Hubness Awareness Sampling for Deep Generative Models in Generation and Evaluation

A thesis submitted in partial fulfilment
of the requirement for the degree of Doctor of Philosophy

Yuanbang Liang

May 2025

Abstract

Despite the rapid progress in Generative Adversarial Networks (GANs), several fundamental challenges remain under-explored, including reliable latent sampling, scalable evaluation, and fairness in generation.

In this work, we propose a unified framework based on hubness sampling, a principle derived from the observation that high-dimensional latent spaces exhibit hub latents. We show that these hub latents are better trained and contribute more to the synthesis of high-quality images. Leveraging this insight, we develop an a priori latent sampling method that outperforms traditional approaches such as the empirical truncation trick, both in efficiency and image quality.

Building on this foundation, we address the computational bottlenecks in evaluating generative models on large datasets. We introduce efficient precision and recall (eP&R) metrics that retain fidelity to the original metrics while significantly reducing computation through hubness-aware sampling and approximate nearest neighbor techniques.

Finally, we extend hubness sampling to promote fairness and diversity in GAN training. Without requiring labels or additional supervision, hubness sampling improves representation across sensitive attributes such as ethnicity, gender, and age, applied to various state-of-the-art GAN architectures, including StyleGAN, Diffusion-GAN, and GANFormer.

In conclusion, this work demonstrates that hubness sampling offers a versatile and powerful toolset for improving image quality, evaluation efficiency, and fairness in generative modeling, while also highlighting opportunities for further optimization in its computational cost.

Contents

List of Figures

List of Tables

List of Algorithms

List of Publications

The work introduced in this thesis is based on the following publications:

- **Yuanbang Liang**, Jing Wu, Yu-Kun Lai, Yipeng Qin. “Exploring and exploiting hubness priors for high-quality GAN latent sampling. (**Spotlight**)” Proceedings of the 39th International Conference on Machine Learning (ICML), 2022.
- **Yuanbang Liang**, Jing Wu, Yu-Kun Lai, Yipeng Qin. “Efficient Precision and Recall Metrics for Assessing Generative Models using Hubness-aware Sampling. (**Spotlight**)” Proceedings of the 41st International Conference on Machine Learning (ICML), 2024.

Other publications:

- **Yuanbang Liang**, Bhavesh Garg, Paul Rosin, and Yipeng Qin. “Deep generative model based rate-distortion for image downscaling assessment. (**Oral, top 0.8%**)” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.
- Yuxiang Liao*, **Yuanbang Liang***, Yipeng Qin, Hantao Liu, and Irena Spasic. 2024. CID at RRG24: Attempting in a Conditionally Initiated Decoding of Radiology Report Generation with Clinical Entities. In Proceedings of the 23rd Workshop on Biomedical Natural Language Processing. Association for Computational Linguistics (ACL).
- Shuang Song, **Yuanbang Liang**, Jing Wu, Yu-Kun Lai, Yipeng Qin. “Feature Proliferation – the ”Cancer” in StyleGAN and its Treatments.” Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023

Acknowledgements

I would like to thank the UK Engineering and Physical Sciences Research Council (EPSRC) to support this research through the Doctoral Training Partnerships (DTP) with No. EP/T517951/1 (2599521).

I am also grateful to express our gratitude to the Advanced Research Computing at Cardiff (ARCCA) for their invaluable computational support, which facilitated the successful completion of this work.

I would like to express my deepest gratitude to my esteemed supervisors, Dr. Yipeng Qin, Prof. Yu-Kun Lai, and Dr. Jing Wu, for their invaluable guidance, unwavering support, and insightful mentorship throughout my academic journey. Their expertise and encouragement have been instrumental in shaping my research and professional growth.

I would like to sincerely thank my family for their unwavering support, encouragement, and unconditional love. Their belief in me has been a constant source of strength and motivation throughout my journey.

Chapter 1

Introduction

Since the 1990s, computation in high-dimensional spaces has posed significant challenges in machine learning and knowledge retrieval [4, 10, 47, 86]. As dimensionality increases, data points become sparse, leading to the well-known "curse of dimensionality", where traditional indexing and algorithmic techniques suffer from reduced efficiency and effectiveness [29, 77]. This sparsity impacts various tasks, including nearest-neighbor search, clustering, and classification, necessitating the development of novel strategies to manage and exploit high-dimensional data structures effectively.

By applying advanced machine learning techniques, such as deep neural networks, probabilistic modeling, and self-supervised learning, generative AI can synthesize highly realistic text, images, music, and even intricate designs. This unprecedented level of creativity and adaptability has propelled its adoption across a diverse range of fields, from natural language processing and digital art to scientific research and automated content generation. Models like GPT, Stable Diffusion, and music-generating AI have demonstrated the potential to create human-like outputs with minimal supervision, bridging the gap between artificial intelligence and human cognition. However, despite these advances, generative models still face numerous challenges related to controllability and reliability. Issues such as mode collapse, unintended biases, and a lack of interpretability hinder their practical deployment across critical applications. Moreover, ensuring unbiased and diversity in generated content remains a significant research challenge.

Since the latent vectors in generative models reside in high-dimensional spaces, this thesis explores the impact of the “curse of dimensionality” within generative spaces. By using high-dimensional theory, we aim to enhance the quality of generated images, mitigate dataset bias, and reduce redundancy in precision and recall metrics for assessing generative models. Through a deeper understanding of high-dimensional sampling and its effects, this research provides novel insights into optimizing generative processes and improving model evaluation efficiency.

1.1 Background

Today, generative Artificial Intelligence (AI) has undergone remarkable advancements, leading to its widespread application across numerous domains, including linguistics, graphics, image synthesis, and video generation. These developments have fundamentally transformed the way humans interact with technology, enabling more sophisticated and creative AI-driven solutions. For image generation, cutting-edge models like Stable Diffusion [112] and Style-GAN[65, 66, 68, 126] have redefined content creation by enabling the generation of highly detailed and realistic images. This capability has broad implications for industries such as marketing, fashion, entertainment, and game development, where high-quality visual assets are essential.

Generative models operate in high-dimensional latent spaces, where each data sample (*e.g.* latent code and feature of generative image) is represented as a high-dimensional vector. While this allows for expressive and complex representations, it also introduces several computational and theoretical challenges. One of the most significant issues is the curse of dimensionality, which refers to the phenomenon where data points become increasingly sparse as the number of dimensions grows [15]. This sparsity affects various machine learning tasks, including clustering, similarity search, and nearest-neighbour computations.

In generative modeling, the high-dimensional nature of latent spaces impacts sampling efficiency, training stability, and evaluation metrics. For example, traditional distance-based methods, such as k-nearest neighbours (k-NN), become less reliable in high-dimensional settings because of the concentration of distances, where the distances between most points become nearly indistinguishable, reducing the reliability of neighbour ranking [37, 76, 130]. Moreover, data distributions in high-dimensional spaces often exhibit hubness, a phenomenon where certain points (hubs) appear more frequently as nearest neighbors than others [107]. Understanding and leveraging these properties is crucial for improving generative model training, evaluation, and sampling strategies.

This thesis explores the interaction between generative models and high-dimensional space, particularly focusing on the hubness phenomenon and contributes to the broader understanding of high-dimensional generative modeling, offering new insights into optimizing generative processes and improving model robustness.

1.2 Motivation

Generative Adversarial Networks (GANs) have achieved remarkable success in generating high-quality images by learning a structured latent space. However, sampling meaningful and high-quality latent vectors remains a persistent challenge. Many existing approaches rely on heuristics, such as the truncation trick, which restricts the latent space to a high-density region to enhance sample quality, or simplistic strategies like the "cherry-pick". While effective in practice, these methods lack a strong theoretical foundation and may inadvertently discard diverse yet high-quality samples, limiting the full potential of the generative model.

In addition to improving sampling strategies, evaluating generative models also presents computational challenges. Precision and Recall (P&R) metrics, commonly used to assess generative performance, rely on distance computations in feature space, often using methods like k-nearest neighbors (k-NN). Both the latent space and the feature space in generative modeling are high-dimensional (typically with dimensions $d > 100$). According to research on the hubness phenomenon [107], high-dimensional vector spaces exhibit an inherent structural property where certain data points, termed hubs, disproportionately appear as nearest neighbors and we claim that this phenomenon may create computation redundancy.

Given these challenges, our work aims to explore and leverage the structural properties of high-dimensional spaces, specifically the hubness phenomenon, in the context of generative modeling. By understanding how hubness influences both latent space sampling and generative model evaluation, we seek to develop more efficient, theoretically grounded approaches to improve the quality, diversity, and assessment of generated samples.

1.3 Research questions

As discussed earlier, the primary objective of this thesis is to investigate the "curse of dimensionality" in the latent space of generative models and develop a novel algorithm to generate high-quality synthetic images while enhancing the efficiency of computing precision and recall metrics for model evaluation. To achieve this, the following research questions will be addressed:

- The hubness latents are in the high density area of the latent space, so are the hubness latents mapping to the high-quality generated images?
- Can leveraging hubness features enhance the computational efficiency of precision and recall metrics for generative model evaluation?
- Can hubness sampling techniques improve generative diversity and mitigate dataset biases?

A detailed examination based on a literature review, experiments, and analysis will be conducted for each one of the questions above. These questions answers hold the potential to provide helpful information about the benefits and limitations of using affordances for generative models.

1.4 Aim and Objectives

This thesis aims to investigate the hubness phenomenon widely existing in the latent space of generative models and the hubness not always to be the negative component in the image synthesis, also having much improvement in different targets. To achieve this aim, the following objectives are pursued:

- Analyzing the Relationship Between Hubness and Generative Image Quality, GANs map latent vectors to images in a high-dimensional space, where some latents (termed hubs) occur frequently as nearest neighbors because of the hubness phenomenon, investigating how hubness latent vectors map to generated images and assess their impact on output quality.

- Due to the presence of hubness in high-dimensional feature spaces, certain data points frequently appear in neighbor sets and the computation of precision and recall for the evaluation of generative models using k-NN, therefore, we aim to analyze whether this redundancy affects computational efficiency and whether alternative hubness-aware approaches can improve the reliability of precision and recall metrics by optimizing the computation for generative models.

- These hub latents are often better trained and result in higher-quality image generation. we want to design and implement hubness-aware sampling methods for training GANs, assessing their impact on training dynamics, image quality, and model diversity, investigating whether leveraging hub latents can enhance model stability and improve convergence efficiency.

By pursuing these objectives, this thesis aims to contribute to discovering the relationship between the generative models and the hubness phenomenon and to proving that the hubness phenomenon, 'dimensional curve', is not always the negative effect for the deep learning.

1.5 Outline

This outline presents a concise summary of the thesis' content comprised of six chapters, aiming to provide a clear overview of the research presented in the following chapters.

Chapter 2 provides a comprehensive review of the relevant literature on various deep generative models, along with their associated evaluation metrics. It also explores the hubness phenomenon and discusses existing reduction techniques. Furthermore, the chapter introduces different sampling strategies for generative models, highlighting their significance in improving generation quality and efficiency.

Chapter 3 explores the hubness phenomenon in the latent space of pretrained GAN models, focusing on leveraging hubness priors for high-quality latent sampling. This chapter introduces, for the first time, a formal definition of the hubness value in the latent space and investigates its correlation with the quality of synthesized images. Furthermore, it examines the potential relationship between the truncation trick and hubness points, suggesting that truncation may inherently align with hubness-aware sampling strategies.

Chapter 4 investigates the redundancy in precision and recall metrics assessments for generative models, highlighting how feature vectors with high hubness values can be leveraged to reduce redundancy and enhance computational efficiency. This chapter also presents extensive experiments across various models and datasets, demonstrating the effectiveness and applicability of the proposed approach.

Chapter 5 introduces the hubness-aware sampling method, which aims to enhance the diversity of GAN-generated outputs while mitigating class imbalance within the dataset. By leveraging hubness samplings, this approach helps reduce bias across different classes, leading to a more balanced and representative generative model.

Chapter 6 concludes the thesis by summarising the main contributions and achievements of this research work. In addition, the limitations and challenges of the approaches developed in this thesis are discussed, and propose future research directions.

Overall, this thesis aims to contribute to the ongoing efforts to enhance the understanding and utilization of the hubness phenomenon in generative models, particularly in GANs. By investigating its impact on image quality, model diversity, and evaluation efficiency, this work provides valuable insights that can inform future advances in generative AI and high-dimensional data analysis.

Chapter 2

Literature review

2.1 Overview

This chapter begins by introducing the fundamental concepts of deep generative models, highlighting some of the most significant models and their applications. The primary objective of generative models is to learn the patterns and distribution of a given dataset (training set) in order to generate new data that is similar to the original data.

Generative models can generally be categorized into four types: Generative Adversarial Networks (GANs), Likelihood-based Models, Energy-based Models and Hybrid Models. These models allow for the generation of different images by manipulating latent vectors, providing versatility in creative and analytical applications.

The evaluation metrics for generative models differ from those used for discriminative models, which typically focus on accuracy with respect to labels. Instead, generative model metrics assess the quality of generated data by measuring the distributional distance between the generated images and real images. Key metrics include: 1)Fréchet Inception Distance (FID) and Inception Score (IS) 2)Kernel Inception Distance (KID) 3)Precision and Recall (P&R) In addition to discussing these metrics, this chapter will introduce the concept of hubness, a phenomenon that commonly occurs in high-dimensional spaces [107, 135, 136, 137]. Hubness has been shown to affect real data classification, particularly in areas such as gene expression, time-series data, and electroencephalography.

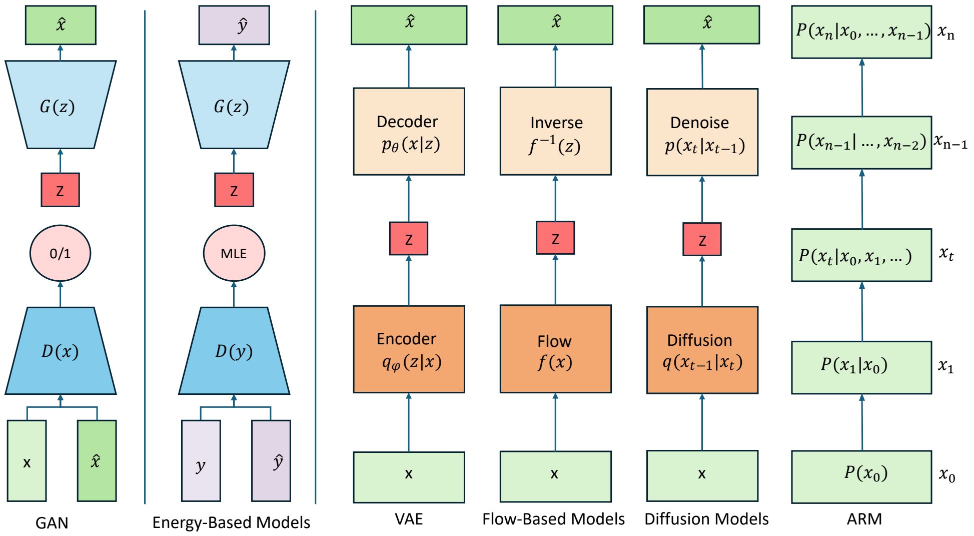


Figure 2.1: Overview of deep generative models, including generative adversarial networks (GAN), energy-based models, variational autoencoder (VAE), flow-based models, and diffusion models.

This thesis will specifically focus on the application of generative models in image generation, discussing the most state-of-the-art and widely used models in this domain.

2.2 Deep Generative Models

Generative models are a class of machine learning algorithms that focus on generating new data points from an underlying distribution. These models learn to understand the distribution of the input data and can generate new samples that are similar to the training data. Deep generative models have achieved impressive results in image-generation tasks. As discussed before, Major models include four types, Generative Adversarial Networks (GANs), Likelihood-based Models, Energy-based Models, and Hybrid Models, which will be reviewed below respectively and the structures of these models are shown in the Fig. 2.1.

2.2.1 Generative Adversarial Networks (GANs).

GANs [42] train two neural networks concurrently: a generator network that produces synthetic outputs, and a discriminator network that distinguishes real from synthetic data. The two networks are pitted against each other in a minimax adversarial game, where the generator tries to fool the discriminator and the discriminator tries to identify fakes. This creates a constant evolutionary pressure that enables GANs to produce increasingly realistic outputs. A core innovation of GANs is using the discriminator not just for evaluation, but directly in the training loop to guide the generator. GANs can produce sharp and photorealistic images, but are notoriously difficult to train due to mode collapse, optimization instability, and other challenges. This has led to significant efforts to stabilize GAN training [7, 46, 97, 105, 164]. Along with these efforts, researchers have extended the synthesis capabilities of GANs to a variety of image generation tasks, including unconditional image synthesis [62, 65, 66, 68, 120], conditional image synthesis [18, 96, 97], image-to-image translation [58, 103, 165, 166], image editing [1, 2], etc.

2.2.2 Energy-based Models

Different with the likelihood-based models, the core component of Energy-based models is the energy function, $E(x)$, where x is the input, which associate the measured compatibility, *i.e.* energy value, to each configuration of the variables [81, 131]. The foundation of the early energy-based models is Hopfield Network [55, 56] and Boltzmann Machine [3], discussing the idea of an energy function applied on model training. The modern generative Energy-based Models have evolved to handle more complex data and larger datasets, often using deep neural networks to parameterize the energy function, *e.g.* EBM [36], DDGM [72], f-EBM [155] and GEBM [44]. Despite their potential, generative EBMs face several challenges. 1)Computational Complexity is the primary challenge in training, because of partition function, normalizing the energy function into the probability distribution [155, 163]. 2) Sampling Efficiency: Generating samples from a trained EBM can be computationally expensive due to the need for MCMC methods, which may require many iterations to converge to low-energy states [5, 53, 100].

2.2.3 Likelihood-based Models

Likelihood-based models are to learn the probability distribution of the training data, $p(x)$, where x represents a data instance (*e.g.*, an image or a sentence). They aim to maximize the likelihood of the observed data. Autoregressive models (ARMs), variational autoencoders (VAEs), normalizing flows, and diffusion models are the four main types of likelihood-based generative models.

Variational Autoencoders (VAEs).

VAEs use variational inference to approximate posterior inference, training an encoder network to map inputs to a latent space z and a decoder network to reconstruct the inputs from the latents [74]. Mathematicly, the VAEs should encourage the latent distribution to possess distributions ($q(z)$) closed to the prior distribution ($p(z|x)$); hence it can be write as $\mathrm{KL}(p(z|x)||q(z))$, where KL is Kullback-Leibler(KL) divergence. Despite their elegant theory, images generated by early VAEs are usually blurry, which was improved by incorporating latent quantization to produce models like VQ-VAE [142] and VQ-VAE2 [109] that can synthesize sharp and high-resolution images.

