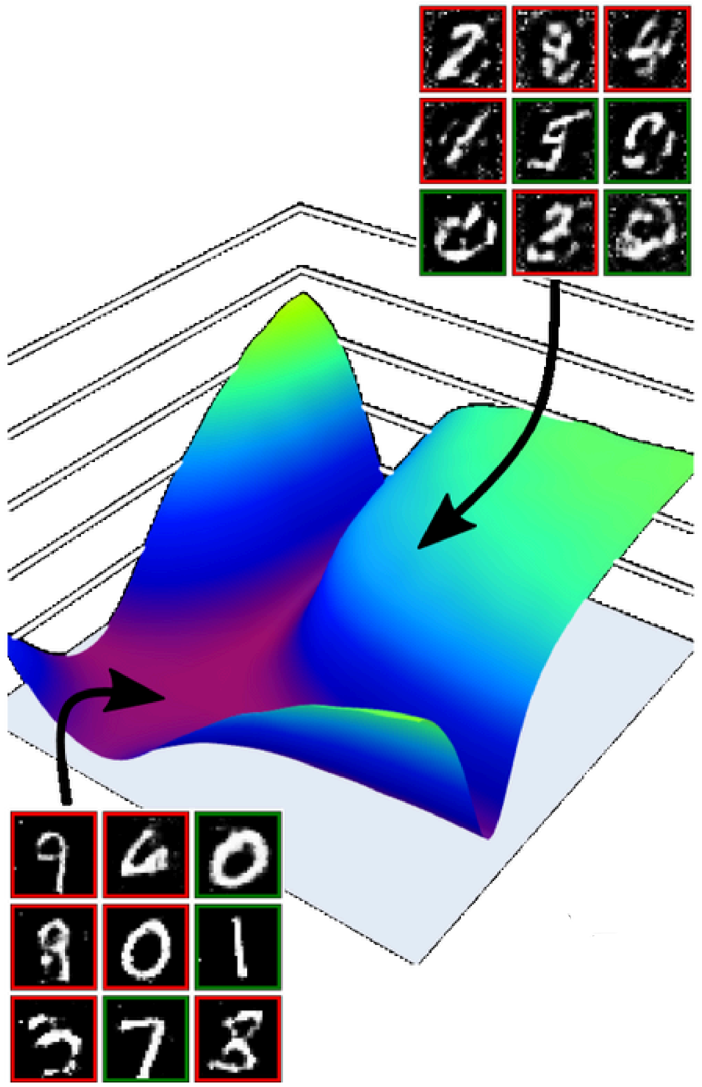
Cameron Mouangue

Optimal Budget Rejection Sample



Deep Learning Project

Table of Contents

1. Theoretical Grounds for OBRS

1. Theoretical Grounds for OBRS

Definition: OBRS is a method designed to improve GAN outputs by minimizing the *f-divergence* $D_f(P \parallel \tilde{P}_a)$ between:

- P: True data distribution.
- $\tilde{P}_a$: Post-rejection distribution.

Objective:

$$\min_a D_f(P \parallel \tilde{P}_a), \quad \text{subject to: } \mathbb{E}_{P_G}[a(x)] \geq \frac{1}{K},\ 0 \leq a(x) \leq 1.$$

Optimal Acceptance Function:

$$a^*(x) = \min\left(\frac{p(x)}{p_G(x)} \cdot \frac{c_K}{M}, 1\right),$$

where $M = \sup_x \frac{p(x)}{p_G(x)}$ and c_K ensures $\mathbb{E}_{P_G}[a^*(x)] = \frac{1}{K}$.

Key Advantages: Improves precision and recall of samples by selectively refining generator outputs while maintaining computational efficiency.

Precision and Recall in Generative Models:

Precision (α): Measures how much of the generated distribution P_G aligns with the true data distribution P. High precision implies fewer low-quality samples.

Recall (β): Measures how much of the true data distribution P is covered by the generated distribution P_G. High recall implies greater diversity in generated samples.

How OBRS Improves These Metrics:

$$a(x) = \min\left(\frac{p(x)}{p_G(x)} \cdot \frac{c_K}{M}, 1\right),$$

where:

High-Quality Samples (Precision): Samples with a higher ratio $\frac{p(x)}{p_G(x)}$ are more likely to be accepted, reducing the chance of generating poor-quality samples.

Diversity (Recall): By optimizing c_K for the rejection budget K, OBRS strikes a balance between filtering low-quality samples and maintaining diversity.

