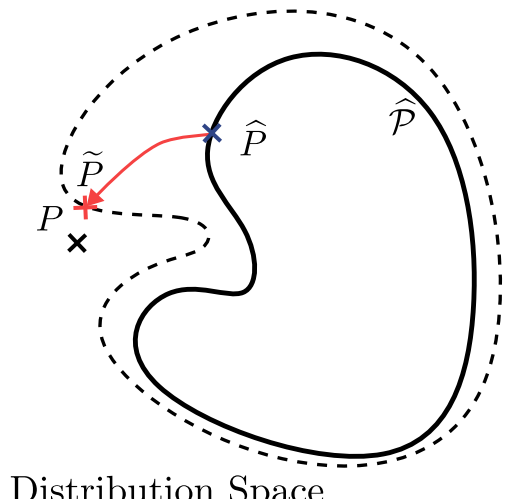
Training generative models with OBRS

Objective

We can train G to optimize:

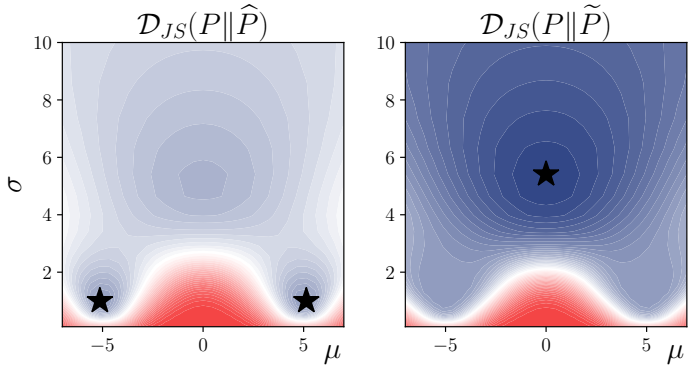
$$\min_{G} \min_{a} \quad \mathcal{D}_f(P \| \widetilde{P}_{a,G})$$

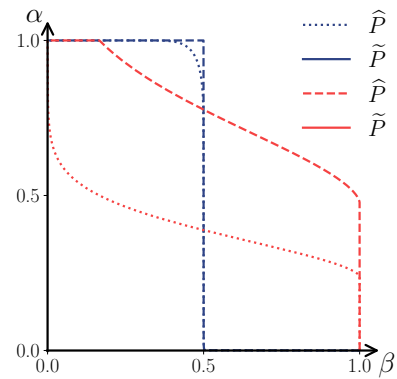
$$\text{s.t.} \quad \text{the acceptance rate is greater than } 1/K.$$

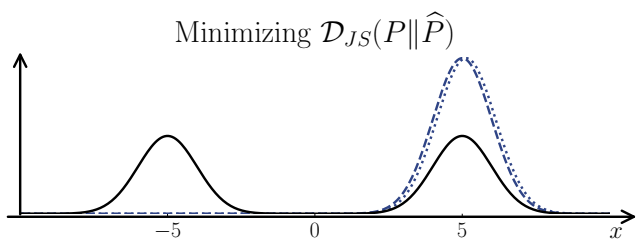


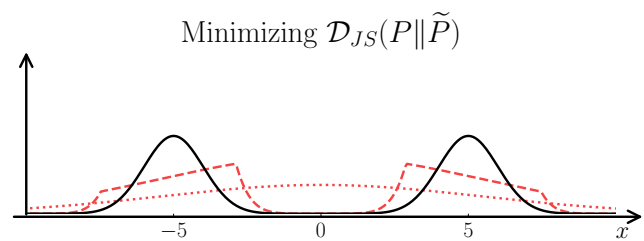
Training with OBRS

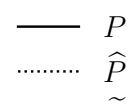
Example











Training with OBRS

In Practice

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

Training with OBRS increases the Recall.

Conlusion

and perspectives

- Quality and diversity are two important aspects of generative models.
- Generally at odds with each other, but can be balanced.
- The trade-off must be optimized.

Conlusion

and perspectives

- Quality and diversity are two important aspects of generative models.
- Generally at odds with each other, but can be balanced.
- The trade-off must be optimized.

Optimizing the trade-off

- Connecting PR-Curves and f-divergences
  - Focusing on the AUC
  - Building a symmetric PR-Divergence
- Minimizing the PR-Divergence
  - Apply the method to Diffusion Models
  - Investigate other training strategies

Conlusion

and perspectives

- Quality and diversity are two important aspects of generative models.
- Generally at odds with each other, but can be balanced.
- The trade-off must be optimized.

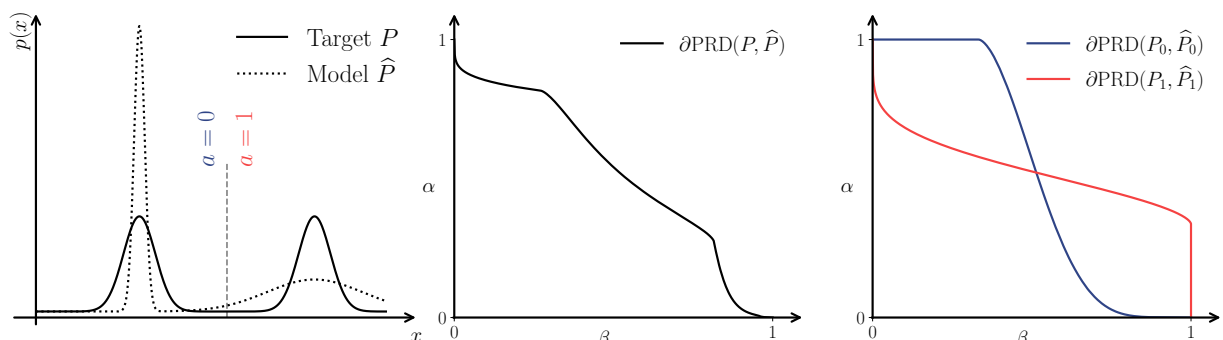
Optimizing the trade-off

- Connecting PR-Curves and f-divergences
  - Focusing on the AUC
  - Building a symmetric PR-Divergence
- Minimizing the PR-Divergence
  - Apply the method to Diffusion Models
  - Investigate other training strategies

Improving the quality and diversity

- Optimal Budgeted Rejection Sampling
  - Investigate the density ratio estimation error
  - Apply the method during denoising in Diffusion Models
- Training to improve diversity
  - Build formal proofs

There are no metrics to distinguish between *fair* and *unfair* generations in terms of Precision and Recall.



Questions

References I

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

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

[3] 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. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. arXiv:2402.10693 [cs].

[4] 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.

[5] Grover, A., Dhar, M., and Ermon, S. (2018). Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models. arXiv:1705.08868 [cs, stat].

[6] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. arXiv:1912.04958 [cs, eess, stat].

[7] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative Flow with Invertible 1x1 Convolutions. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.*, volume 31.

References II

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] Siry, R., Webster, R., Simon, L., and Rabin, J. (2023). On the Theoretical Equivalence of Several Trade-Off Curves Assessing Statistical Proximity. *Journal of Machine Learning Research*, 24.

[14] Yann LeCun, Corinna Cortes, and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs*, 2.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.