Normalizing Flows

Normalizing flows are trained on factorized distributions [32, 102, 110], enabling efficient and exact evaluation of both sampling and density estimation. These models effectively transform a simple normal distribution into a complex distribution that closely approximates the real data distribution. To enhance and optimize these models, REALNVP [33] introduced non-volume preserving transformations, improving the compatibility with unsupervised learning tasks such as sampling and log-density estimation. GLOW [75] further advanced this framework by applying invertible 1x1 convolutions to replace fixed permutations, thereby enhancing the model's learning capabilities without increasing computational time. However, training flow models often requires managing the constraints imposed by the Jacobian determinant, necessitating deeper models to achieve desired performance levels. To address this, Self-Normalizing Flows were proposed, reducing computational complexity while optimizing the model to achieve comparable data likelihood values [71].

Diffusion Models.

Diffusion models [31, 54, 127] train a neural network to reverse a stochastic diffusion process. They start with a data sample x and apply a diffusion process that gradually adds Gaussian noise over multiple timesteps to arrive at a noisy sample x_t, and then it is trained to predict the noise added at each timestep during the forward process, enabling the reconstruction of the original image x from its noisy versions x_t. By training the model to denoise the diffused samples, it learns to generate high-quality samples. Diffusion models avoid problematic generator-discriminator training and provide exact log-likelihoods. However, sampling requires running the full diffusion process in reverse, which is computationally expensive. Extensions like DDIM [128] have made diffusion models more efficient. Thanks to their training stability, diffusion models have been widely used in text-to-image synthesis and editing tasks, including the Latent Diffusion model [112] that inspired Stable Diffusion, DALLE-2 [108], Imagen [115], DreamBooth [114], MUSE [23].

Recent advancements in diffusion models, when combined with language models, have introduced the concept of the prompt space, enabling text-driven synthesis and editing of images [51, 143]. This approach demonstrates the existence of concept subspaces within the text embedding space of diffusion models. Diffusion models have seen widespread application across various domains, including super-resolution (SR) [41, 82, 158], image restoration [89, 90, 149, 167], image editing [28, 51, 69, 152, 162], and image recognition [12, 24, 48, 153].

Autoregressive Models (ARMs)

Generative Autoregressive models (GARMs), grounded in Bayesian networks, generate each data point by conditioning on preceding ones, and they have been widely applied in the generation of images, text, and video. In [140, 141], the introduction of PixelCNN and PixelRNN demonstrated the feasibility of generating complex natural images one pixel and one color channel at a time, necessitating thousands of predictions per image [139]. This sequential generation process introduces significant limitations, including slow inference and exposure bias, where errors can accumulate over time. Furthermore, because GARMs predict the next element as a single best guess, they may constrain model diversity, reduce the exploration of alternative possibilities in the output sequence, and struggle with capturing multimodal data distributions.

2.2.4 Hybrid Models

These combine elements of different types of generative models, such as VAEs with GANs (VAE-GANs) [80, 95, 157], Energy-based GANs [30, 104, 163], Energy-based Diffusion [35, 40] and Diffusion with GANs [145, 150], to leverage the strengths of both approaches.These models have become increasingly popular for their ability to generate high-quality data and capture complex data distributions. Generative hybrid models represent a significant advancement in the field of machine learning by integrating the strengths of various generative approaches, demonstrating their ability to generate high-quality, realistic data while capturing complex data distributions.

2.3 Metrics for Assessing Deep Generative Models

Metrics serve as essential tools for evaluating the performance and quality of deep models, playing a crucial role in validating their effectiveness. This section will focus on the metrics used specifically for assessing generative models. Unlike traditional models, where accuracy is a primary measure, generative models are evaluated based on the distance between the distributions of the real dataset and the generated dataset. The discussion will cover four of the most widely used metrics: Fréchet Inception Distance (FID), Inception Score (IS), Kernel Inception Distance (KID), and Precision & Recall.

2.3.1 Fréchet Inception Distance (FID)

The FID metric, introduced by [52], computes the Fréchet distance between features of the real and generated images extracted by an Inception-V3 feature extractor. A lower FID score indicates a higher similarity between the distributions of real and generated images, implying better image quality and diversity in the generated samples. Thus, the computation of FID consumes $O(n)$ time and space as the feature extraction takes $O(n)$ time and space while the Fréchet distance computation also takes linear time $O(n)$ when using a fixed Inception-V3 network.

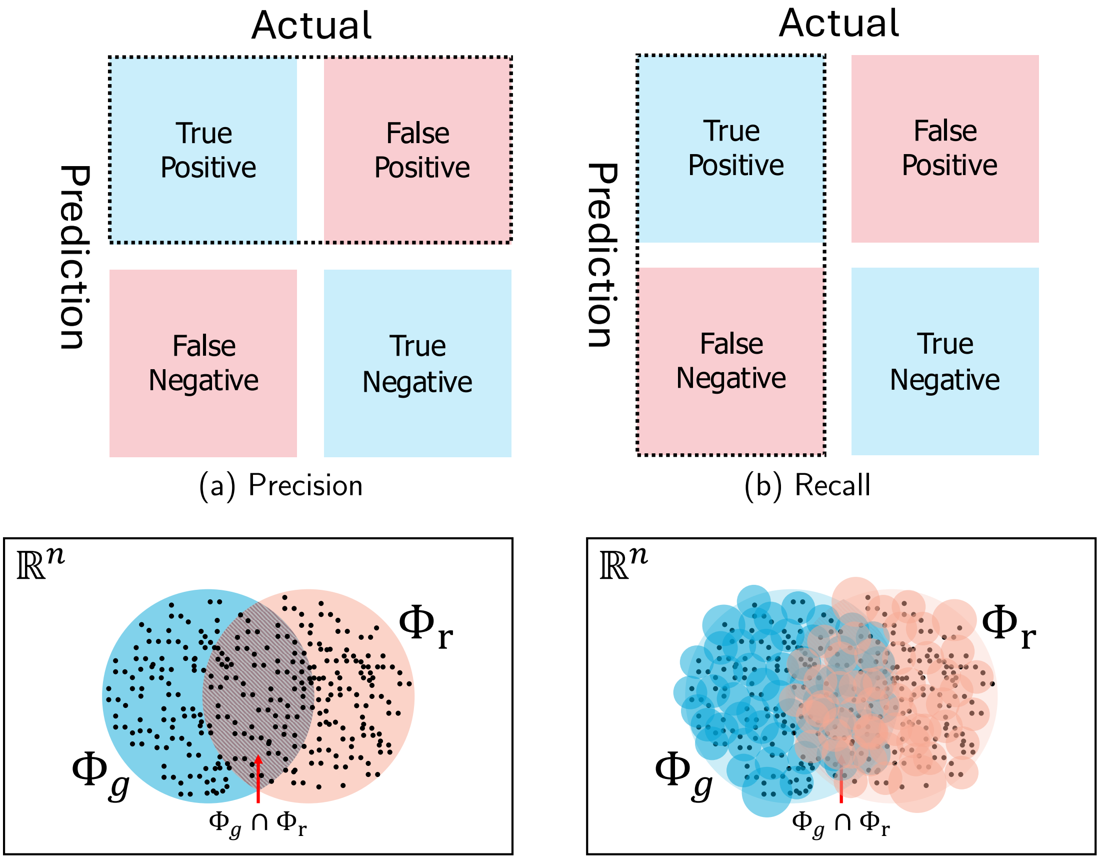
2.3.2 Inception Score (IS)

The IS metric, proposed by [117], uses a pre-trained Inception-v3 classification model to compute the conditional label distribution $p(y|x)$ for each generated image x. IS measures two main aspects: i) the diversity of generated images, indicated by the entropy of $p(y|x)$, and the precision of generated images, indicated by the KL divergence between the marginal distribution $p(y)$ and the conditional distribution $p(y|x)$ for each x. A higher IS generally indicates the model can generate more realistic and diverse images. Similar to that of FID, the computation of IS consumes $O(n)$ time and space as the feature extraction takes $O(n)$ time and space while the computation of IS metric takes linear time $O(n)$.

The FID and IS metrics were improved by [27] to FID_∞ and IS_∞, which apply Quasi-Monte Carlo integration to reduce bias and improve reliability of them for finite samples.

2.3.3 Kernel Inception Distance (KID)

KID is also a widely used metric for generative models, computing the maximum mean discrepancy (MMD) of the real distriution and generative distribution with kernel function [6, 17]. MMD is an integral probability metric to compare the difference between X and Y using $\mathbf{E}_{x \sim p}[f(x)]$ and $\mathbf{E}_{y \sim q}[f(y)]$ $(x \in X,\ y \in Y)$ to denote expectations with respect to p and q, where $x \sim p$ indicates x has distribution p and $y \sim q$ indicates y has distribution q [45]. As demonstrated in [14], the empirical Wasserstein distance, commonly used in FID, can introduce bias in the gradients, particularly with finite sample sizes, which negatively impacts stochastic gradient descent and expectation estimation. Consequently, when FID is computed on small sample sizes, this bias can become significant. To address this issue, KID utilizes MMD by mapping inputs into Reproducing Kernel Hilbert Spaces (RKHS) with unit



(a) Precision

(b) Recall

(c) True manifold

(d) Estimating manifold

Figure 2.2: The precision and recall for traditional model and generative models.(a)(b): The traditional precision and recall computing with the labels to compute the precision $= \frac{\#\text{TP}}{\#(\text{TP+FP})}$ and the recall $= \frac{\#\text{TP}}{\#(\text{TP+FN})}$, where TP: True Positive, FP: False Positive and FN: False Negative. (c)(d): The illustration of the precision and recall for assessing generative models.

balls, thereby ensuring that, unlike FID, KID remains unbiased even with smaller sample sizes.

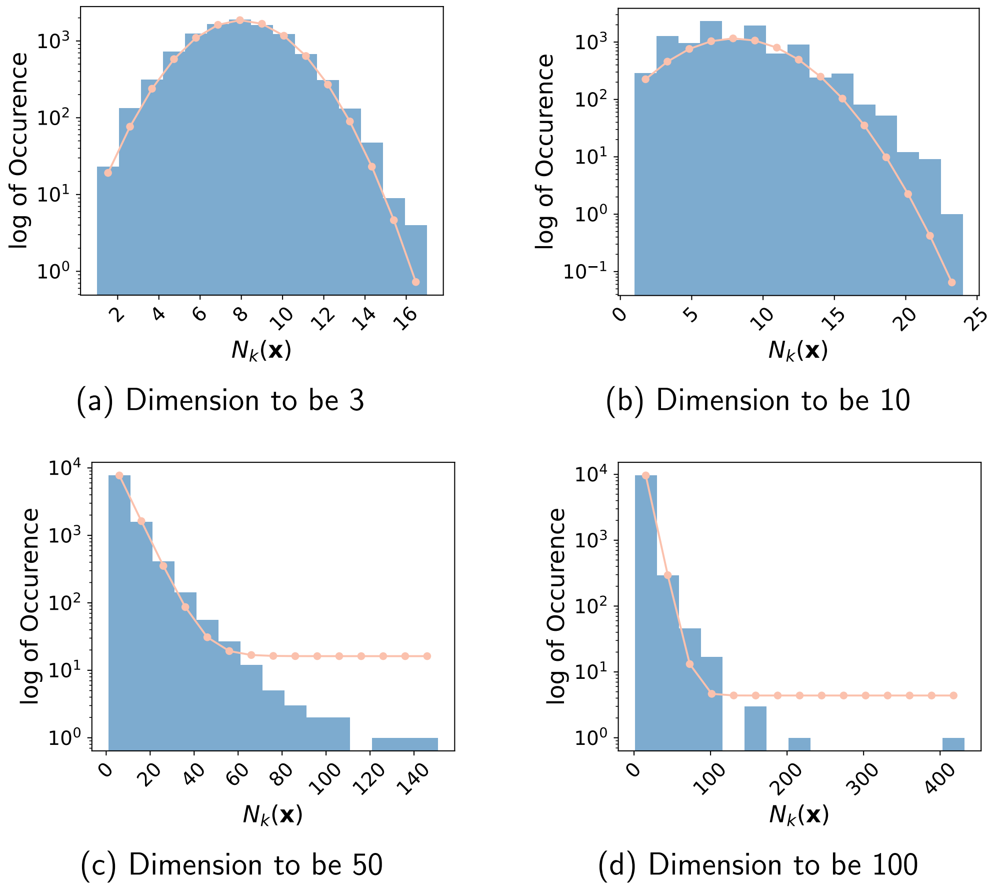
2.3.4 Precision and Recall (P&R).

Despite their effectiveness, FID and IS metrics are single scores and thus cannot differentiate between specific failure modes, *e.g.*, mode dropping or collapsing [88], or provide insights into the underlying causes of poor performance. The P&R metrics were employed to address this issue [78, 116, 124]. In short, precision measures the percentage of generated samples that are considered high-quality and indistinguishable from real data, indicating the *quality* of generated samples; recall measures the percentage of all potential high-quality samples that the generator was able to produce, indicating the *diversity* of generated samples, as shown in the Fig. 2.2.

Specifically, [116] formulated P&R through relative probabilistic densities between the distributions of real and generated images, which are non-trivial to estimate. Addressing this issue, they proposed a practical algorithm based on the maximal achievable values of an alternative definition of P&R. Their method was generalized by [124] to accommodate arbitrary distributions and link P&R to type I and type II errors of likelihood ratio classifiers. Observing that the P&R implementation proposed by [116] relies on relative densities and thus cannot correctly identify mode collapse/truncation, [78] propose to model the real and generated image manifolds directly using the k-nearest neighbors of samples, which is the state-of-the-art (SOTA) version of P&R for assessing generative models. Although more accurate, their method is computationally expensive as k-NN consumes $O(n^2)$ time and space, making their metrics infeasible to compute using commodity hardware on the large datasets used by modern deep generative models.

2.4 Hubness Phenomenon

Hubness is a widely recognized phenomenon of nearest neighbors search in high-dimensional spaces that arises from the well-known “curse of dimensionality” [107].



(a) Dimension to be 3

(b) Dimension to be 10

(c) Dimension to be 50

(d) Dimension to be 100

Figure 2.3: Hubness Phenomenon in random samplings ($\mathbf{x} \subseteq \mathrm{R}^d$). $N_k(\mathbf{x})$ is the number of vectors from $\mathbf{x}$ that have $x \in \mathbf{x}$ included in their list of k nearest neighbors [107], also called hubs value, and the more detail you can see in Fig 2.4 and in this figure, we assume the k to be 3. In the low-dimension samplings, (a) and (b), the distribution of the $N_k(\mathbf{x})$ is not obviously skewed. In the high-dimension sampings, (c) and (d), the distribution skews to the right and there are some example with large $N_k(\mathbf{x})$

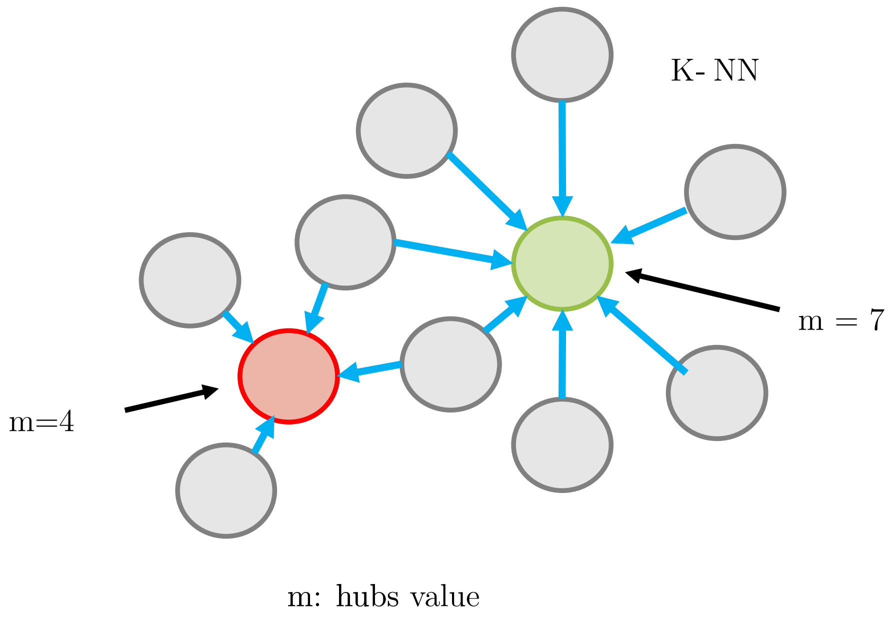


Figure 2.4: The example to explain the hubness phenomenon, we first randomly sample n latents from the latent distribution and perform k-nearest neighbour on them. Then, we compute the hubs value m of a given latent as how many times it is among the k-nearest neighbours of other latents. For example, the hubs value of the red latent is 4 and that of the green point is 7.

It pertains to the inherent characteristics of data distributions in high-dimensional spaces and reveals a counter-intuitive fact: even with uniform distributions, high dimensionality gives rise to the emergence of "popular" nearest neighbors [98, 99, 107], as Fig 2.3 shown, *i.e.*, points that are significantly more likely to be among the k-nearest neighbors of other points within a given sample set, denoted as *hubs* points. In other words, the hub points are those that are much more likely to be among the k-nearest neighbours of other points in a sample set. This fact poses challenges for algorithms that rely on nearest neighbor search. Addressing such challenges, hubness-aware methods were proposed and applied in various areas, *e.g.* gene expression classification [20, 21], time-series classification [135], electroencephalograph classification [22] and few/zero-shot learning [34, 122, 160].