No Precision-Recall Tradeoff :

$$\alpha' = \min\{1, K \cdot \alpha\}, \quad \beta' = \beta$$

For a fixed recall (β), precision increases proportionally to K until capped at 1, leading to a "vertical scaling" of the PR-curve.

...

Two ways of using OBRS :

- **1. Apply OBRS to a Pretrained Model:**

  Refines the output of an existing generator P_G to produce $\tilde{P}_a$.

- **2. Train with OBRS (Tw/OBRS):**

  Embeds the rejection sampling process into training by directly minimizing $D_f(P \parallel \tilde{P}_a)$.

  The generator learns to output samples closer to P during training, considering the post-rejection refinement.

  Advantages:

  Produces a generator G inherently optimized for precision and recall, reducing dependency on rejection after training.

  Leads to a flatter loss landscape, avoiding local minima during optimization.

Loss Function for Training with OBRS:

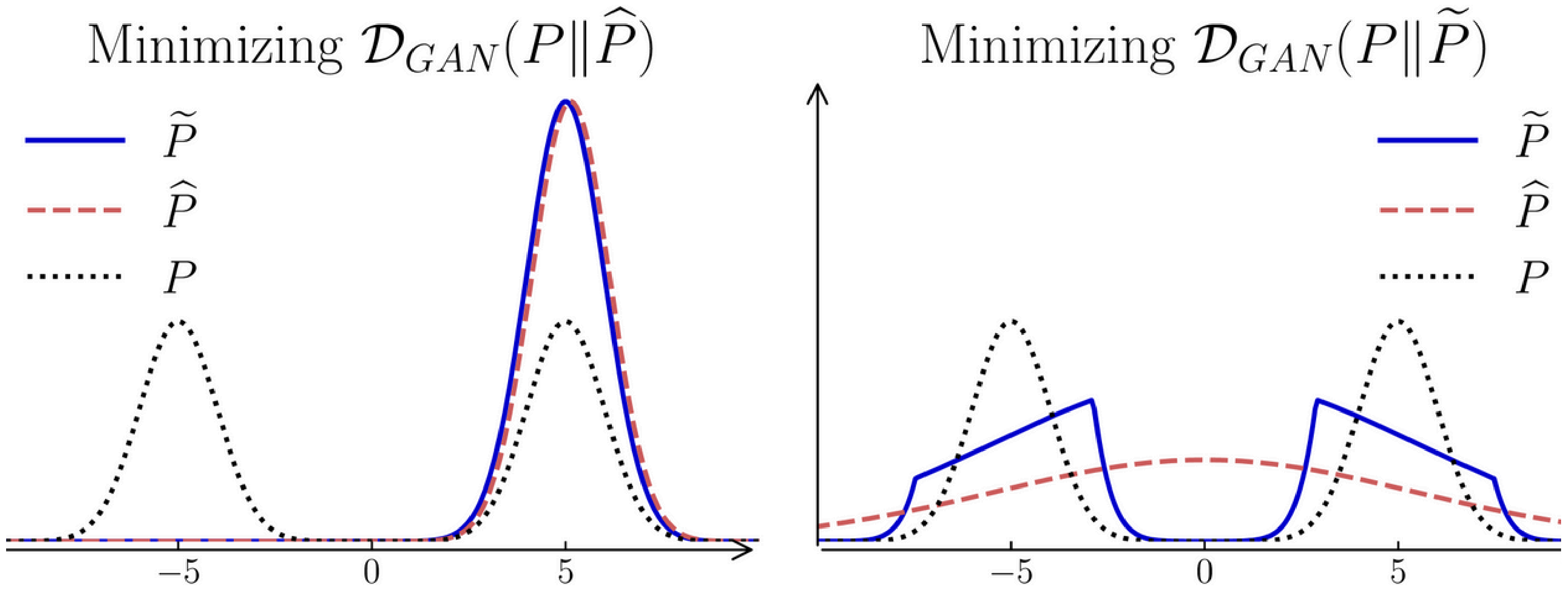
$$\mathbb{E}_{x \sim P_G}\left[Ka^*(x) f\left(\frac{\nabla f^*(T(x))}{Ka^*(x)}\right)\right],$$