2.4.1 Hubness Reduction Method

Hubness phenomenon is described as a kind of the impact of the notorious "curse of dimensionality", appearing disproportionately often as nearest neighbors, skewing learning algorithms and causing performance issues in tasks like classification or

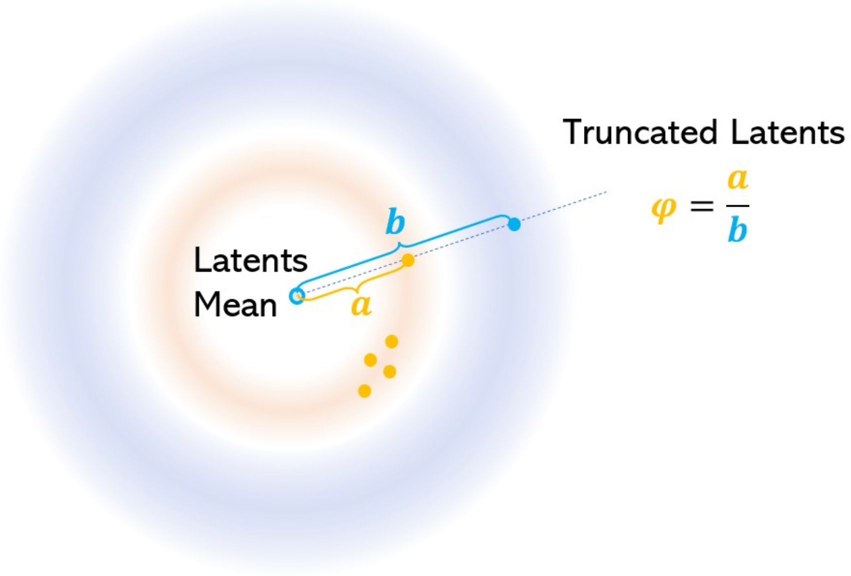


Figure 2.5: The illustration of the truncation trick which truncates randomly sampled latents to their mean with a scaling parameter ϕ. The smaller ϕ will make the latents closer to the mean and get the higher quality of generative images; in versa, the latent will far away to the center and the generative quality will decrease.

clustering. Several methods have been proposed to mitigate the effects of hubness, *e.g.*Mutual Proximity (MP) measure, Local Scaling (LS) measure, Shared-Neighbors (SN) measures, Two dissimilarity (DSL) measures, Hubness-aware k-Nearest Neighbor (H-kNN) and Z-Score (ZS) Normalization. The main idea to mitigate the hubness, it is to balance the two points (x, y) distance $d_{x,y}$ and the related surrounding points, which can flatten the the density gradient which is expressed as a crucial reason to cause the hubness [49].

However, instead of mitigation, recent works have demonstrated that depending on the task, the hubness phenomenon can be very useful. This thesis will show that the hubness phenomenon can be used as a prior to identify high-quality latents in GAN latent spaces. Following the same philosophy, this work introduces a new method to improve the computational efficiency of P&R metrics and the diversity of the GAN-series models' manifold by incorporating hubness-aware sampling.

2.5 GAN Latent Sampling.

To avoid the issue of unstable image generation without curation, “cherry-picking”, and obtain high-quality synthesized images, three workaround solutions were proposed: i) the truncation trick [18, 66, 93] as shown in Fig. 2.5 ii) Importance Sampling [19, 84] and iii) Interpolating in Latent Space [147]. Between them, the first approach is a naive solution as one can always “cherry-pick” high-quality ones from a set of synthesized images in an *a posteriori* manner with visual inspection. Obviously, this method is inefficient as it requires intensive human labor and is not applicable for large-scale image synthesis tasks. Unlike “cherry-picking”, the truncation trick is an automatic method that can synthesize high-quality images by normalizing sampled latents to be close to their mean. However, it is a purely *empirical* method with few insights. Importance sampling is employed to enhance the efficiency of the sampling process, particularly in regions of the latent space that are more likely to yield high-quality samples. Additionally, interpolating between points in the latent space facilitates the generation of smooth transitions between samples, which is valuable for investigating the continuity and underlying structure of the latent space. In this thesis, it proposes a novel latent sampling method for GANs based on the observation of *hubness* phenomenon in their high dimensional latent spaces, which is efficient with solid theoretical insights and also shows that the truncation trick is a naive approximation of our method due to the “central clustering effect” of *hub* latents.

2.6 Fairness in Generative Models

Fairness in training is a widely studied topic in classification tasks, aiming to reduce biases in input data, ensure independent decision-making, and promote equitable predictions across different groups, even when those groups exhibit distinct characteristics or historical disparities. In contrast, the objective of fairness in generative models is to achieve balanced training, also known as equal representation, which refers to generating samples that follow a uniform distribution across categories [118, 134, 151]. Recent research suggests that the fairness of GAN models can be enhanced by prelearning the feature distribution through approaches such as weak supervision and transfer learning [25, 134].

Meanwhile, fairness research is also crucial in text-to-image diffusion models, where the objective is to ensure that outputs neither favor nor exclude specific groups based on various attributes. In [38], a fair diffusion model is introduced by integrating fair instructions and fair guidance with a text encoder to promote equitable generation. Furthermore, methods for optimizing inclusive prompts and cross-attention maps are proposed in [134, 148, 159] to mitigate dataset bias and improve fairness in diffusion models. To address training set bias, weakly supervised learning is employed through density ratio estimation. However, this approach is prone to estimation errors due to the density-chasm problem. Recent work [73] proposes a time-dependent density ratio estimation method to alleviate this issue.

Additionally, [132] introduces a fairness metric for generative models using statistical methods, offering a quantitative framework for evaluating fairness in generated outputs. In recent advancements, the AI-Face benchmark [87] has been developed to facilitate the training, evaluation, and analysis of fairness in generative face models.

2.7 Summary

The literature review provides a comprehensive overview of deep generative models, focusing on their theory, key types, applications, and the metrics used for their evaluation. It begins by introducing the concept of generative models, which are designed to learn patterns and distributions from datasets to generate new, similar data.

The review categorizes generative models into four main types: Generative Adversarial Networks (GANs), Likelihood-based Models, Energy-based Models, and Hybrid Models. GANs involve a generator and a discriminator in a competitive training process, which has proven effective in producing high-quality, realistic images, despite challenges like mode collapse and optimization instability. Likelihood-based models, including Autoregressive Models, Variational Autoencoders (VAEs), Normalizing Flows, and Diffusion Models, offer different approaches to generating data, each with its own strengths and limitations in terms of complexity, training stability, and output quality. Energy-based models, which utilize an energy function to measure the compatibility of different data configurations, are also discussed. These models, while promising, face challenges such as computational complexity and sampling efficiency. Hybrid models, which combine elements of different generative approaches, represent an advanced strategy to leverage the strengths of multiple methods, enhancing the quality and diversity of generated data.

The review also touches on the "hubness" phenomenon in high-dimensional spaces, which impacts data classification and is a focus of this thesis in relation to GANs. In addition, it covers key datasets used in training deep generative models, such as LSUN, AFHQ, CelebA-HQ and FFHQ.

Finally, the review discusses metrics for assessing the quality of generative models. Unlike traditional discrimination models that are evaluated based on accuracy, generative models are assessed by measuring the distributional distance between generated and real data. Metrics such as Fréchet Inception Distance (FID), Inception Score (IS), Kernel Inception Distance (KID), and Precision & Recall are highlighted as the most popular methods for evaluating the performance of deep generative models. These tools are crucial for ensuring the effectiveness of the models and guiding further improvements.

Chapter 3

Hubness Sampling for High-Quality GAN Latent Sampling

Despite the extensive studies on Generative Adversarial Networks (GANs), how to reliably sample high-quality images from their latent spaces remains an under-explored topic. In this section, we propose a novel GAN latent sampling method by exploring and exploiting the *hubness priors* of GAN latent distributions. Our key insight is that the high dimensionality of the GAN latent space will inevitably lead to the emergence of *hub* latents that usually have much larger sampling densities than other latents in the latent space. As a result, these *hub* latents are better trained and thus contribute more to the synthesis of high-quality images. Unlike the a posteriori "cherry-picking", our method is highly efficient as it is an a priori method that identifies high-quality latents before the synthesis of images. Furthermore, we show that the well-known but purely empirical truncation trick is a naive approximation of the central clustering effect of *hub* latents, which not only uncovers the rationale of the truncation trick, but also indicates the superiority and fundamentality of our method. Extensive experimental results demonstrate the effectiveness of the proposed method.

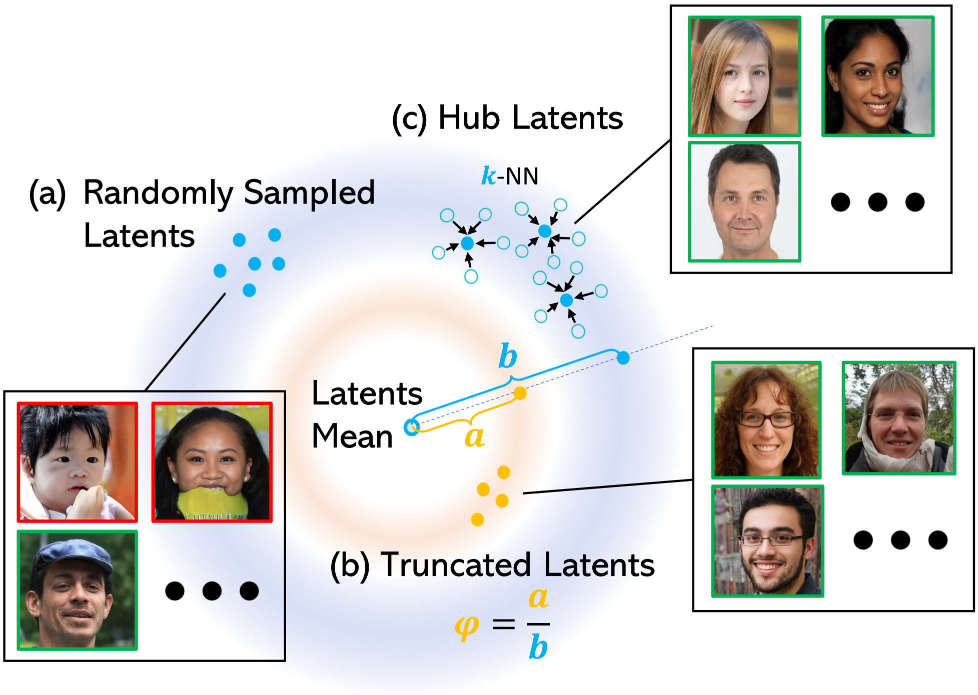


Figure 3.1: Our method *vs.* random latent sampling and the truncation trick [18, 66, 93]. All images are generated using StyleGAN2 [67]. (a) Random latent sampling yields both high-quality (green box) and low-quality (red box) images; (b) The truncation trick improves the quality of synthesized images by empirically truncating randomly sampled latents according to a scaling parameter ψ (*e.g.* $\psi = 0.7$), which is a naive approximation of the "central clustering effect" of our *hub* latents; (c) Our method identifies high-quality latents as the *hub* latents that are more likely to be among the k-nearest neighbors of other latents [107]. The blue and orange rings illustrate the high-dimensional Gaussian (latent) distribution [94] and their truncated version respectively.

3.1 Introduction

Generative adversarial networks (GANs) are a type of deep generative models that have revolutionized a variety of applications in computer vision and computer graphics, *e.g.* image synthesis [66, 103, 166], image editing [1, 2, 138], image-to-image translation [58, 111, 165]. Among them, novel image synthesis via random latent sampling is the most fundamental. It not only generates novel instances from the data distribution, but also measures how close the learned distribution is to the data distribution. Through the lens of the quality of synthesized images, we have witnessed significant progress in GANs over the past several years. Specifically, starting from the groundbreaking vanilla GAN [43], DCGAN [106] laid the foundation for GAN architectures as deep convolutional neural networks; ProGAN [63] showed that GANs can generate high-quality images at high resolutions; BigGAN [18] addressed the problem of class-conditional image synthesis; the StyleGAN series [64, 66, 67] further boosted the quality and controllability of synthesized images with their style-based generator architectures and several novel techniques.

Nevertheless, with such improvements, the quality variance among images generated by randomly sampled latents has become increasingly striking (Fig. 3.1). Without curation, the quality of GAN synthesized images can occasionally be very low, which hinders the deployment of GANs in real-world applications. As a naive solution, "cherry-picking" is commonly used to select high-quality images from those synthesized with randomly sampled latents in an *a posteriori* manner. However, in the absence of reliable quantitative measures of the quality of a single GAN-synthesized image[1], existing "cherry-picking" methods are barely manual, thereby being tedious and unscalable. Addressing this issue, the well-known "truncation trick" [18, 66, 93] was proposed, which "truncates" randomly sampled GAN latents towards their mean based on the observation that the images synthesized from close-to-mean latents are usually of higher quality. Although effective, the truncation trick is a purely empirical "trick" that brings few new insights to the community.

In this chapter, we propose a novel latent sampling method for GANs by exploring and exploiting the *hubness* phenomenon [107] in their latent spaces, which facilitates their synthesis of high-quality images in an *a priori* manner. Specifically, our key insights include: i) the high dimensionality of the GAN latent space will inevitably lead to the emergence of *hub* latents that are much more likely to be among the nearest neighbors of other latents in the latent space, *i.e.* the hubness phenomenon; ii) in general, the quality of a GAN synthesized image is positively correlated with the *hub value* of its corresponding latent, *i.e.* the number of times a latent becomes a k-nearest neighbor (k-NN) of other latents in a given latent sample set. We believe that this positive correlation originates from the well-known close relationship between k-NN and density estimation. In other words, a higher *hub value* usually indicates a higher sampling density, which has a positive effect on the training and

[1] Existing quantitative measures like FID and Inception scores are all statistical ones that are only applicable to distributions.

thus the quality of synthesized images. Therefore, we formulate the above insights as the proposed *hubness priors* and propose a corresponding method to sample high-quality GAN latents that yield high-quality synthesized images. Compared to "cherry-picking", our method is highly efficient as it is *a priori* (*i.e.* our high-quality latents are determined before the synthesis of images) and automatic (*i.e.* with little human-intervention). Furthermore, we show that the well-known truncation trick is a naive approximation of the "central clustering effect" of our *hub* latents [107]. This not only uncovers the rationale of the truncation trick, but also indicates that our method is superior and more fundamental. Extensive experimental results demonstrate the effectiveness of the proposed method.

In summary, our contributions include:

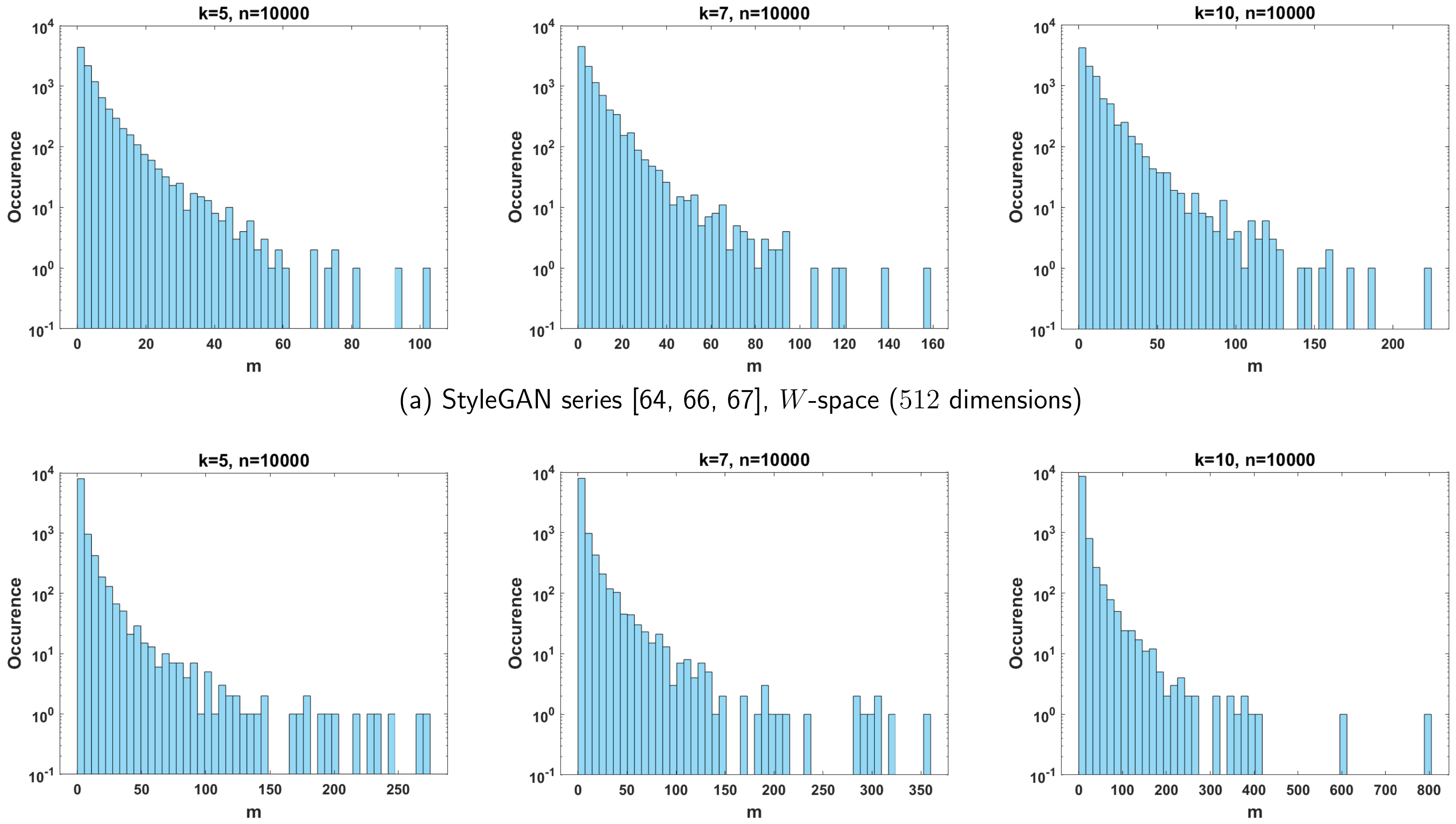
- We uncover the existence of *hubness* phenomenon in the GAN latent space, which has a significant correlation with the quality of GAN synthesized images, *i.e.* the proposed *hubness priors*.

- We propose a novel GAN latent sampling algorithm that identifies high-quality *hub* latents based on our *hubness priors*, which allows efficient and high-quality image synthesis for GANs.

- We show that the well-known truncation trick is a naive approximation of the "central clustering effect" of our *hub* latents. This not only uncovers the rationale of the truncation trick, but also indicates that our method is superior and more fundamental.

3.2 Hubness Priors for GAN Latent Sampling

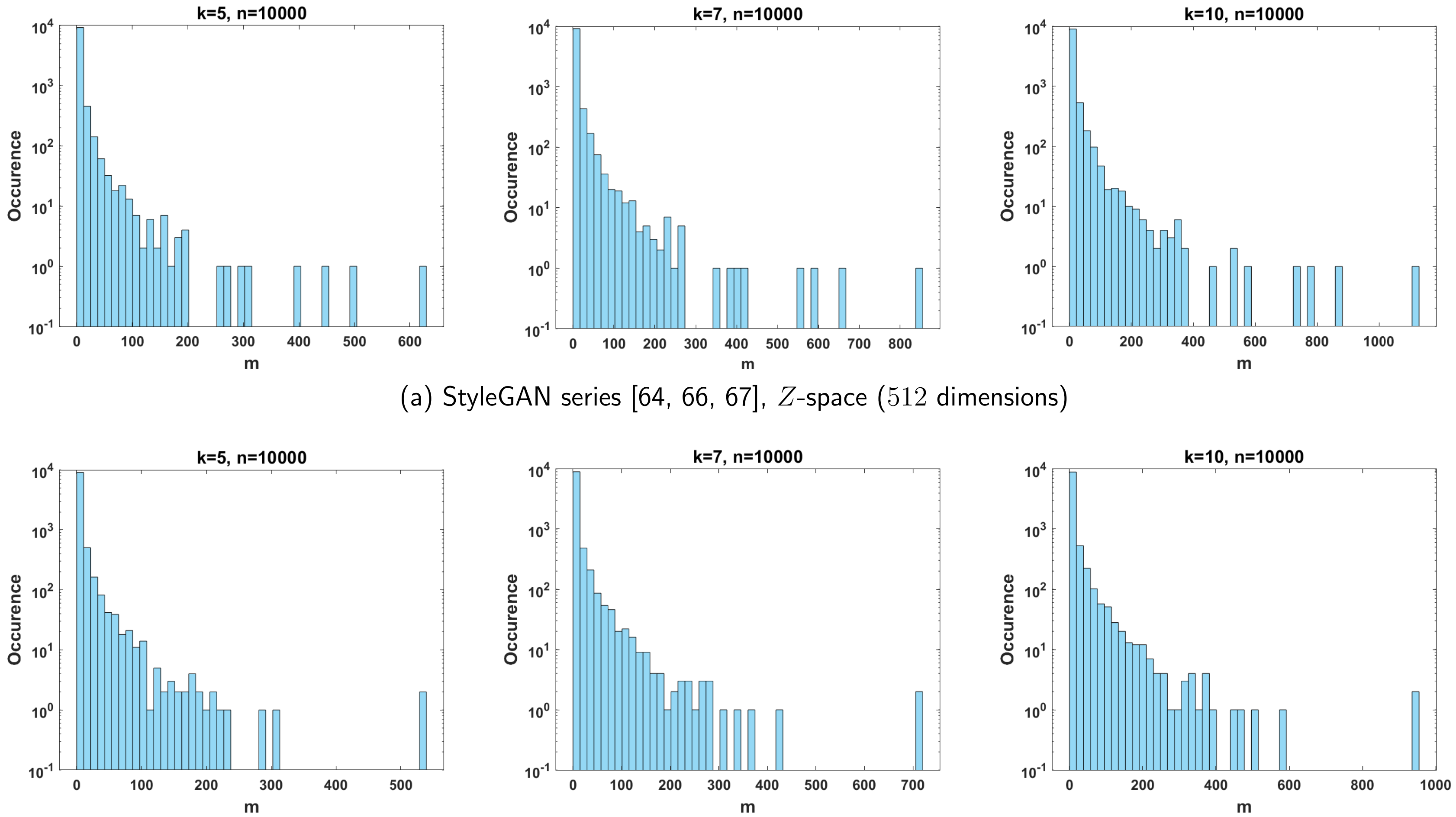
In this section, we first explore the hubness of GAN latents (Section 3.2.1) and then exploit the insights obtained as priors to develop a novel algorithm for the sampling of high-quality latents for GANs (Section 3.2.2).



(a) StyleGAN series [64, 66, 67], W-space (512 dimensions)

(b) BigGAN [18], Z-space (128 dimensions)

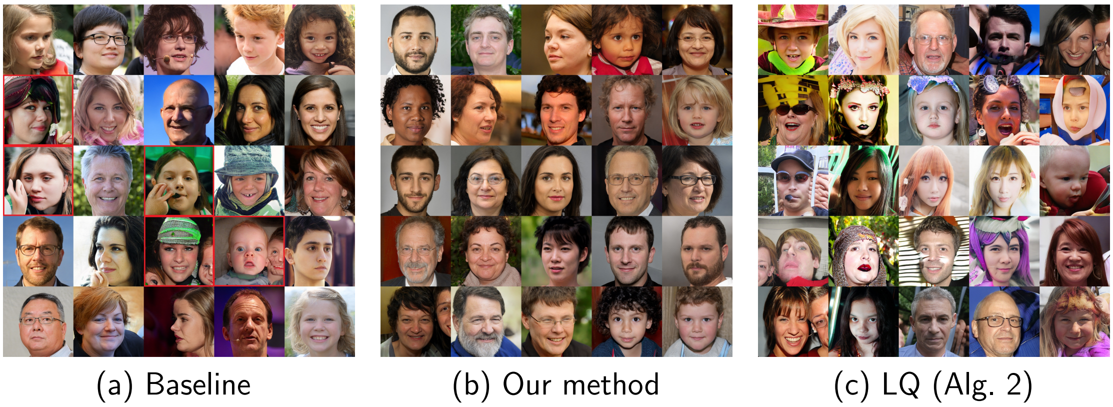
Figure 3.2: Distributions of m-hub latents for state-of-the-art GANs, $k = 5, 7, 10$ (the k-NN algorithm) and $n = 10000$ (size of latent sample set S). All distributions are highly tailed to the right, which shows the existence of hubness phenomenon [107] in GAN latent spaces. Note that y-axis is in log-scale.



(a) StyleGAN series [64, 66, 67], Z-space (512 dimensions)

(b) ProGAN [63], Z-space* (512 dimensions)

Figure 3.3: Distributions of m-hub latents for state-of-the-art GANs, $k = 5, 7, 10$ (the k-NN algorithm) and $n = 10000$ (size of latent sample set S). *: Although both are 512-dimensional, the ProGAN [63] latents are sampled directly from $\mathcal{N}(0, I)$ while the StyleGAN latents further normalized the sampled latents to be of the same norm [66]. All distributions are highly tailed to the right, which demonstrates the existence of hubness phenomenon [107] in GAN latent spaces.



(a) Baseline (b) Our method (c) LQ (Alg. 2)

Figure 3.4: (a) and (b): Effectiveness of our method (hubness priors) against the baseline (random latent sampling). We use $n = 10000$, $k = 5$ and hub value threshold $t = 50$ in our method. The StyleGAN2 [67] images generated using our method are almost always of high quality while those generated using the baseline contain both high-quality and low-quality (red boxes) results. (c): Low-quality (LQ) StyleGAN2 [67] images generated using the **reversed** version of our method, *i.e.* Algorithm 2, where $n = 10000$, $k = 5$ and hub value threshold $t_{lq} = 1$. Almost all images are of low quality.

3.2.1 Exploring Hubness of GAN Latents

Inspired by previous studies on the hubness phenomenon of data distributions in high dimensional space [107], let Z

$subseteq\mathbb{R}^d$ be a d-dimensional GAN latent space, $S = \{z_1, z_2, ..., z_n\}$, $z_i \in Z$ be a set of latents sampled from a d-dimensional standard normal distribution $\mathcal{N}(0, I)$, k be the parameter of the k-nearest neighbor algorithm, we define *m-hub latents* as:

Definition 3.2.1. Latent code z_i $(1 \leq i \leq n)$ is an m-hub latent if z_i is among the k-nearest neighbors of m $(m < n)$ sampled latents in S, where m is the *hub value* of z_i.

With the above definition, we explore the hubness of GAN latents by investigating the distributions of m-hub latents in the latent spaces of state-of-the-art GANs [18, 63, 64, 66, 67]. As Fig. 3.2 and 3.3 shows, it can be observed that the distributions of m-hub latents are highly tailed to the right. Thus, we argue that the samples of GAN latents are not uniformly distributed and that a small portion of them are much more likely to be close to other latents in the latent space, *i.e.*, with large m. Therefore, these latents tend to have larger sampling densities and are thus better trained than other latents during GAN training. Based on the heuristics that well-trained latents are more likely to yield high-quality images, we conjecture that the hubness phenomenon can be used as priors to identify GAN latents that generate high-quality results:

Conjecture 3.2.2. **(Hubness Priors)** The quality of GAN synthesized images and the hub values m of their corresponding latents are positively correlated.

Please see Section 3.3.2 for an empirical justification of our conjecture.

Algorithm 1 GAN Latent Sampling with Hubness Priors

Input: a set of GAN latents $S = \{z_1, z_2, ..., z_n\}$ randomly sampled from a standard normal distribution $\mathcal{N}(0, I)$, a hyper-parameter k, a threshold t

Output: S_{hq}

```
  # Step 1
  $m_{1,2,...,n} \leftarrow 0$
  for $i \leftarrow 1$ to $n$ do
    $\{\mathrm{idx}_1, \mathrm{idx}_2, ...\mathrm{idx}_k\} \leftarrow k\text{-NN}(z_i)$
    for $j \leftarrow 1$ to $k$ do
      $m_{\mathrm{idx}_j} \leftarrow m_{\mathrm{idx}_j} + 1$
    end for
  end for
  # Step 2
  $S_{hq} \leftarrow \emptyset$
  for $i \leftarrow 1$ to $n$ do
    if $m_i \geq t$ then
      $S_{hq} \leftarrow S_{hq} \cup z_i$
    end if
  end for
```

Remark on Random Latent Sampling Previously, it was widely believed that GAN latents are *unbiased* as they are sampled from a simple but well-behaved noise distribution, *i.e.,* the standard normal distribution $\mathcal{N}(0, I)$. In high-dimensional spaces, the concentration of measure phenomenon causes the mass of this distribution to concentrate on a thin hyperspherical shell [94]. Since the distribution is isotropic, this geometry implies that all sampled latents effectively approximate a uniform distribution across the shell's surface; they possess similar norms[2] and should theoretically contribute to the sampling process in an equivalent manner. While in this chapter, we counter this popular belief by showing that GAN latents are actually

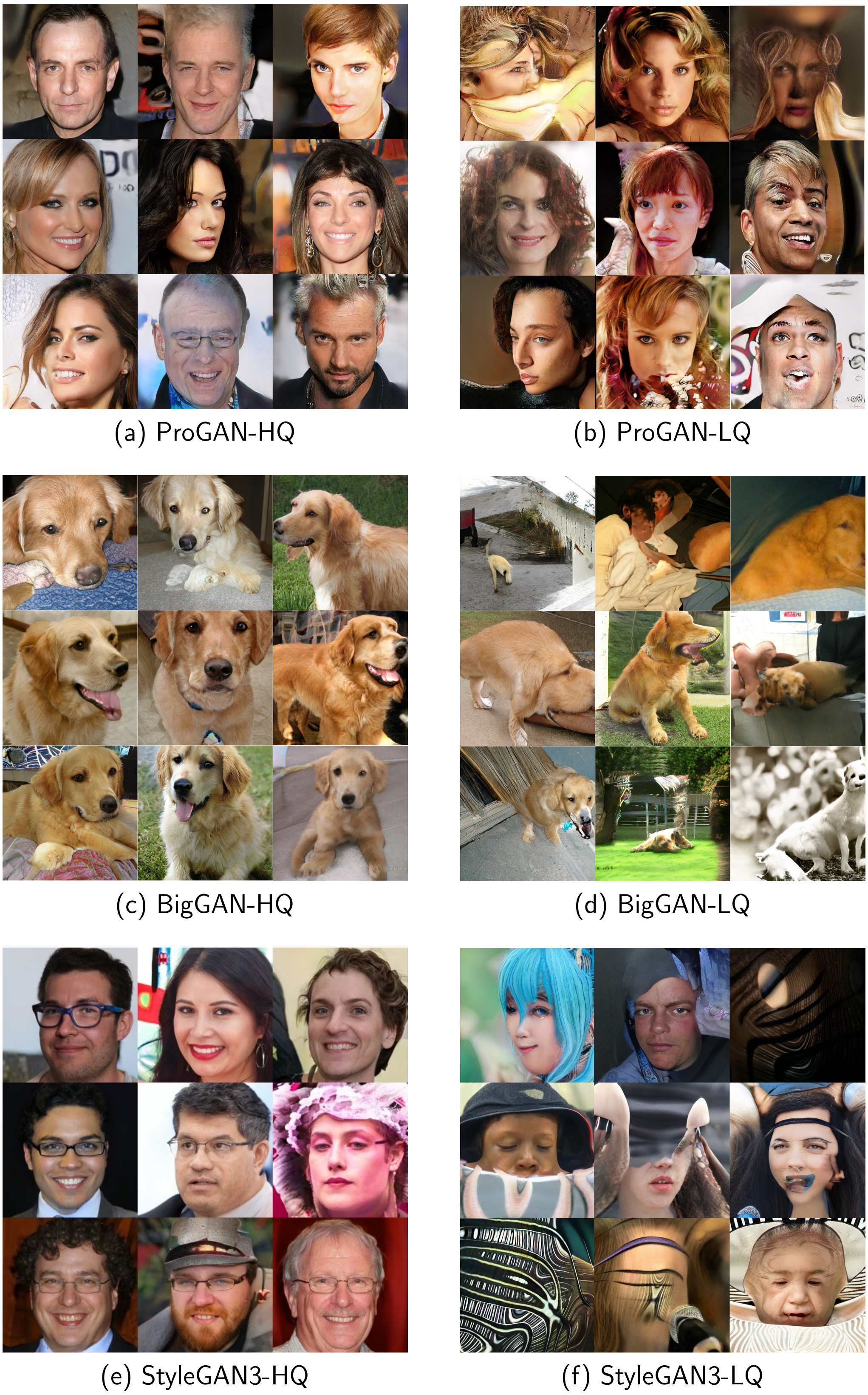
[2] In latest implementations [64, 66, 67], the latents are explicitly normalized to be of the same norm.

biased from the observation of hubness phenomenon in GAN latent spaces. Among all latents, the *hub* ones tend to have higher sampling densities and are thus better trained by GANs, thereby generating higher quality images.

3.2.2 Exploiting Hubness Priors for High-quality GAN Latent Sampling

As Conjecture 3.2.2 states, the identification of high-quality GAN latents relies on their hub values m. Thus, given a set of GAN latents $S = \{z_1, z_2, ..., z_n\}$ randomly sampled from a standard normal distribution $\mathcal{N}(0, I)$, a hyper-parameter k, and a threshold t, we utilize the proposed hubness priors and design a simple two-step GAN latent sampling algorithm: First, we compute the hub value m_i for each latent $z_i \in S$ using a standard k-NN (k-nearest neighbor) algorithm; Second, we identify z_i as a high-quality latent if m_i is larger than a user-defined threshold t, and add z_i into a set S_{hq}. The set S_{hq} is the output of our algorithm, which contains all the high-quality latents identified. Algorithm 1 shows the pseudocode of our algorithm. Note that our algorithm is fundamental and widely applicable to different types of GANs as long as they sample latents from a standard normal distribution, *e.g.* conditional GANs [18].

Relationship to Truncation Trick. To our knowledge, the truncation trick [18, 66, 93] is the only a priori method to sample high-quality GAN latents before our work, which is based on a heuristic that high-quality latents are those close to their mean. However, such a heuristic is purely empirical with few insights. Surprisingly, the proposed *hubness priors* have revealed the rationale of the truncation trick: the *hub* latents obtained by our method tend to cluster towards their mean [107]. Thus, we argue that the well-known truncation trick is a naive approximation of our method as it only captures near-mean *hub* latents but overlooks those that are relatively far from the mean. Please see Section 3.3.6 for an empirical justification of our claims.



(a) ProGAN-HQ

(b) ProGAN-LQ

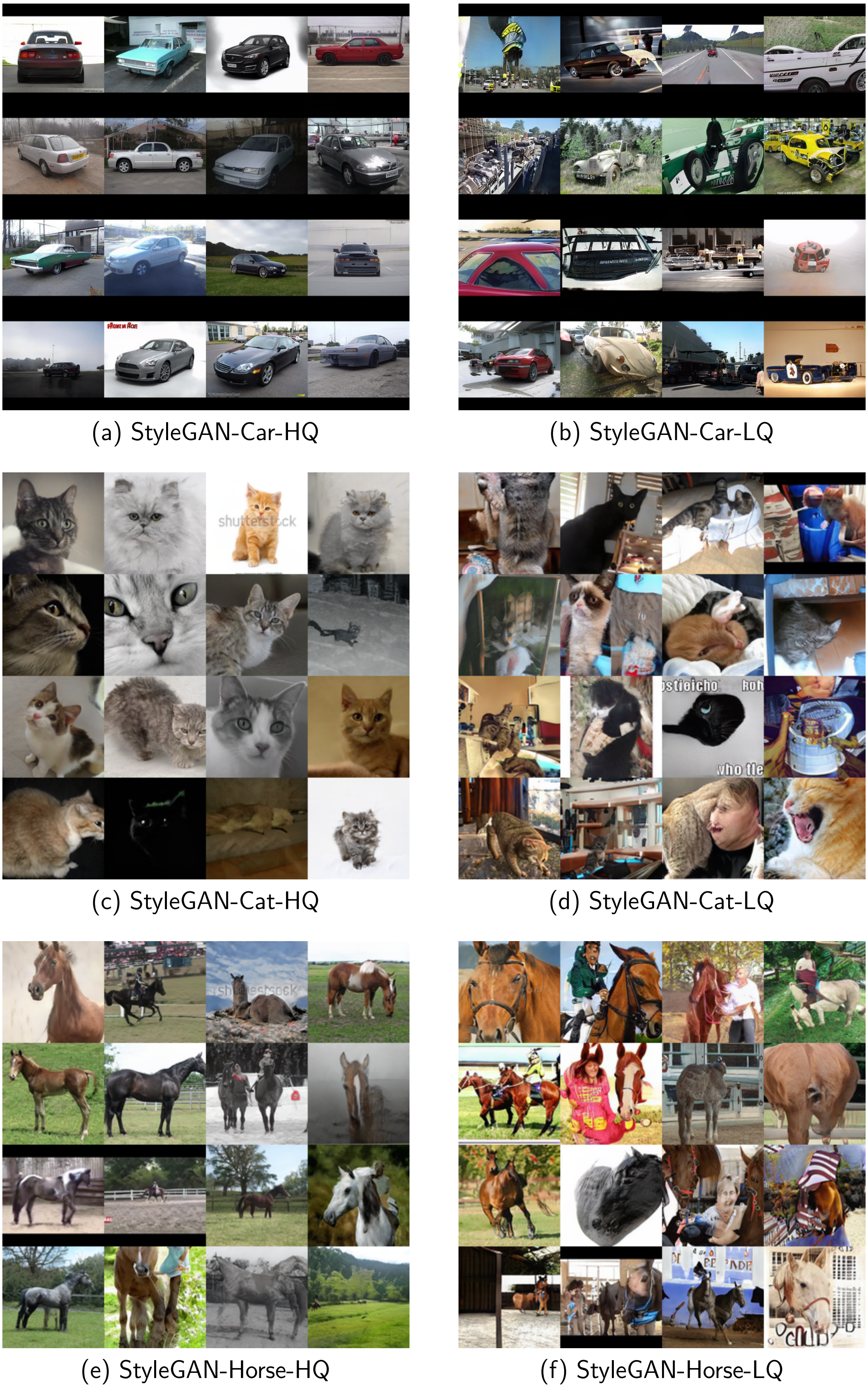
(c) BigGAN-HQ

(d) BigGAN-LQ

(e) StyleGAN3-HQ

(f) StyleGAN3-LQ

Figure 3.5: Performance of our method on ProGAN [63], BigGAN [18] and StyleGAN3 [64]. It can be observed that our method works well on all GAN architectures. (a) and (b), (c) and (d), (e) and (f) are images synthesized using high-quality (HQ) and low-quality (LQ) latents obtained by our method with ProGAN, BigGAN and StyleGAN3 respectively. We use Algorithm 1 to obtain HQ latents and Algorithm 2 to obtain LQ latents respectively. We use $n = 10000$, $k = 5$, $t = 50$ and $t_{lq} = 1$.