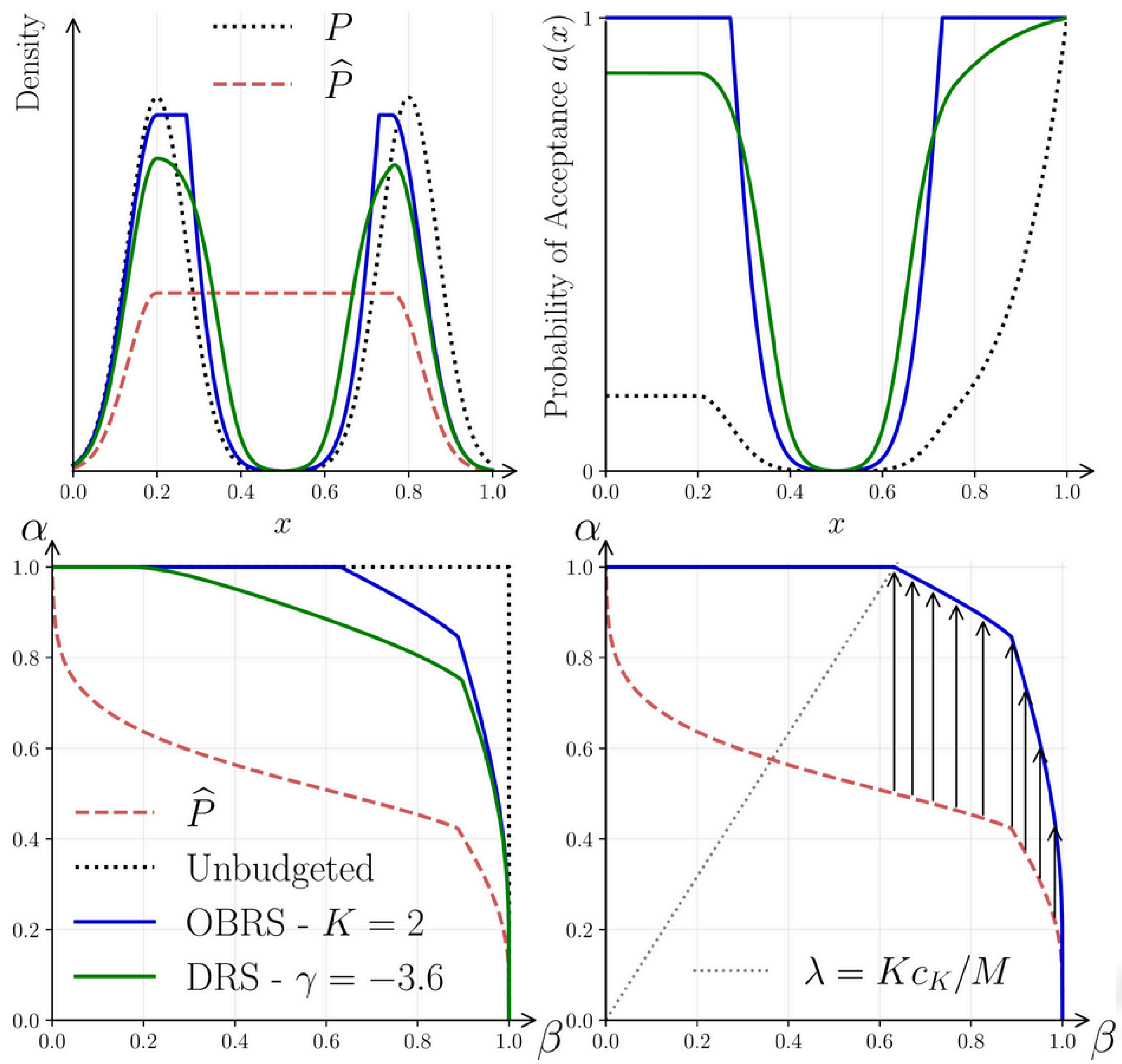
where:

$a^*(x) = \min\left(\frac{p(x)}{p_G(x)} \cdot \frac{c_K}{M}, 1\right)$.

$\nabla f^*(T(x))$ is the likelihood ratio estimated by the discriminator T.