(a) StyleGAN-Car-HQ

(b) StyleGAN-Car-LQ

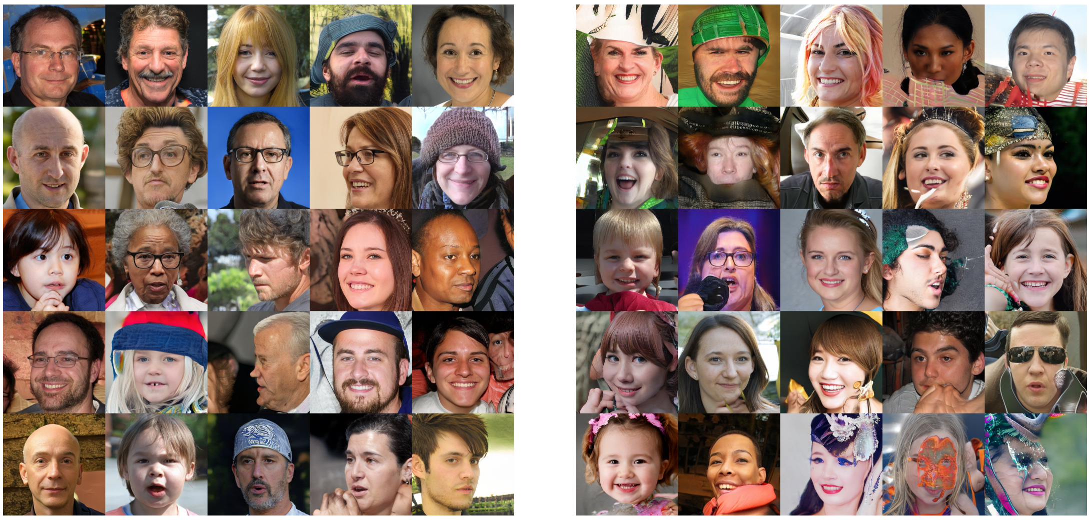
(c) StyleGAN-Cat-HQ

(d) StyleGAN-Cat-LQ

(e) StyleGAN-Horse-HQ

(f) StyleGAN-Horse-LQ

Figure 3.6: Performance of our method on StyleGAN2 pretrained on different image domains. It can be observed that our method works well on all domains. (a) and (b), (c) and (d) are images synthesized from high-quality (HQ) and low-quality (LQ) latents obtained by our method using a StyleGAN2 pretrained on the cars domain and the cats domain, respectively. We use Algorithm 1 to obtain HQ latents and Algorithm 2 to obtain LQ latents respectively. We use $n = 10000$, $k = 5$, $t = 50$ and $t_{lq} = 1$.



(a) StyleGAN-Z-HQ

(b) StyleGAN-Z-LQ

Figure 3.7: Performance of our method on StyleGAN2's Z-space. We use $n = 10000$, $k = 5$, $t = 50$ and $t_{lq} = 1$.

3.3 Experimental Results

3.3.1 Experimental Setup

Due to its a priori nature, our method allows for the sampling of high-quality GAN latents before the synthesis of images. Thus, for the sampling of StyleGAN's Z-space and other GANs' latents, we use an Intel(R) Core(TM) i7-10875H CPU; for the sampling of StyleGAN's W-space latents, we use a GeForce RTX 2080 Ti GPU as the computation involves passing Z-space latents through a fully-connected mapping network [64, 66, 67]. For the synthesis of high-quality images, we use publicly-released Github codes of StyleGANs[3] [64, 66, 67], BigGAN[4] [18], ProGAN[5] [63] with a GeForce RTX 2080 Ti GPU. Unless specified, all results are generated with the W-space of StyleGAN2 [67]. All quantitative results are averaged over three runs. **Note that JPEG is applied to compress the synthesized images to meet the size limit.**

[3] StyleGAN2,3: `https://github.com/NVlabs/stylegan2`, `https://github.com/NVlabs/stylegan3`.

[4] `https://github.com/ajbrock/BigGAN-PyTorch`

[5] `https://github.com/tkarras/progressive_growing_of_gans`

3.3.2 Effectiveness of Hubs Priors

As Figs. 3.4 (a) and (b) show, we compare the images generated by StyleGAN2 [67] using our method with those generated using the baseline, *i.e.* random latent sampling. It can be observed that our method consistently yields high-quality images while the baseline generates both high-quality and low-quality images, which demonstrates the effectiveness of the proposed *hubness priors*. Quantitatively, we observed better FID scores of images generated using our method than those by the baseline (Table 3.3).

Low-quality Latents. As Conjecture 3.2.2 implies, the proposed *hubness priors* can also be used to identify low-quality latents that yield unrealistic synthesized images. Thus, as a complement to high-quality latent sampling, we implement low-quality GAN latent sampling by reversing the thresholding scheme in Algorithm 1 to $m_i \leq t_{lq}$ and have Algorithm 2. The pseudocode of our low-quality GAN latent sampling algorithm (Algorithm 2) is a simple inverse of Algorithm 1, using a different thresholding scheme. Generative adversarial networks (GANs) are a type of deep generative models that have revolutionized a variety of applications in computer vision and computer graphics, *e.g.* image synthesis [66, 103, 166], image editing [1, 2, 138], image-to-image translation [58, 111, 165]. As Fig. 3.4 (c) shows, almost all synthesized images are of low quality, which justifies the effectiveness of the proposed *hubness priors*.

In fact, our *hubness priors* can be used to sort all sampled latents into a ***hubness spectrum*** according to their hub values m in Fig. 3.8, where the quality of images changes from high to low from left to right with decreasing m.

3.3.3 Versatility

To demonstrate the versatility of our method, we show that it generalizes across different GAN architectures, different image domains and different latent spaces of the StyleGAN series [64, 66, 67].

Different GAN Architectures. As Fig. 3.5 shows, to justify that our method works across different GAN architectures, we show that our method also works on

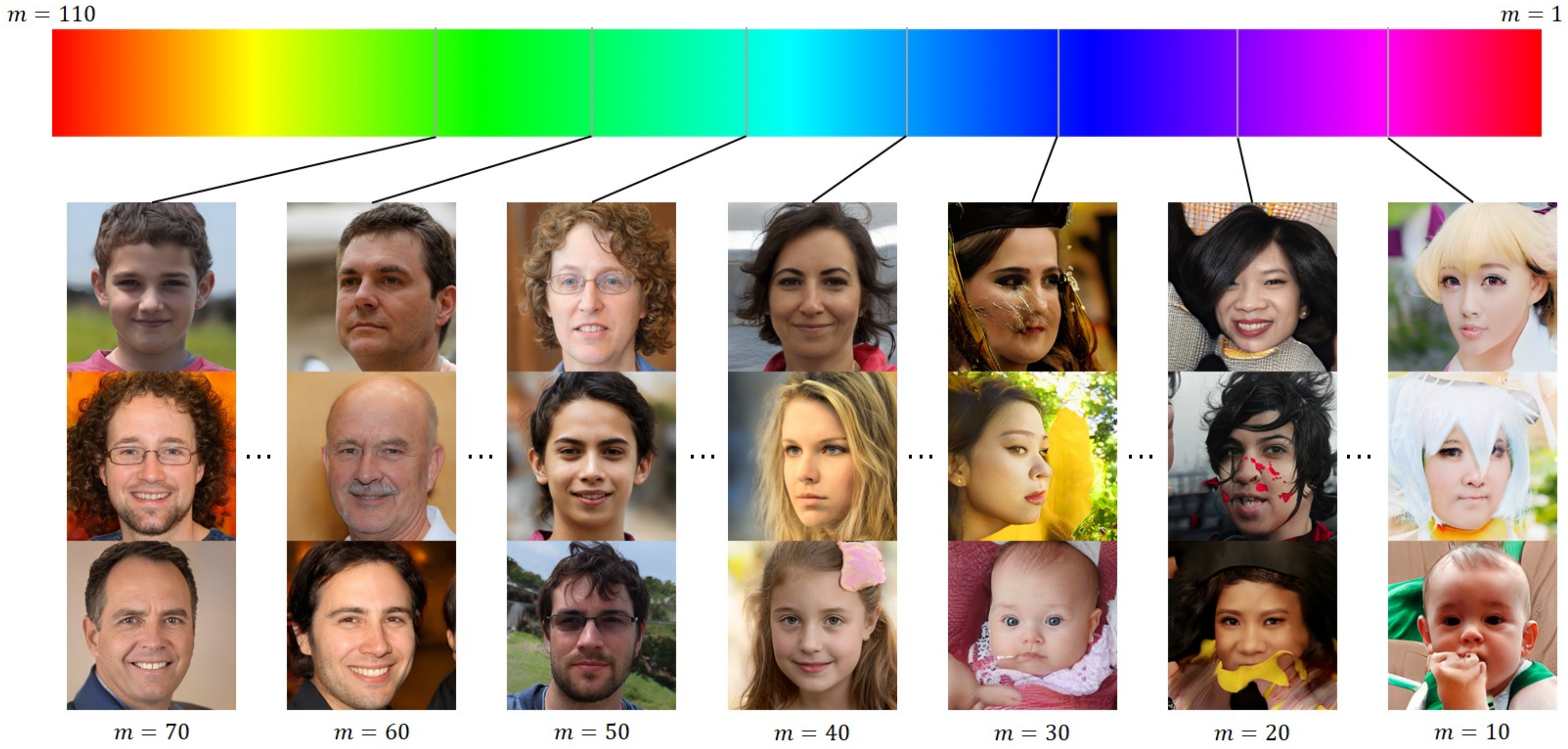


Figure 3.8: The *hubness spectrum* of StyleGAN2 [67] synthesized images ranked according to their *hub* values m. We use $n = 10000$, $k = 5$. Note that the spectrum is highly tailed to the left and thus there are few images in the range $m = (70, 110)$.

Algorithm 2 Low-quality GAN Latent Sampling with Hubness Priors

Input: a set of GAN latents $S = \{z_1, z_2, ..., z_n\}$ sampled from a standard normal distribution $\mathcal{N}(0, I)$, a hyper-parameter k, a threshold t_{lq}

Output: S_{lq}

```
# Step 1
$m_{1,2,...,n} \leftarrow 0$
for $i \leftarrow 1$ to $n$ do
    $\{\mathrm{idx}_1, \mathrm{idx}_2, ...\mathrm{idx}_k\} \leftarrow k\text{-NN}(z_i)$
    for $j \leftarrow 1$ to $k$ do
        $m_{\mathrm{idx}_j} \leftarrow m_{\mathrm{idx}_j} + 1$
    end for
end for
# Step 2
$S_{lq} \leftarrow \emptyset$
for $i \leftarrow 1$ to $n$ do
    if $m_i \leq t_{lq}$ then
        $S_{lq} \leftarrow S_{lq} \cup z_i$
    end if
end for
```

three other state-of-the-art GAN architectures, *i.e.* ProGAN [63], BigGAN [18], and the recent StyleGAN3 [64].

Different Image Domains. As Fig. 3.6 shows, to justify that our method works across different image domains, we show that our method also works on StyleGAN2 models pretrained on other images domains[6]: cars, cats and horses.

StyleGAN's Z-space. As Fig. 3.7 shows, our method also works for the Z-space of StyleGAN2 [67]. However, we observed that the quality variance of synthesized images is slightly lower when using the W-space. Thus, we propose to use the W-space for StyleGAN2.

3.3.4 Justification of Algorithmic Choices

Threshold t. In our method, given a fixed latent sample set S, the threshold t determines the trade-off[7] between image quality and number of output latents: the larger t, the higher image quality, but the fewer output images. However, as Fig. 3.9

[6] All pre-trained networks are available at: `https://github.com/NVlabs/stylegan2`.

[7] Note that this trade-off only applies to a fixed S. Our method can generate an infinite number of high-quality samples by simply using multiple latent sets S_1, S_2, ..., S_N or a larger S.

Table 3.1: FID scores of StyleGAN2 images synthesized using our method with different choices of k, t and n, whose default values are $k = 5$, $t = 50$ and $n = 10000$. We sample 2,000 images to compute the FIDs, whose rationale is discussed in Sec. 3.3.6.

k	FID↓	t	FID↓	n	FID↓
3	22.793	60	20.749	10000	22.782
5	22.782	50	22.782	20000	22.021
7	22.720	40	24.517	30000	21.679
10	22.560	35	25.412	40000	19.124

and Table 3.1 show, we observed that the image quality remains high for various choices of t. Since the image quality is not sensitive to the choice of t in a relatively large range, we suggest using $t = 50$ as the default value for the case when $n = 10000$, $k = 5$. Note that we can easily extend our algorithm to output a user-specified number of images (denoted as n') by using a revised scheme: if there are enough images in S, we first sort all images in the descending order of hub value m, and keep the top n' latents; otherwise, we successively draw more latent sets S_i and keep all $m > t$ images from them until we get n' images.

Hyper-parameter k. We tested the performance of our algorithm with various choices of $k = 1, 3, 5, 7, 10$ in the k-NN algorithm. Apart from the case when no hub latents can be found ($k = 1$), we show the results of $k = 3, 5, 7, 10$ in Fig. 3.10 and Table 3.1. It can be observed that the image quality is not sensitive to the choice of k. Nevertheless, we noticed that using a larger k yields more output hub latents for a given latent set S and threshold t, but at the cost of slightly longer computation (Sec. 3.3.5). To achieve a balance, we suggest using $k = 5$ as a default value when $n = 10000$, $t = 50$.

Size of Latent Sample Set n. As Table 3.1 shows, we also test the performance of the proposed method against various sizes $n = 10000, 20000, 30000, 40000$ of latent sample set S and Fig. 3.1 for qualitative results. Similar to above, we observed that (i) although the FID scores get slightly better with increasing n, the image quality is not sensitive to the choice of n; (ii) using a larger n yields more output hub latents but at the cost of longer computation (Sec. 3.3.5). To achieve a balance, we suggest using $n = 10000$ as a default value when $k = 5$, $t = 50$.

Table 3.2: Running time of our method using the StyleGAN2 W-space with different choices of k and n. The default parameter values are $k = 5$, $t = 50$ and $n = 10000$.

k	Time(s)	n	Time(s)
3	163s	10000	167s
5	167s	20000	647s
7	176s	30000	1272s
10	185s	40000	2554s

3.3.5 Running Time

Table 3.2 shows the running time of our method with different choices of k and n. It can be observed that the running time increases mildly with k but significantly with n.

3.3.6 Relationship with Truncation Trick

The truncation trick [18, 66, 93] has been widely used in state-of-the-art GANs. Specifically, it truncates randomly sampled latents $\mathbf{w}$ to $\mathbf{w}' = \bar{\mathbf{w}} + \psi(\mathbf{w} - \bar{\mathbf{w}})$ to obtain high-quality latents that yield high-quality synthesized images, where $\bar{\mathbf{w}}$ is the mean of a large number of randomly sampled latents, ψ is a scaling parameter. As discussed in Section 3.2.2, we argue that it is a naive approximation of our method.

Distance to the Means of Hub and All Latents. To justify our claim, we first investigate the distances of our *hub latents* to their mean and their distances to the mean of all sampled latents. As Fig. 3.12 shows, it can be observed that: i) Our hub latents are closer to both the hub mean and the all latent mean than randomly sampled latents, which justifies the "central clustering effect" of our hub latents [107]. ii) Surprisingly, the distances of most hub latents are around 6.0 to 7.0 for both cases, which is roughly the same as the distances of randomly sampled latents truncated with a parameter $\psi = 0.7$, *i.e.* the StyleGAN-recommended [66] parameter value for the truncation trick. However, StyleGAN obtained the value $\psi = 0.7$ empirically via try-and-error while we obtain it as a byproduct of our method, which justifies the superiority and fundamentality of our approach. iii) A small portion of our hub latents are of larger distances (*e.g.* around 7.5 and 8.0) to the means, which will be overlooked by the truncation trick with $\psi = 0.7$. In addition, applying the truncation

Table 3.3: Comparison of FID scores of StyleGAN2 synthesized images using our method and the truncation trick. FFHQ-1 and FFHQ-2: real images sampled from the FFHQ dataset [66]; Hubs (50): our method with $t = 50$; Truncated (0.7): truncation trick with $\psi = 0.7$; Random: random sampling. We sample 2000 latents/images for all methods compared. The FID scores between i) FFHQ-1 and FFHQ-2; and ii) Random and FFHQ-1,FFHQ-2 are used as baselines. Dist2Mean: distances of sampled latents to the all latent mean.

Methods	FID↓		Dist2Mean
	FFHQ-1	FFHQ-2	
FFHQ-2	16.505	—-	—-
Hubs (50)	21.955	23.609	6.247
Truncated (0.7)	25.097	25.127	6.893
Random	35.455	35.598	9.847

trick with $\psi = 0.8$ are prone to get low-quality latents that yield low-quality images while our “distant” hub latents are still of high quality (Fig. 3.13). This further justifies the superiority of our method against the truncation trick.

The truncation trick is an effective but naive geometric solution that exploits the high-dimensional structure revealed by the hubness phenomenon: the trick works because it acts as a rigid, distance-based filter that shrinks the latent distribution toward the mean ($\mathbf{w}' = \bar{\mathbf{w}} + \psi(\mathbf{w} - \bar{\mathbf{w}})$), thereby successfully isolating the desirable, high-quality non-hub samples which exhibit the central clustering effect. This is necessary because the statistically problematic, low-quality samples are the hub samples that are inevitably clustered at the periphery of the high-dimensional space due to the concentration of measure effect. By cutting off the peripheral region based on an empirically derived threshold like $\psi = 0.7$, the truncation trick effectively eliminates these artifact-prone hub latents , although it remains limited as a pure geometric heuristic, imperfectly discarding some high-quality samples that naturally reside just beyond that arbitrary cutoff.

FID Scores. As Table 3.3 shows, we also justify the superiority of our method by comparing the FID scores [52] of images generated by StyleGAN2 using both the truncation trick, $\psi = 0.7$ [66] and our method. Specifically, we compute the FID scores between images generated by i) real images sampled from the FFHQ datasets, *i.e.* FFHQ-1 and FFHQ-2 in Table 3.3; ii) our hub latents and FFHQ-1, FFHQ-2; iii) truncated latents ($\psi = 0.7$) and FFHQ-1, FFHQ-2; iv) randomly sampled latents and

FFHQ-1, FFHQ-2. It can be observed that i) both our method and the truncation trick outperform random sampling; ii) our method achieves better FID scores than the truncation trick. Note that we intentionally used a small number of images (*i.e.* 2,000) to compute FID to avoid covering the entire distribution and thus suffer less from the restriction of latent spaces. In comparison with the results in [60] and the bias-free FID_∞ [26] computed with 10K images (Table 3.4), our FID scores of "Truncated (0.7)" images are better than "Random", which is consistent with human perception. Note that our method outperforms Truncated (0.7) in both cases. Examples of StyleGAN2 synthesized images after the truncation trick ($\psi = 0.7$) are shown in Fig. 3.15 Nevertheless, even using a small number of images, FID may still not be a good evaluation metric for our task. Therefore, we resort to the precision and recall metrics [79] that make more sense.

Table 3.4: FID_∞ scores [26] computed with 10K images, which are ineffective as they capture the entire distribution and thus suffer from the restriction of latent spaces. Red: random sampling has the best score, which contradicts human perception as the images sampled with it are of the lowest quality.

Method	Hubs (50)	Truncated (0.7)	Random
$\text{FID}_\infty \downarrow$	**15.398**	15.761	2.923

Precision and Recall [79]. As Table 3.5 shows, our method achieves a high precision comparable to Truncated (0.3) which sacrifices the synthesis diversity (*i.e.* low recall) while retaining a very high recall comparable to Random which includes many low-quality results (*i.e.* low precision). This further justifies the superiority of our method.

Table 3.5: Comparison of precision and recall [79] of StyleGAN2 synthesized images using our method and the truncation trick.

Method	Precision↑	Recall↑
Hubs (50)	<u>0.890</u>	<u>0.324</u>
Truncated (0.3)	**0.892**	0.015
Truncated (0.7)	0.811	0.223
Random	0.720	**0.393**

As Table 3.6 shows, our method outperforms Truncated (0.7) with the BigGAN [18] architecture pretrained on the 1000-class ImageNet ILSVRC 2012 dataset on precision and recall [79], which further justifies the superiority of our method.

Table 3.6: Quantitative results with BigGAN (ImageNet).

Method	Precision↑	Recall↑
Hubs (50)	**0.147**	**0.311**
Truncated (0.7)	0.131	0.264

3.3.7 Impact on Class Balance

We further investigate how our method affects the class balance of unconditional GANs pre-trained on multi-class datasets. As Fig. 3.14 shows, we evaluate the class balance of a StyleGAN2 model pretrained on the CIFAR10 dataset with i) random sampling[8] (*i.e.* Random), ii) truncation trick ($\psi = 0.7$) and iii) our hubness-based sampling method. Specifically, we sample 50,000 images each and use a pretrained CIFAR10 classifier[9] to estimate their class distributions. Note that although a "larger" difference can be observed visually, our method actually preserves the class balance better as it has a smaller Wasserstein distance to the distribution of Random than the truncation trick. In addition, as Table 3.7 shows, our method achieves a better Inception Score [117] that favours balanced and high-confidence classifications, which further justifies the superiority of our method in preserving class balance.

Table 3.7: Evaluation of class balance with Inception Scores (IS) [117] of StyleGAN2 pretrained on the CIFAR10 dataset using our hubness-based sampling ($t = 50$), the truncation trick ($\psi = 0.7$), and the random sampling methods.

Method	Hubs (50)	Truncated (0.7)	Random
IS	**6.212**	6.059	7.080

[8] `https://github.com/POSTECH-CVLab/PyTorch-StudioGAN`

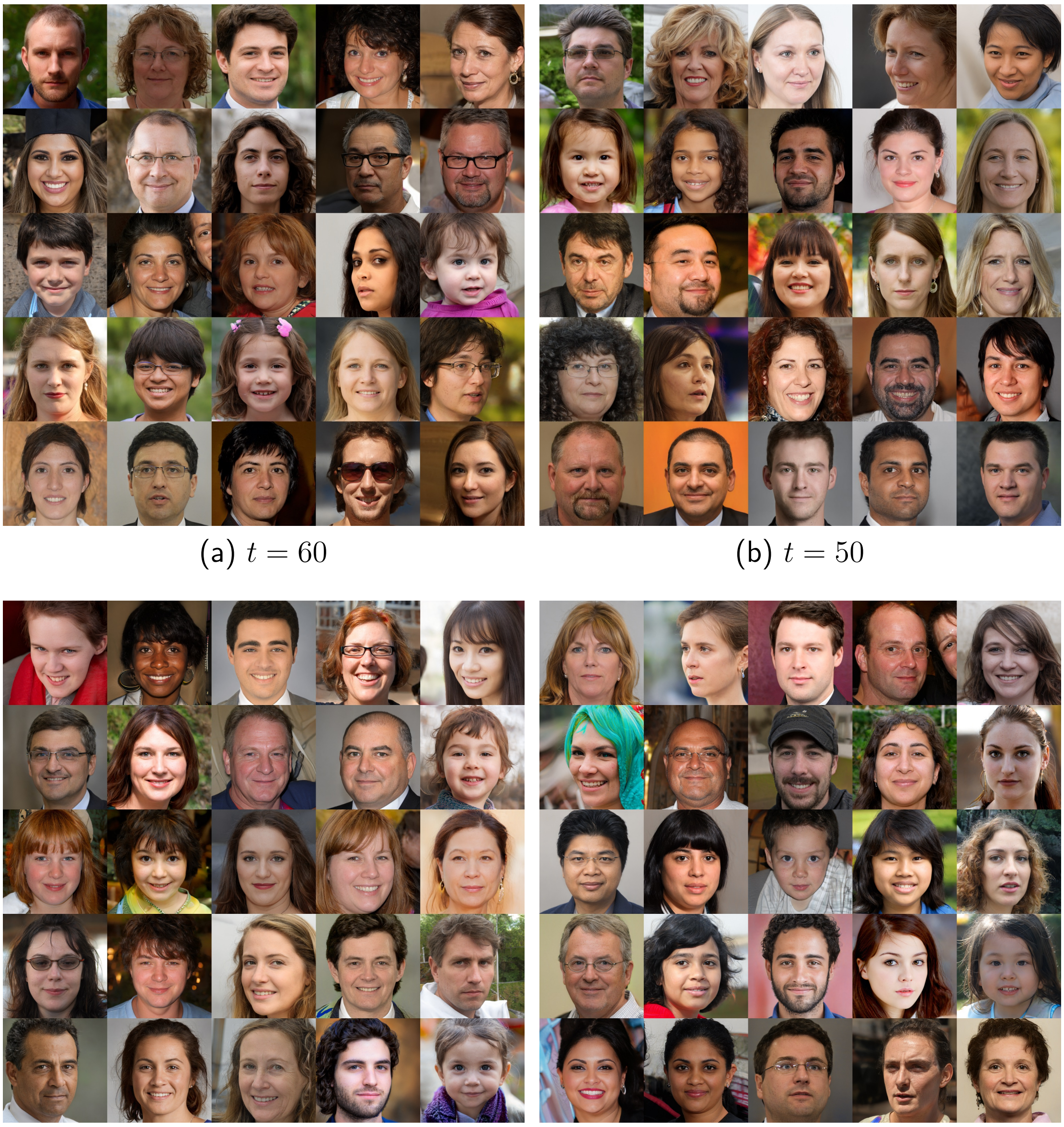
[9] `https://github.com/open-mmlab/mmclassification`, ResNet50

3.4 Limitation and Future Work

Although our method allows for the sampling of high-quality latents, the quality of synthesized images is bounded by the performance of the pre-trained GANs used to synthesize them. Also, we observed that the proposed *hubness priors* may overlook some relatively high-quality images with small *hub* values m (Fig. 3.16). We conjecture that the reason might be that the limited sizes of latent sample sets (*e.g.* $n = 10000, 20000, ...$) cannot capture all *hub* latents. This is partially verified by our experiment on the choice of n. However, it is difficult to test larger n due to the $O(n^2)$ time complexity to compute the hub values m for all points in a latent sample set. We hope to investigate this issue in future work. We also hope to apply our insights on the hubness phenomenon in GAN latent space to improve the training of GANs and make GANs unbiased for all latents. The acceleration of our algorithm is also a very interesting direction for future work.

3.5 Conclusions

In this chapter, we address the quality variance of GAN synthesized images by investigating the sampling of GAN latents. Specifically, we first show that GAN latents are not uniformly distributed in the latent space due to the *hubness* phenomenon of data distributions in high dimensional space. In addition, there exist *hub* latents that are much more likely to be nearest neighbors of others and contribute more to the synthesis of high-quality images. Then, we formulate the above as the *hubness priors* and propose a novel GAN latent sampling algorithm, which allows for efficient and high-quality image synthesis for GANs. Furthermore, we show that the well-known truncation trick is a naive approximation of our method that utilizes the “central clustering effect” of *hub* latents, which not only uncovers the rationale of the truncation trick, but also indicates that our method is superior and more fundamental.



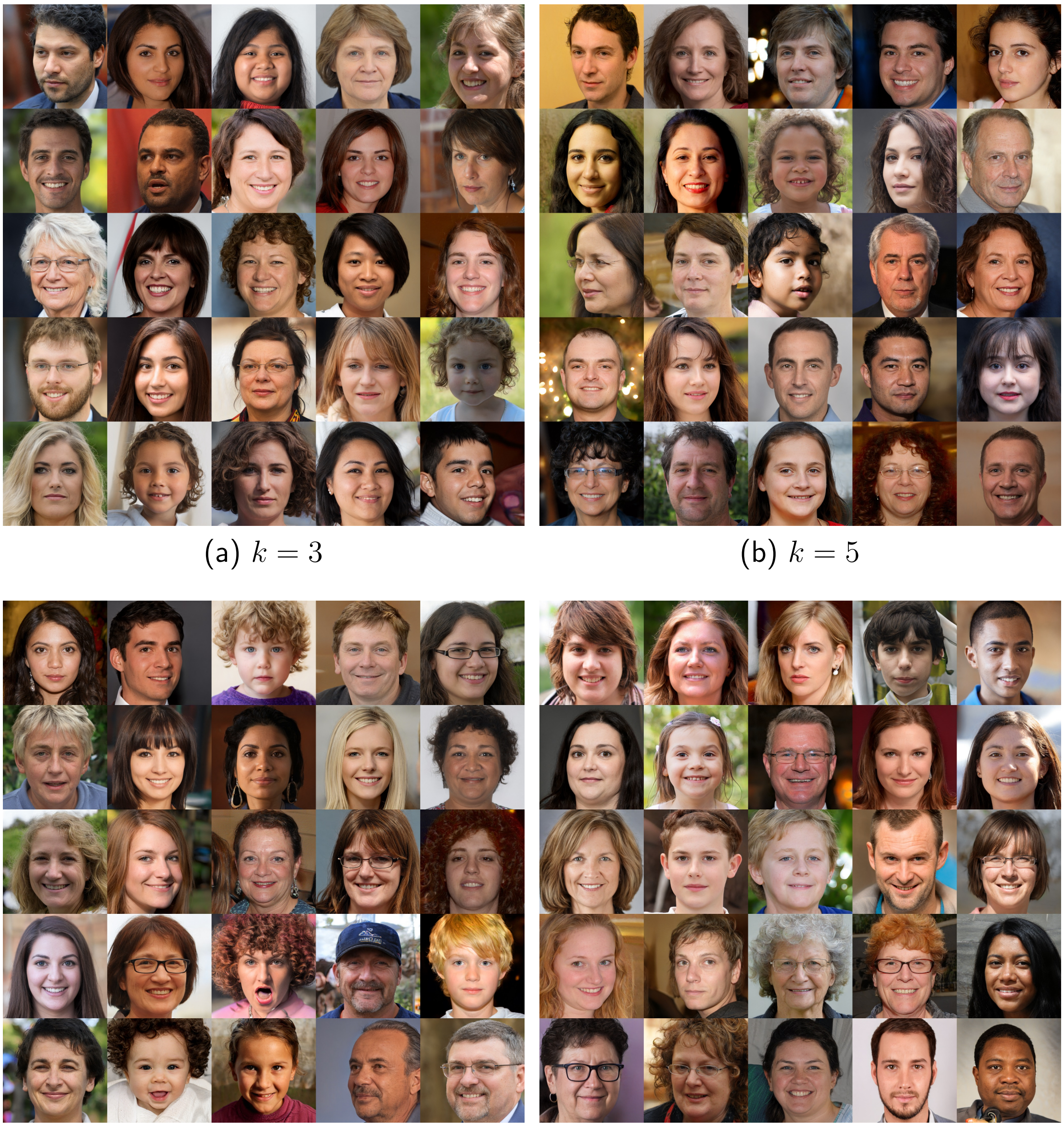
(a) $t = 60$

(b) $t = 50$

(c) $t = 40$

(d) $t = 35$

Figure 3.9: Performance of our method with different choices of threshold $t = 60, 50, 40, 35$. We use $n = 10000$, $k = 5$.



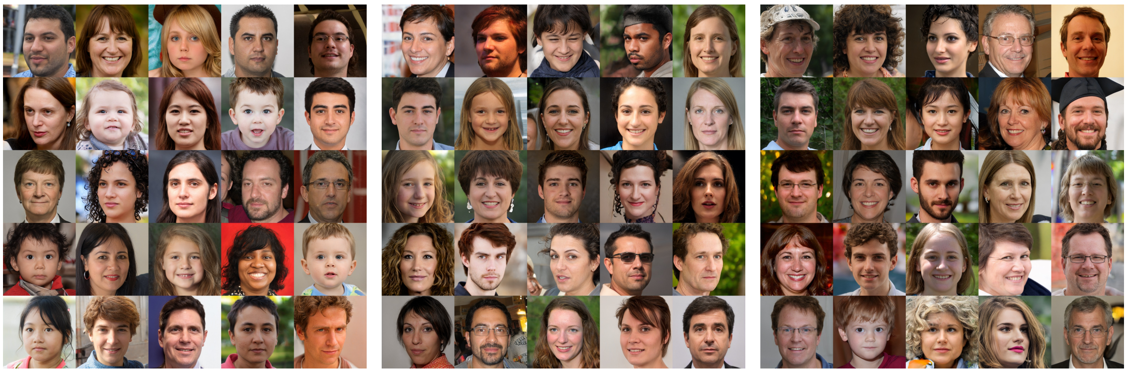
(a) $k = 3$

(b) $k = 5$

(c) $k = 7$

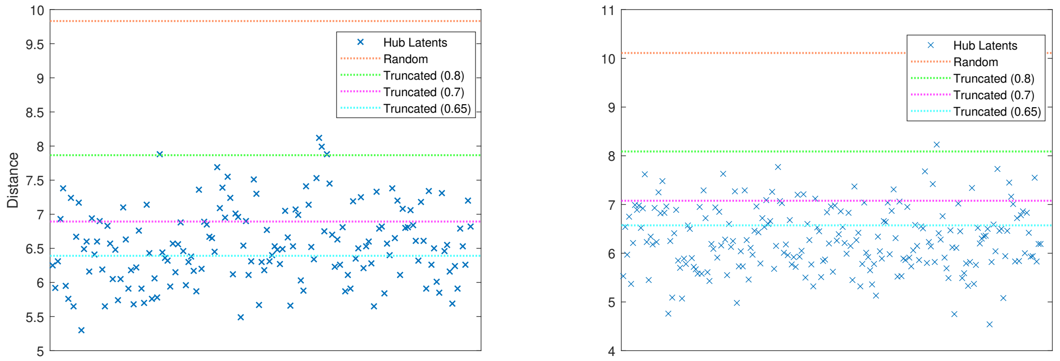
(d) $k = 10$

Figure 3.10: Performance of our method with different choices of hyper-parameter $k = 3, 5, 7, 10$. We use $t = 50$, $n = 10000$.



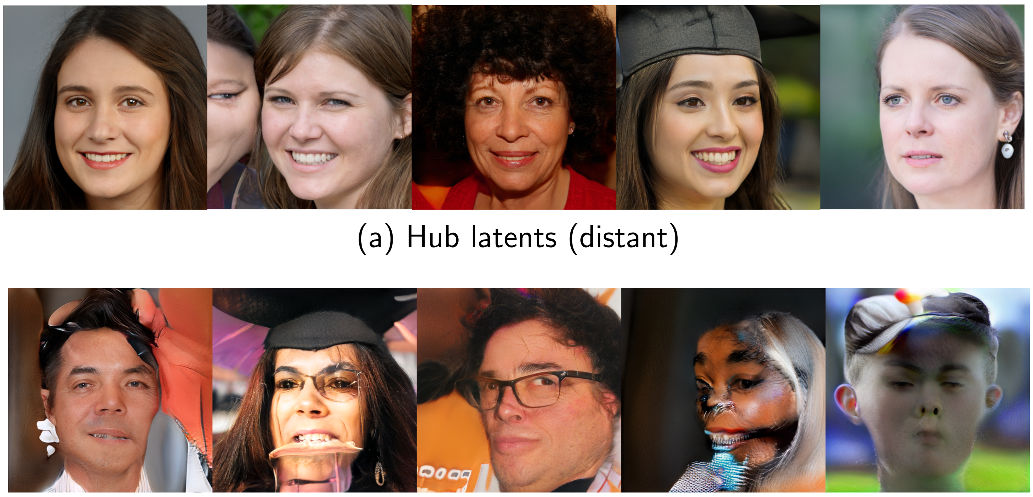
(a) $n = 20000$ (b) $n = 30000$ (c) $n = 40000$

Figure 3.11: Performance of our method with different sizes $n = 20000, 30000, 40000$ of sample set S. We use $k = 5$, $t = 50$.