Visualising OBRS





2. Adapting the method

2.Adapting the method

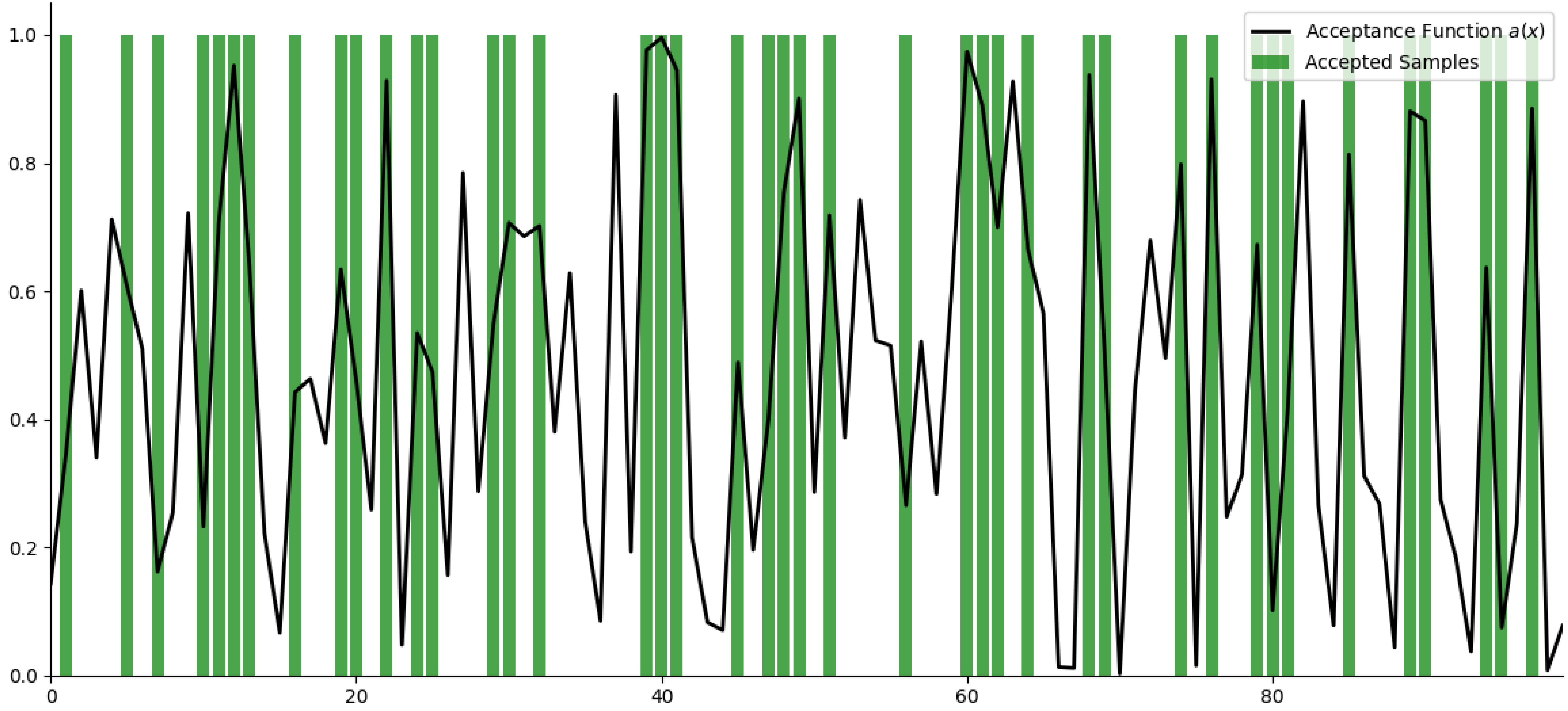
...

Version	Description	Key Modifications
Version 0	No OBRS Applied	Baseline GAN. PR taken directly from the generator without any rejection sampling.
Version 1	OBRS Applied to Generation Only (K=2)	OBRS applied only during the generation phase to refine generated samples.
Version 2	Full OBRS Applied (K=2)	OBRS applied to both generation and training, optimizing the generator during training for alignment with the target distribution.
Version 3	OBRS Applied to Training Only (K=2)	OBRS applied solely during training to refine the generator's output while leaving the generation phase untouched.
Version 4	Full OBRS Applied (K>2)	Full OBRS applied with a higher rejection budget $K > 2$ to test its impact on PR.

Additional Notes for OBRS Training:

- **Discriminator's Learning Rate Adjustment:** To effectively train with OBRS, I had to lower the discriminator's learning rate by a factor of 10.
- This adjustment was necessary to maintain stability and prevent the discriminator from becoming too aggressive, allowing the generator to benefit from the refined feedback during training.