(a) Dist. to the mean of all sampled latents (b) Dist. to the mean of hub latents

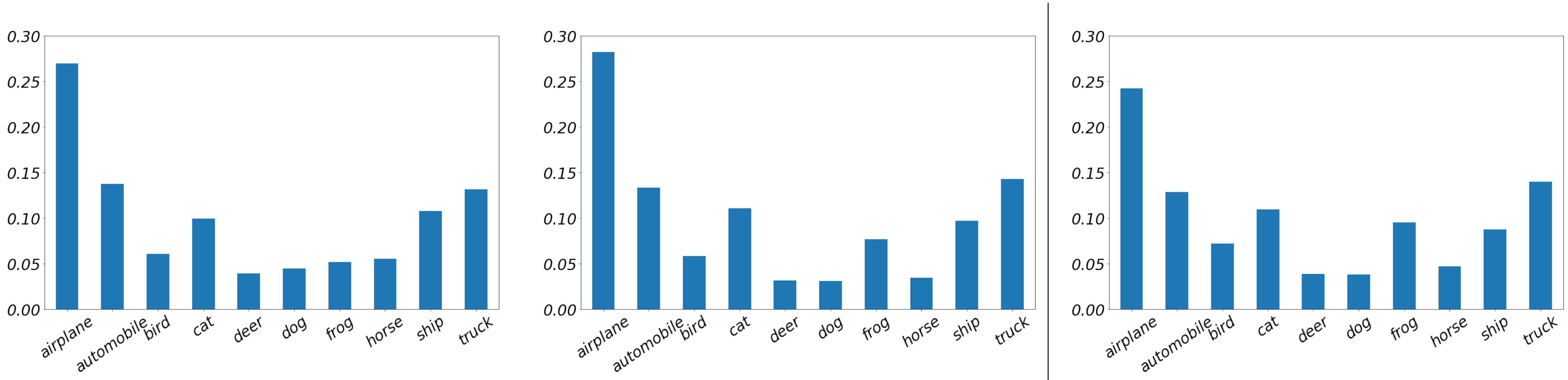
Figure 3.12: The distances of our hub latents to (a) the mean of all sampled latents and (b) the mean of hub latents. Random: the average distance of randomly sampled latents; Truncated (ψ_0): the average distance of latents after truncation trick ($\psi = \psi_0$).



(a) Hub latents (distant)

(b) Truncated latents ($\psi = 0.8$)

Figure 3.13: StyleGAN2 images synthesized from (a) distant hub latents far from their mean; (b) truncated latents ($\psi = 0.8$).



(a) Hubs (50), WD = 0.0094

(b) Truncated (0.7), WD = 0.0102

(c) Random

Figure 3.14: The class distributions of the StyleGAN2 model pretrained on the CIFAR10 dataset with (a) our hubness-based sampling ($t = 50$), (b) the truncation trick ($\psi = 0.7$), and (c) the random sampling methods. WD: Wasserstein distance.

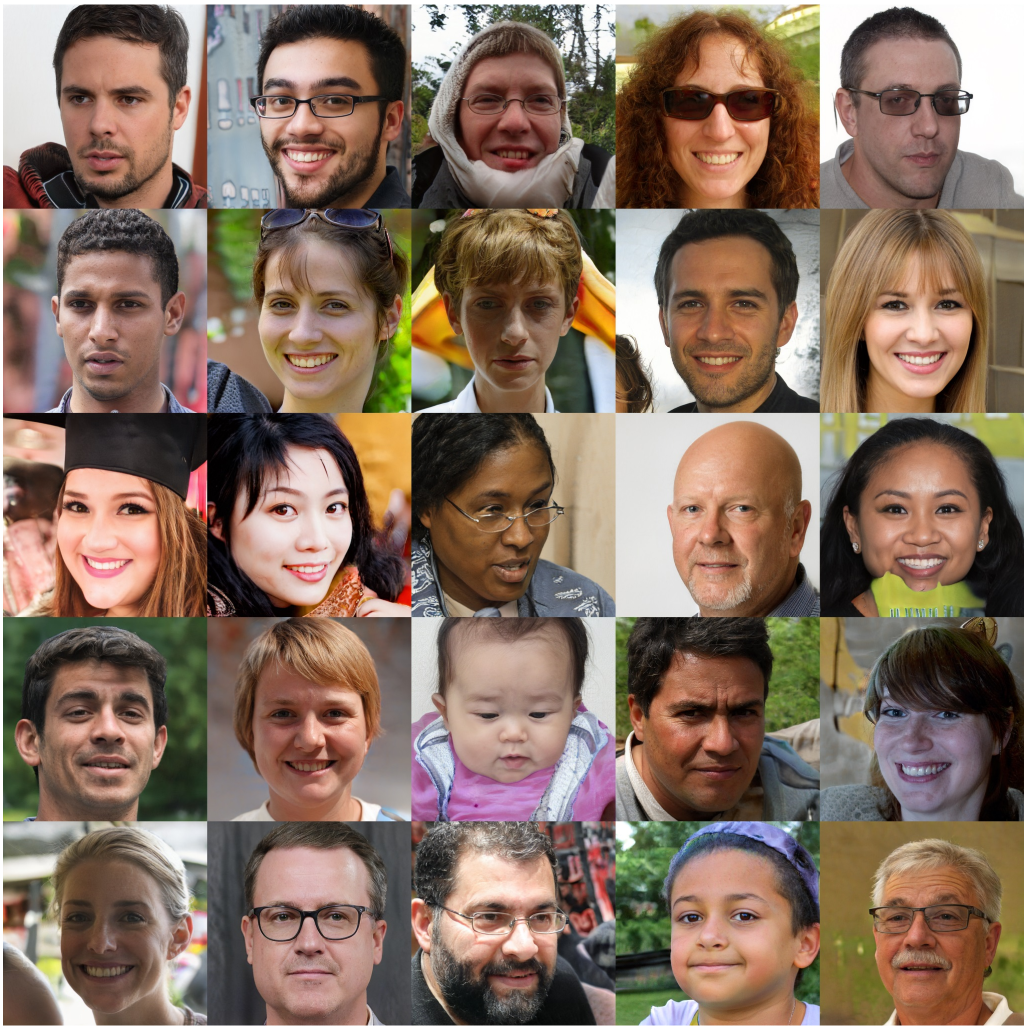


Figure 3.15: Examples of StyleGAN2 synthesized images after the truncation trick ($\psi = 0.7$).

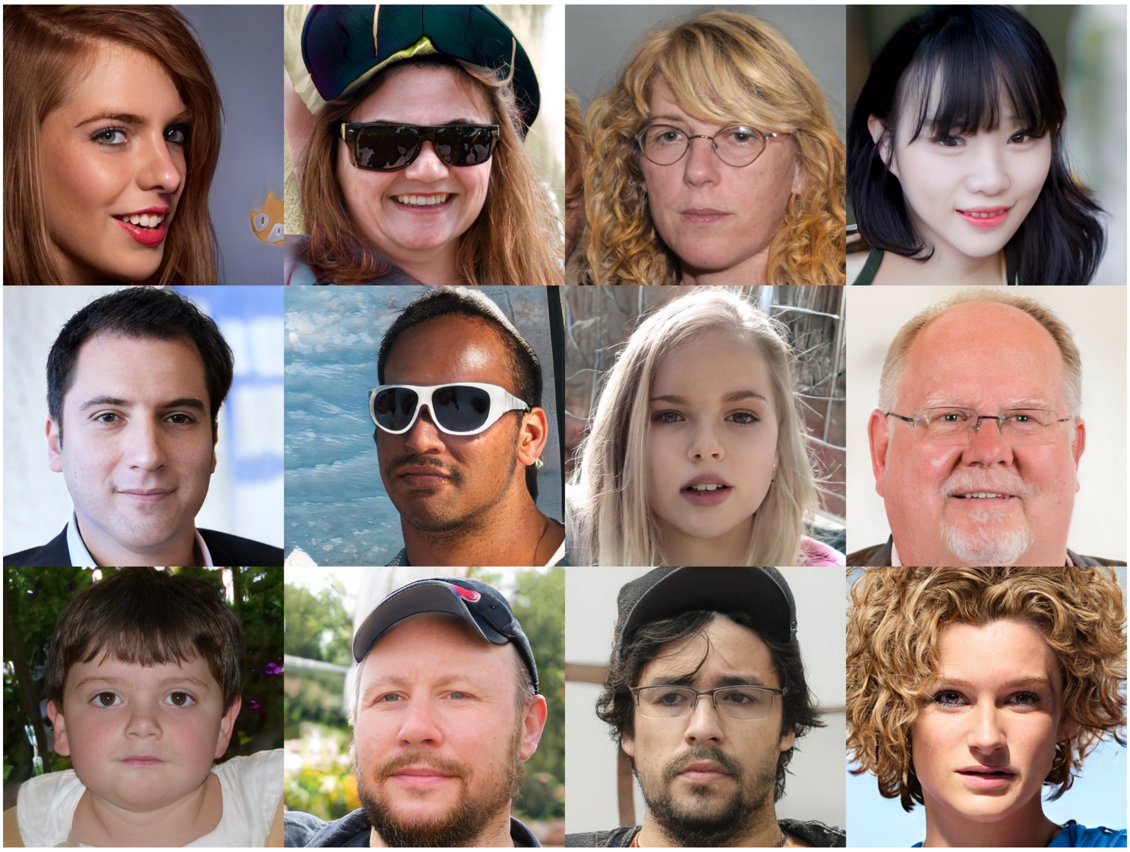


Figure 3.16: Relatively high-quality StyleGAN2 [67] synthesized images with small hub values m. However, there are still small artifacts in these images (*e.g.* background and facial details).

Chapter 4

Efficient Precision and Recall for Assessing Generative Models with Hubness Sampling

Despite impressive results, deep generative models require massive datasets for training. As dataset size increases, effective evaluation metrics like precision and recall (P&R) become computationally infeasible on commodity hardware. In this chapter, we address this challenge by proposing efficient P&R (eP&R) metrics that give almost identical results as the original P&R but with much lower computational costs. Specifically, we identify two redundancies in the original P&R: i) redundancy in ratio computation and ii) redundancy in manifold inside/outside identification. We find both can be effectively removed via hubness-aware sampling, which extracts representative elements from synthetic/real image samples based on their hubness values, *i.e.*, the number of times a sample becomes a k-nearest neighbor to others in the feature space. Thanks to the insensitivity of hubness-aware sampling to exact k-nearest neighbor (k-NN) results, we further improve the efficiency of our eP&R metrics by using approximate k-NN methods. Extensive experiments show that our eP&R matches the original P&R but is far more efficient in time and space.

4.1 Introduction

Deep generative models have achieved great success by combining deep learning with generative modeling. However, they have also inherited the data-hungry nature of deep learning, requiring massive datasets for training. For instance, the FFHQ dataset used to train StyleGAN contains 70 thousand images [66], while the Latent Diffusion model leveraged LAION-400M's 400 million text-image pairs [112]. Stable Diffusion pushed this even further, training its models on LAION-5B's 5 billion pairs [121]. Despite their impressive results, the massive scale of datasets used to train modern deep generative models presents challenges for evaluation. As dataset size increases, some of the most effective evaluation metrics [52, 78, 116, 117, 124], which compare generated and real image distributions, may become computationally infeasible for commodity GPUs and ordinary research institutions. To continue advancing the state of the art (SOTA), developing more efficient evaluation metrics becomes critical.

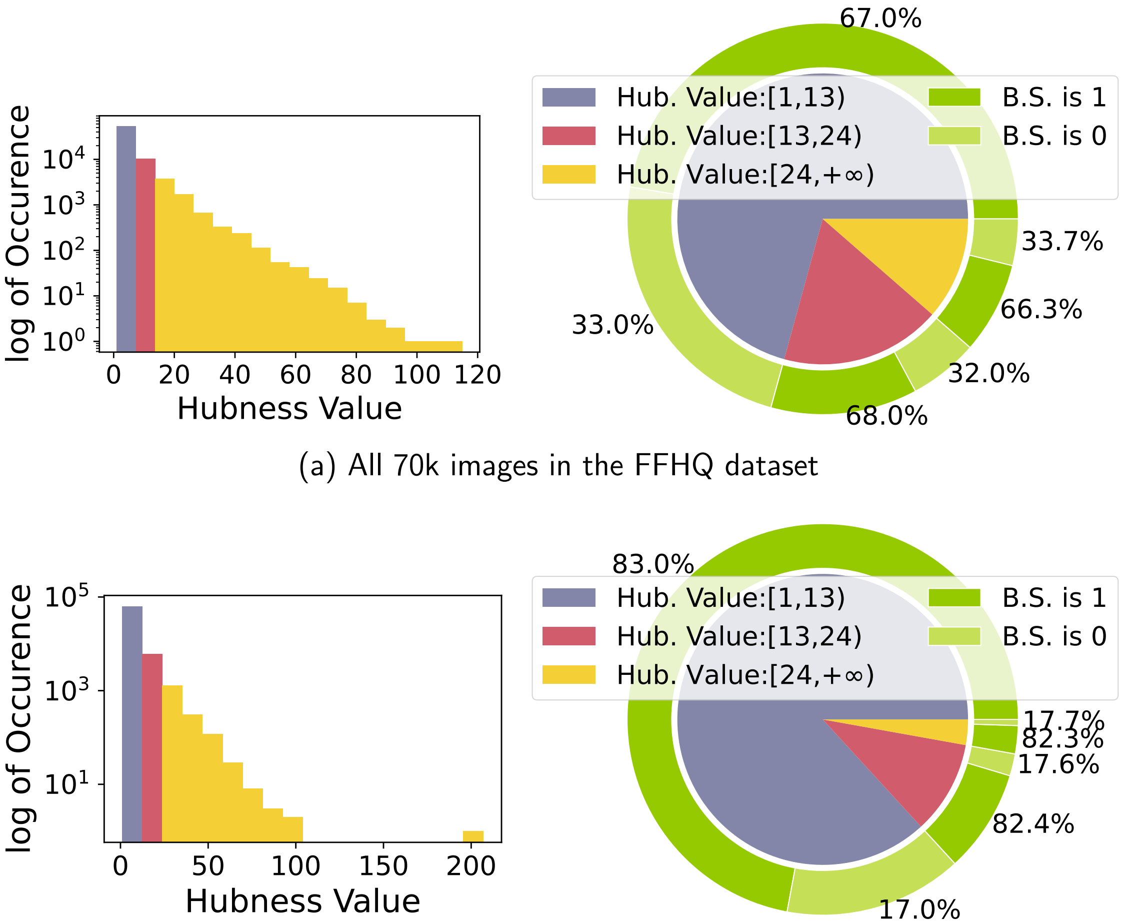
Among the most effective evaluation metrics, Fréchet Inception Distance (FID) [52] and Inception Score (IS) [117] are relatively computationally efficient. Let n be the number of samples, they have linear time and space complexity of $O(n)$, as they rely on simple statistics of extracted features. Specifically, the feature extraction takes $O(n)$ time and space while the statistics (*e.g.*, mean) computation also takes linear time $O(n)$ for a given feature extractor like Inception v3 [129] with fixed feature dimensions. However, both FID and IS are single scores that cannot distinguish between different failure modes. Addressing this issue, the precision and recall (P&R) metrics [78, 116, 124] were employed. Intuitively, precision measures the *quality* of synthesized images while recall measures their *diversity*. Although effective, the SOTA version of P&R [78] requires costly pairwise distance calculations (*e.g.*, in k-nearest neighbor algorithm) between extracted features of samples and sorting, consuming $O(n^2 \text{log} n)$ time and space, thus becoming computationally infeasible when evaluating deep generative models trained on large-scale datasets.

In this work, we address the high computational costs of precision and recall (P&R) metrics with a novel solution based on *hubness-aware sampling*. Specifically, we have identified two important types of *redundancies* in the computation of P&R: i) redundancy in the P&R ratio computation and ii) redundancy in identifying whether a sample is within or outside of a manifold (*e.g.*, synthetic or real image manifold). Interestingly, we find that both these redundancies can be effectively removed by hubness-aware sampling. In a nutshell, hubness-aware sampling extracts a small number of m representative elements from the real/synthetic samples based on their *hubness values*, defined as the number of times a sample becomes a k-nearest neighbor (k-NN) to others in the feature space [85, 107]. We denote such representative elements as "hubs", which have higher hubness values than their peers. We conjecture that the validity of our approach comes from the fact that hubness values are effective importance identifiers for samples with respect to the k-NN results on which P&R [78] relies. In addition, utilizing the fact that the identification of hubness points relies on their relatively higher hubness values rather than exact k-NN results, we further improve the efficiency of our eP&R metrics using approximate k-NN methods, a brief introduction of which can be found in the Sec. 4.4. Extensive experimental results demonstrate that the P&R calculated using such representative elements is almost identical to the original P&R, but consumes much less time and space. Our contributions include:

- We propose *efficient precision and recall* (eP&R) metrics for assessing generative models, which give almost identical results as the original P&R [78] but consume much less time and space. Theoretically, our eP&R run in $O(mn \log n)$ time and consume $O(mn)$ space (m is the number of of hubs samples and $m < n$), which are much more efficient than the original P&R metrics that run in $O(n^2 \log n)$ time and consumes $O(n^2)$ space.

- We identify two important types of redundancies in the original P&R metrics and uncover that both of them can be effectively removed by hubness-aware sampling [85, 107]. In addition, the insensitivity of hubness-aware sampling to exact k-nearest neighbor (k-NN) results allows for further efficiency improvement by using approximate k-NN methods.