Implementation of a(x)



Binary Acceptance (Bernoulli Sampling) with a(x)

Challenges and Solutions :

Challenges Encountered

- OBRS is a New Method
- Numerical Instabilities
- Computational Expense of c_K

Solutions Implemented

- Careful Reading and Following the Theory
- Clamping Values and Using Gradient Clipping
- Using the Betas Parameter to Smooth Training Progress
- Reducing Computational Load of c_K

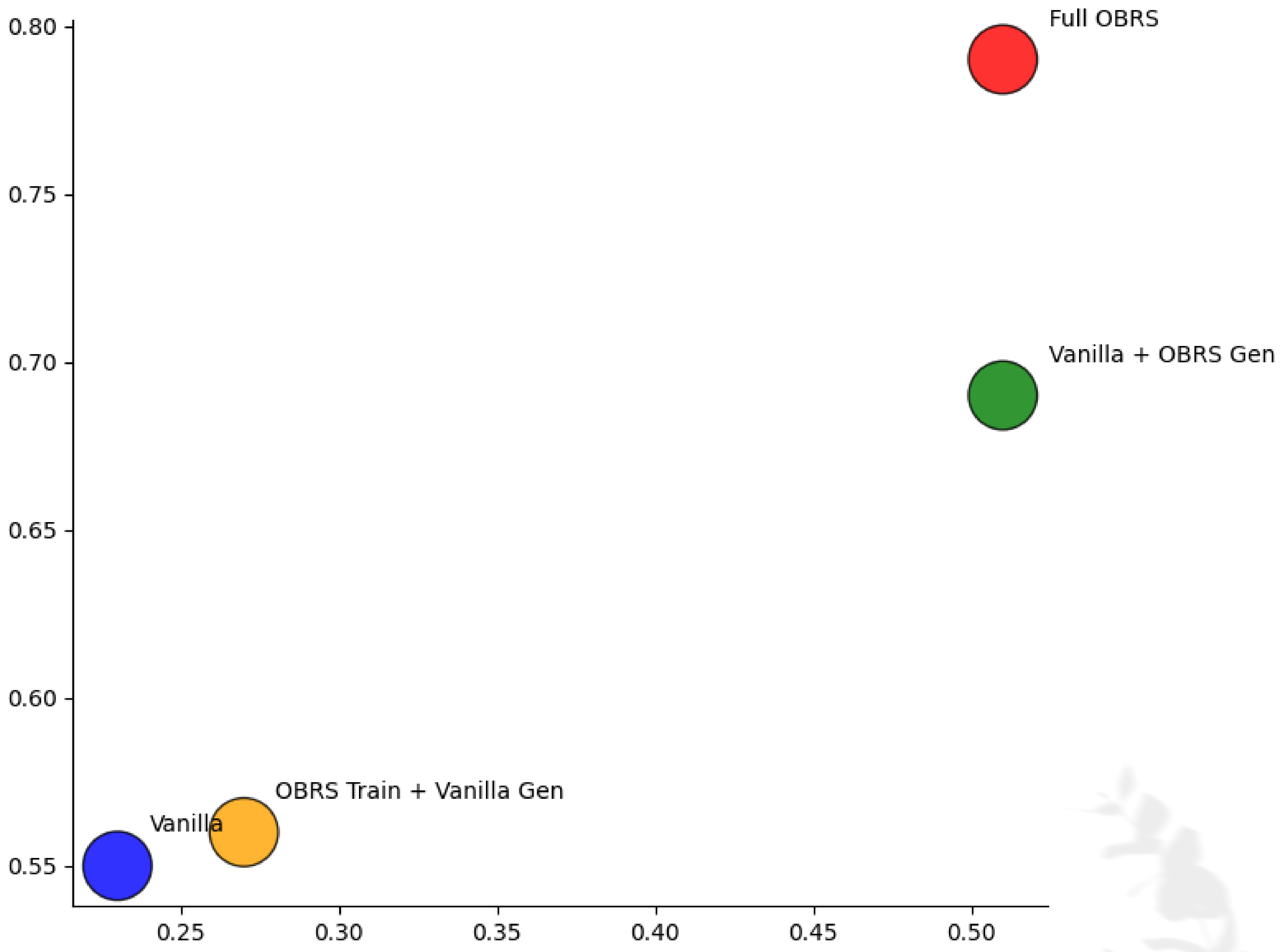
3. Results

...

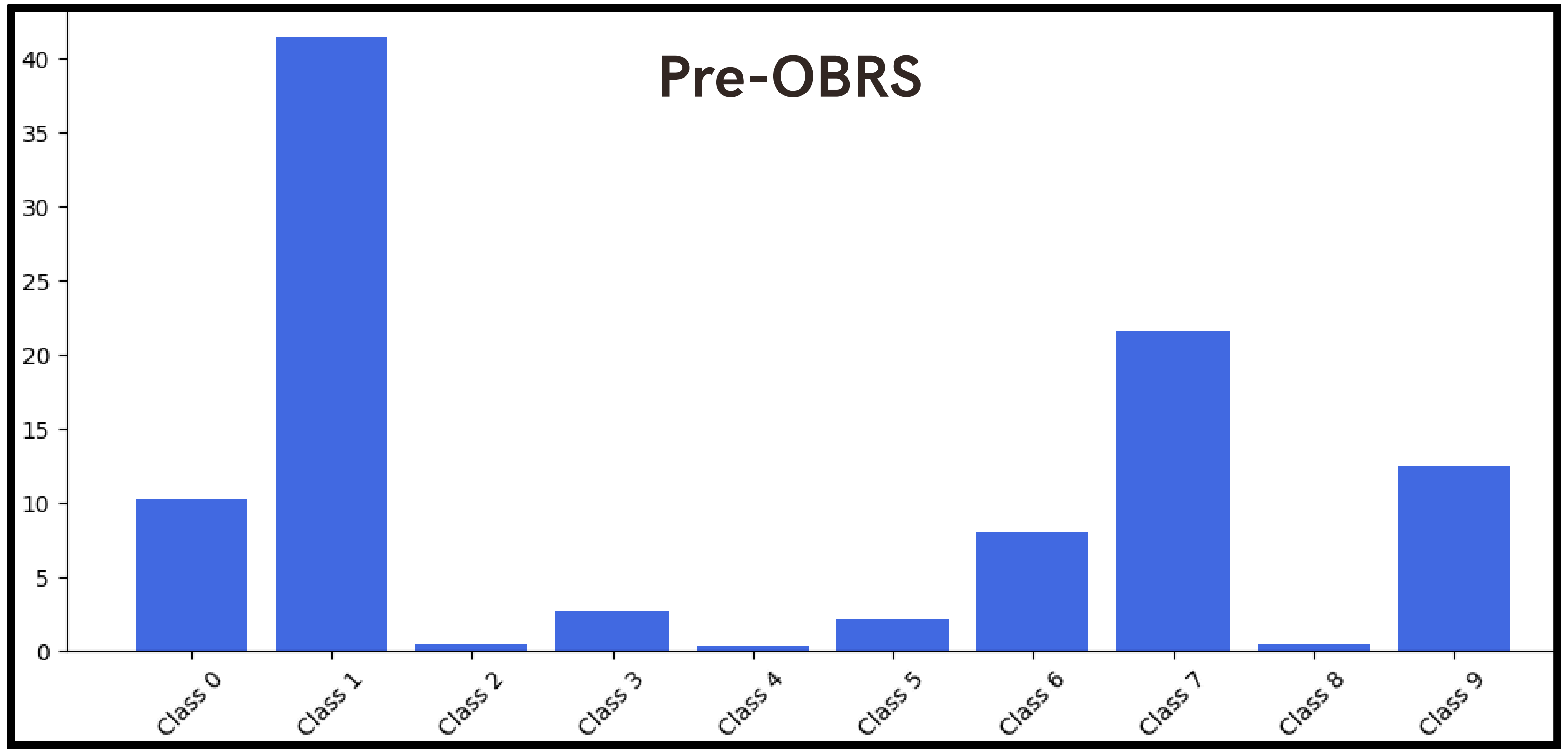
3. Results

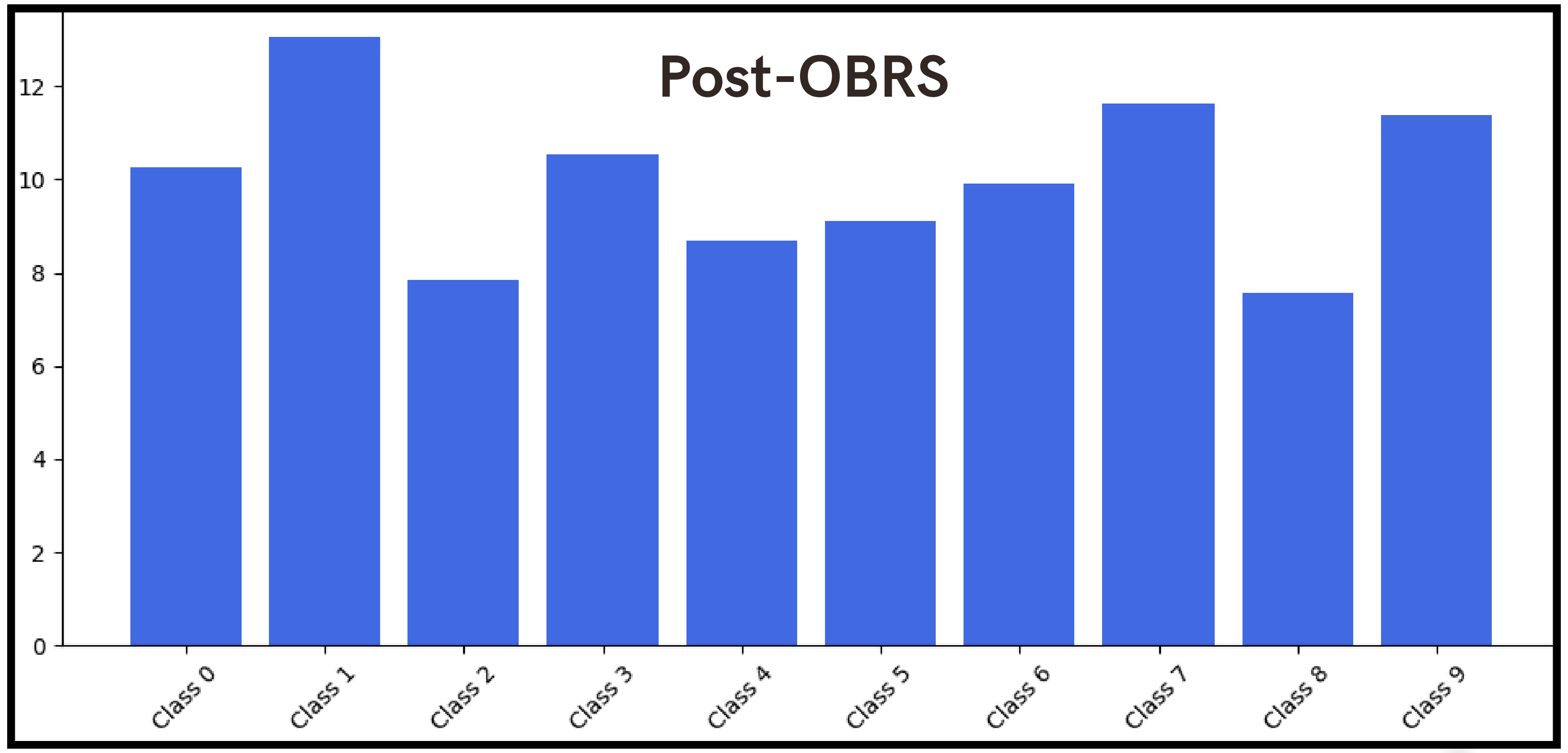
Performance Comparison of Versions:

- **Vanilla GAN (Baseline):**
  FID: 29
  Precision: 0.55
  Recall: 0.23
- **OBRS Training + Vanilla Generation (K=5):**
  FID: 16
  Precision: 0.56
  Recall: 0.27
- **Vanilla Training + OBRS Filtering (K=2):**
  FID: 45
  Precision: 0.69
  Recall: 0.51
- **Full OBRS (Training + Generation, K=2):**
  FID: 31
  Precision: 0.79
  Recall: 0.51



Improvement in mode diversity



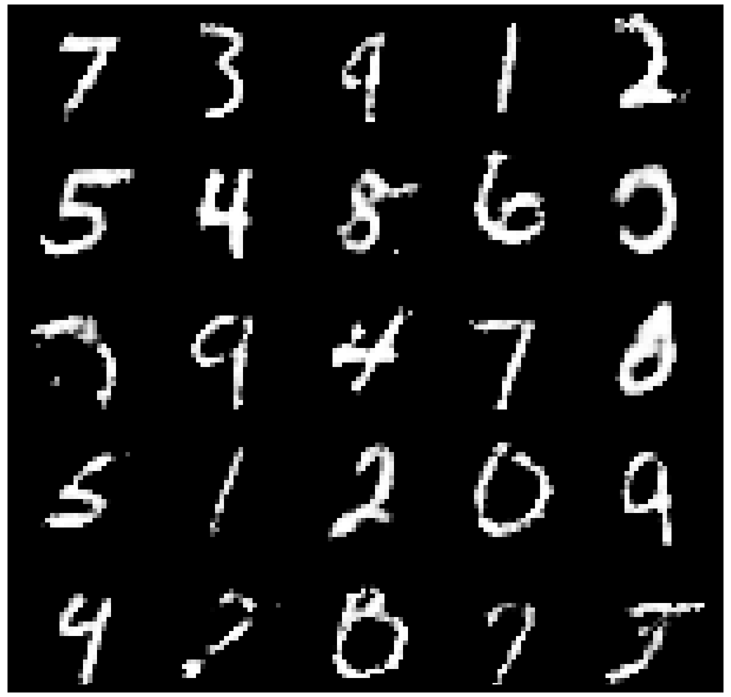


Near Max Potential

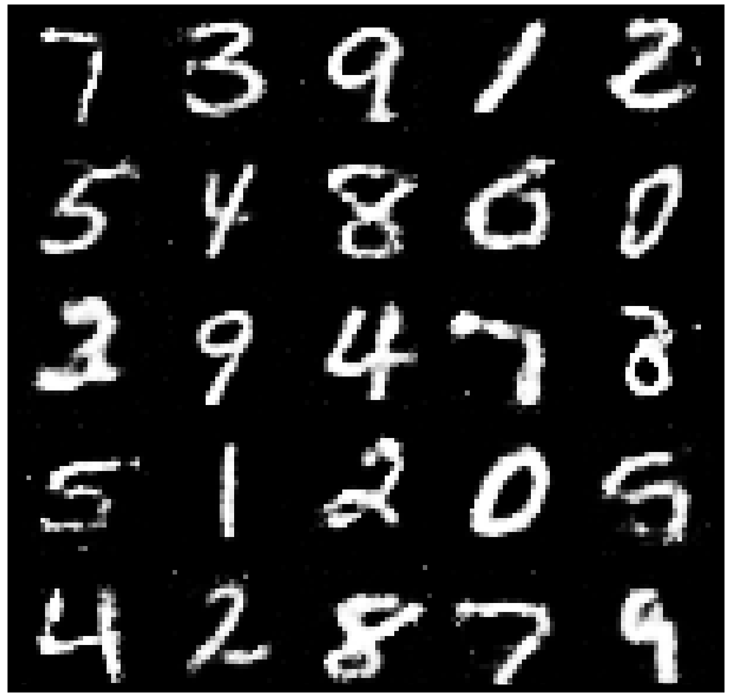
Key Results for $K = 5$ (Full OBRS, 200 Epochs):

- **FID:** 15.1
- **Precision:** 0.88
- **Recall:** 0.66

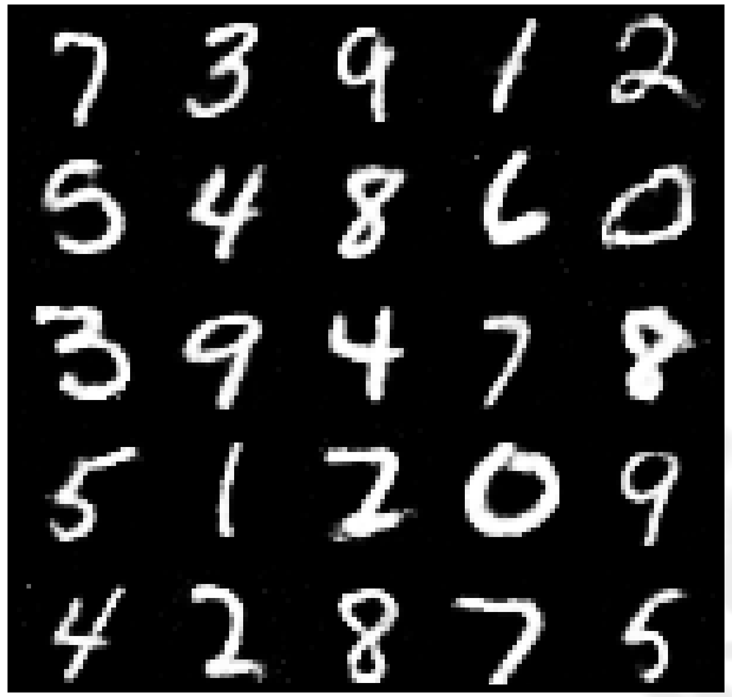
These results demonstrate the **near-max potential** of OBRS for improving sample quality and diversity.



Vanilla



OBRS K=2



OBRS K=5

4. Conclusion

4. Conclusion

- **Empirical Results:**

  OBRS greatly improved precision and recall even for a basic model, demonstrating its effectiveness.

  Incorporating rejection problem **during the training phase** proved to be very beneficial, supporting the theory behind OBRS.

- **Insights:**

  Increasing K improved PR, though **high values of** K require more testing to understand their impact.

  A challenge arises as **computational costs increase** during the **generation stage** when using larger values of K.

- **Future Directions:**

  Investigate how OBRS performs on **more complex datasets** and whether it still provides benefits at scale.

Thank you for your attention

if you have any questions, feel free to ask !