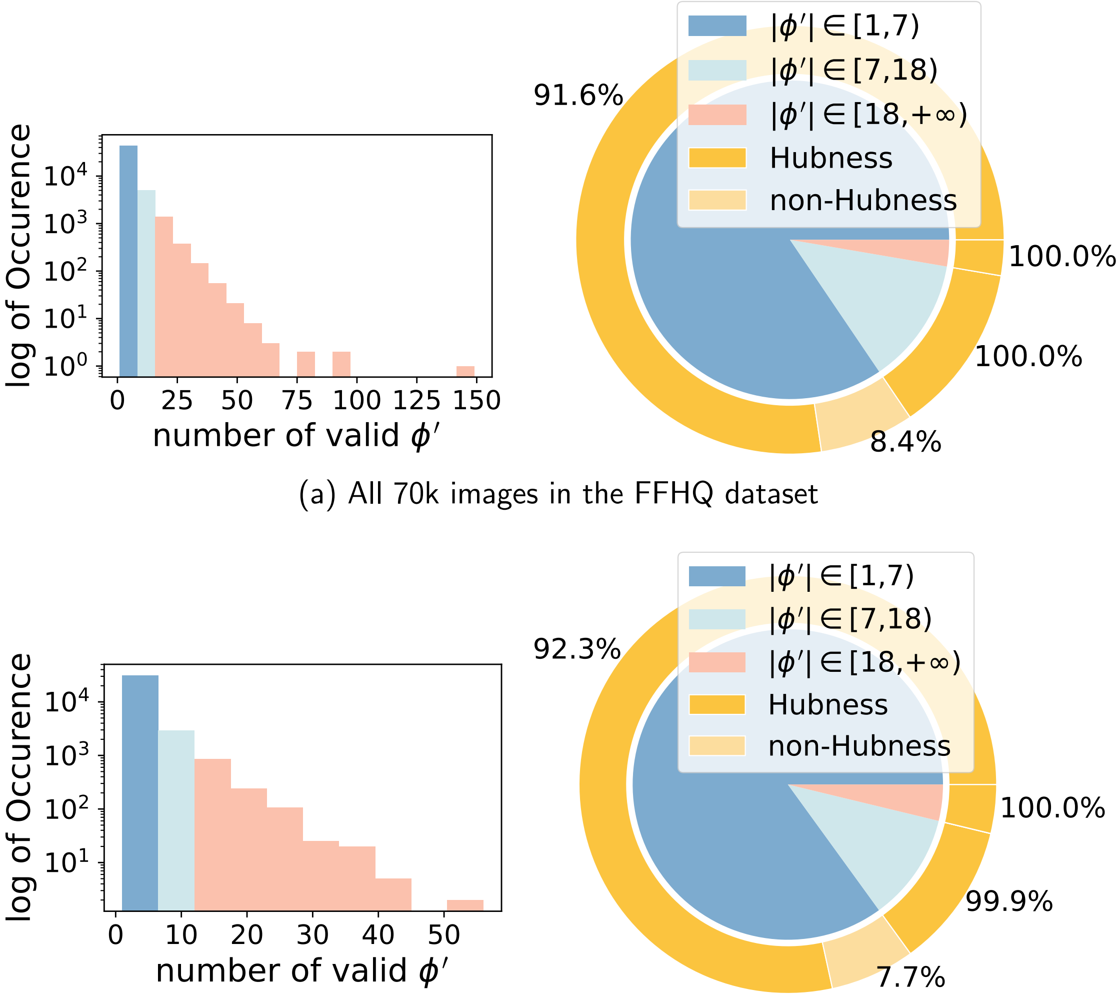
- Extensive experimental results demonstrate the effectiveness of eP&R metrics.



(a) All 70k images in the FFHQ dataset

(b) 70k images generated by StyleGAN2

Figure 4.1: Samples with similar hubness values are effective representative samples in terms of P&R ratio calculation. (a) Left: Histogram of sample occurrences (log scale) *vs.* hubness value (FFHQ). The samples are grouped into different colors based on similar hubness values. Right: Pie chart showing that all three groups share similar ratios of samples identified as 1 vs. 0 (green *vs.* light green) using Eq. 4.3 for recall calculation. (b) The same experiment as (a) but on StyleGAN-generated samples for precision calculation. Following [78], we use VGG-16 as a feature extractor and StyleGAN2 trained on the FFHQ dataset as the generative model to be assessed. Please see Section. 4.6.9 for the validation of insensitivity of the choice of group split points. Hub.: Hubness; B.S.: binary score.



(a) All 70k images in the FFHQ dataset

(b) 70k images generated by StyleGAN2

Figure 4.2: Most samples ϕ with $f(\phi, \mathbf{\Phi}) = 1$ (Eq. 4.3) are included in the k-NN hypersphere of at least one hubs sample ($t = 3$) of the other distribution. (a) Left: Histogram of sample occurrences (log scale) *vs.* the times a sample is included in the k-NN hypersphere of a sample of the other distribution, *i.e.*, valid ϕ' (FFHQ). Please see Sec. 4.3 for an intuitive illustration. The samples are grouped into different colors based on similar numbers of valid ϕ'. Right: Pie chart showing the ratio of samples within the k-NN hypersphere of *hubness* vs. *non-hubness* samples from the other distribution, to the total number of samples ϕ with $f(\phi, \mathbf{\Phi}) = 1$ in each group. Hubness: points with hub values above a threshold $t \geq 3$; Non-hubness: $t < 3$. (b) The same experiment as (a) but on StyleGAN-generated samples. Following [78], we use VGG-16 as a feature extractor and StyleGAN2 trained on the FFHQ dataset as the generative model to be assessed. Please see Section. 4.6.9 for the validation of insensitivity of the choice of group split points.

4.2 Preliminaries

As proposed by [78], the precision and recall (P&R) metrics for assessing generative models are defined as:

$$\text{precision}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_g|} \sum_{\phi_g \in \mathbf{\Phi}_g} f(\phi_g, \mathbf{\Phi}_r), \tag{4.1}$$

$$\text{recall}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_r|} \sum_{\phi_r \in \mathbf{\Phi}_r} f(\phi_r, \mathbf{\Phi}_g) \tag{4.2}$$

where $\mathbf{\Phi_g}$ and $\mathbf{\Phi_r}$ are the sets of feature vectors corresponding to the generated and real image samples, respectively; $|\mathbf{\Phi}|$ denotes the number of samples in set $\mathbf{\Phi}$ and $|\mathbf{\Phi}_g| = |\mathbf{\Phi}_r|$; $f(\phi, \mathbf{\Phi})$ is a binary function determining whether a sample ϕ lies on a manifold represented by $\mathbf{\Phi}$:

$$f(\phi, \mathbf{\Phi}) = \begin{cases} 1, \text{ if } \|\phi - \phi'\|_2 \leq \|\phi' - \text{NN}_k(\phi', \mathbf{\Phi})\|_2 & \text{for } \textbf{at least one } \phi' \in \mathbf{\Phi} \\ 0, \text{ otherwise,} \end{cases} \tag{4.3}$$

where $\text{NN}_k(\phi', \mathbf{\Phi})$ denotes the kth nearest neighbour of ϕ' in $\mathbf{\Phi}$. Intuitively, their precision and recall metrics estimate the generative and real image manifolds with a collection of hyperspheres, respectively, with each feature vector sample as the center and the distance between it and its kth nearest neighbor as the radius. A sample ϕ is determined to lie on a manifold if it lies within the hyperspheres of that manifold and vice versa.

4.3 Illustration figure and relevant discussions for valid ϕ'

As Fig. 4.3 shows, by "the times a sample is included in the k-NN hypersphere of a sample of the other distribution, *i.e.*, valid ϕ'", we count the number of times ϕ (yellow cube) is within the k-NN hypersphere of $\phi' \in \Phi$ (red rhombuses).

Table 4.1: Approximation errors compared to the original Precision and Recall (P&R) metrics. B.L.: the original P&R metrics as the baseline [78]. eP&R: our efficient P&R metrics. Error(%): relative error $\epsilon = \frac{|x-\hat{x}|}{|x|}$, where x is the B.L. result and $\hat{x}$ is our eP&R result.

(a) Approximation errors of eP&R computed using StyleGAN2 trained on different datasets.

	FFHQ		LSUN-Car		LSUN-Church		LSUN-Cat		LSUN-Horse	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
eP&R	0.719±0.002	0.501±0.002	0.732±0.001	0.422±0.002	0.608±0.002	0.392±0.003	0.758±0.001	0.408±0.003	0.693±0.001	0.416±0.003
B.L.	0.716±0.001	0.493±0.001	0.725±0.001	0.426±0.001	0.592±0.001	0.389±0.002	0.766±0.001	0.401±0.002	0.682±0.001	0.413±0.002
Error	0.4%	1.6%	0.9%	0.9%	1.9%	0.7%	1.0%	1.7%	1.5%	0.7%

(b) Approximation errors of eP&R calculated using different generative models and the FFHQ dataset.

	StyleGAN3		Projected-GAN		VQ-VAE-2		Latent Diffusion	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
eP&R	0.684±0.002	0.559±0.001	0.707±0.001	0.462±0.001	0.719±0.001	0.163±0.004	0.704±0.001	0.471±0.002
B.L.	0.680±0.001	0.553±0.001	0.698±0.001	0.460±0.001	0.716±0.001	0.162±0.002	0.711±0.001	0.460±0.001
Error	0.6%	1.0%	1.2%	0.4%	0.4%	0.6%	1.0%	2.3%

Table 4.2: Time and space consumption of our eP&R metrics compared to the original P&R metrics using StyleGAN2 trained on the FFHQ and LSUN-Church datasets respectively. B.L.: the original P&R metrics as the baseline [78]. eP&R: our efficient P&R metrics. Time (S): serial implementation. Time (P): parallel implementation using CUDA. The profiling items are in one-to-one correspondence with the lists in Sec. 4.5.3 using Python time() and memory-profiler `https://github.com/pythonprofilers/memory_profiler/tree/master`. Please see section. 4.6.8 for additional results on the large LSUN-Horse dataset with 1.5m images, which further demonstrates the effectiveness of our eP&R metrics.

(a) FFHQ (70k images).

Profiling	B.L.			Profiling	eP&R		
	Time (S)	Time (P)	Memory		Time (S)	Time (P)	Memory
DMs ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	160s	66s	15.84 GB	Subspace ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	4s	3s	3.01 GB
				A. hubs ($\mathbf{\Phi}_r^{hub}$, $\mathbf{\Phi}_g^{hub}$)	2s	1.2s	–
				eDMs	72s	32s	11.23 GB
Sorting	104s	22s	–	eSorting	50s	12s	–
Radii	2.2s	2.2s	0.58 GB	Radii	1.7s	1.7s	0.30 GB
DM ($\mathbf{\Phi}_r \leftrightarrow \mathbf{\Phi}_g$)	85s	34s	19.24 GB	eDM ($\mathbf{\Phi}_r^{hub} \leftrightarrow \mathbf{\Phi}_g^{hub}$)	18s	9s	8.74 GB
P&R	48s	28s	–	eP&R	11s	6s	–
Total/Peak	399s	144s	19.90 GB	Total/Peak	165s	**75s**	**14.24 GB**

(b) LSUN-Church (120K images).

Profiling	B.L.			Profiling	eP&R		
	Time (S)	Time (P)	Memory		Time (S)	Time (P)	Memory
DMs ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	211s	110s	35.24 GB	Subspace ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	9s	7s	5.40 GB
				A. hubs ($\mathbf{\Phi}_r^{hub}$, $\mathbf{\Phi}_g^{hub}$)	3.3s	3.3s	–
				eDMs	107s	48s	18.83 GB
Sorting	164s	42s	–	eSorting	95s	28s	–
Radii	5s	5s	0.81 GB	Radii	4s	4s	0.51 GB
DM ($\mathbf{\Phi}_r \leftrightarrow \mathbf{\Phi}_g$)	143s	61s	37.04 GB	eDM ($\mathbf{\Phi}_r^{hub} \leftrightarrow \mathbf{\Phi}_g^{hub}$)	36s	15s	9.54 GB
P&R	90s	40s	–	eP&R	17s	8s	–
Total/Peak	613s	238s	37.24 GB	Total/Peak	269s	**113s**	**25.01 GB**

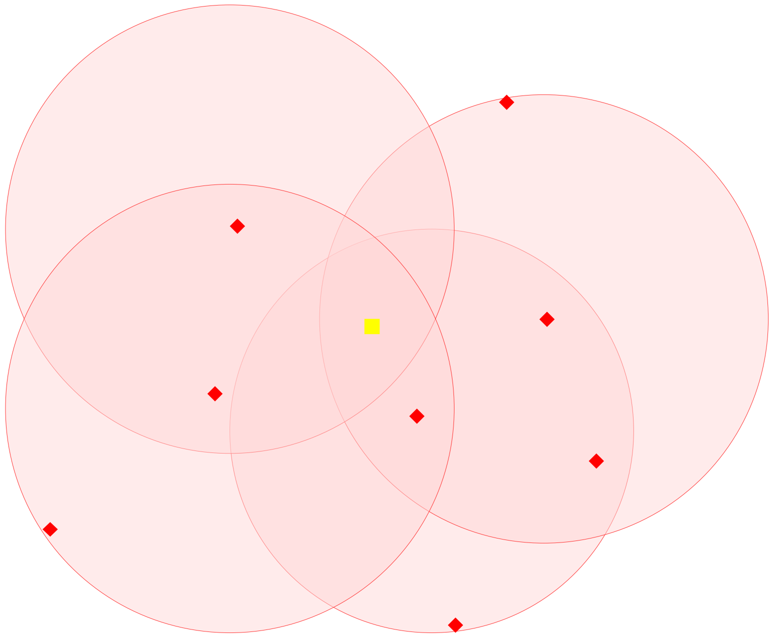


Figure 4.3: Illustration of valid ϕ'. ϕ is represented by a yellow cube and $\phi' \in \Phi$ set are represented by red rhombuses.

4.4 A Brief Introduction to Approximate k-NN algorithm

The k-nearest neighbors (k-NN) algorithm is a popular machine learning method for classification and regression. Given a new data point, it finds the k closest training examples based on a distance metric like Euclidean distance. A major limitation of k-NN is that it requires computing the distance between the new point and all points in the training set, which can be slow for large datasets.

Approximate k-NN algorithms are techniques that try to speed up neighbor search by sacrificing some accuracy. The key idea is to avoid exhaustively calculating distances to all points. Some common approaches include:

- Tree-based data structures like kd-trees [16] that allow efficient searching of nearest points without checking all data.

- Hashing techniques [13, 91] that map similar points to the same buckets, narrowing the search.

- Dimensionality reduction methods like random projections [146] that can compress data while preserving relative distances.

- Graph-based algorithms [39, 50, 83, 92] that connect neighboring points then traverse the graph instead of computing all distances.
- Sampling/filtering [11, 59, 101, 161] methods that find candidates in subsections of data.

The tradeoff is between accuracy and speed. Approximate methods may miss some true nearest neighbors, but can query large datasets much more efficiently. Performance gains allow k-NN to scale better to big data. Appropriate techniques depend on factors like data size, dimension, and desired accuracy. We refer interested audiences to [8, 83, 123, 144] for more details.

4.4.1 The accuracy to get the hubness vectors

The table 4.3 compares the performance of various Approximate Nearest Neighbor (ANN) algorithms in retrieving hubness points, measured by Accuracy (%) and search Time (s). A key distinction across the columns is the hardware acceleration capability: the methods on the left side (IVF [59, 61], IVF-PQ [59, 61], IVF-SQ) are typically designed with GPU acceleration support, allowing for highly parallel and fast computations. In contrast, the algorithms on the right side (LSH [13, 91], HNSW [92], ScaNN [9]), while often optimized for speed on CPUs, are not universally or easily accelerated by GPUs in standard configurations (their primary implementations are CPU-based). The table serves as powerful evidence that ANN algorithms provide a highly effective solution, enabling near-production-quality accuracy (mostly $\geq 90\%$) at a fraction of the computational cost and time required by exhaustive search methods to get the hubness points.

4.5 Efficient Precision and Recall

Although effective, Eqs. 4.2 and 4.3 are computationally expensive due to the calculation of pairwise distances between samples and the sorting required by k-NN, which grows quasi-quadratically with the number of samples. This prevents them

Table 4.3: The accuracy of the different ANN to get the hubness points. The methods on the left side (IVF [59, 61], IVF-PQ [59, 61], IVF-SQ) are typically designed with GPU. In contrast, the algorithms on the right side (LSH [13, 91], HNSW [92], ScaNN [9]), while often optimized for speed on CPUs, are not universally or easily accelerated by GPU.

	IVF	IVF-PQ	IVF-SQ	LSH	HNSW	ScaNN
Accuracy(%)	99.731	90.061	71.903	94.644	98.642	98.691
Time(s)	1.085	1.062	7.323	4.329	3.075	18.277

Table 4.4: Ablation study. Alg. Variations: variants of our metrics. Time (P): parallel implementation using CUDA. P&R: original P&R metrics [78]. Ob. 4.5.1: we replace the $\mathbf{\Phi}_g$ in precision calculation and $\mathbf{\Phi}_r$ in recall calculation with their hubs versions $\mathbf{\Phi}_g^{hub}$ and $\mathbf{\Phi}_r^{hub}$ respectively (Eq. 4.2). Ob. 4.5.2: we replace the $\mathbf{\Phi}_r$ in precision calculation and $\mathbf{\Phi}_g$ in recall calculation with their hubs versions $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_g^{hub}$ respectively (Eq. 4.2). eP&R: our efficient P&R metrics, which uses Ob. 4.5.1, 4.5.2 and "Efficient Hubs Sample Identification" (approximate k-NN) together. *: when used alone, (2)(3) cannot save time and space as they still require the full distance matrices.

Alg. Variations	Precision	Recall	Time (P)	Memory
(1) P&R (Original)	0.716±0.001	0.493±0.001	144s	19.90 GB
(2) P&R + Ob. 4.5.1	0.715±0.005	0.497±0.004	138s*	19.32* GB
(3) P&R + Ob. 4.5.2	0.708±0.005	0.501±0.005	140s*	19.31* GB
(4) P&R + Ob. 4.5.1, 4.5.2	0.719±0.002	0.494±0.001	104s	15.84 GB
(5) eP&R (Ours)	0.719±0.002	0.501±0.001	**75s**	**14.21 GB**

from being computed on large datasets with commodity GPUs and hampers the progress of the field. To improve the computational efficiency of precision and recall (P&R) metrics, we identify two important types of *redundancies* in Eqs. 4.2 and 4.3 (Sec. 4.5.1) and propose to address them using *hubness-aware sampling*, whose insensitivity to exact k-NN results allows for further efficiency improvement (Sec. 4.5.2). We also conduct a computational complexity analysis (Sec. 4.5.3) to demonstrate the high computational efficiency of our method.

4.5.1 Redundancies in Precision and Recall Calculations

As mentioned above, we have identified two important types of *redundancies* in P&R calculations: i) redundancy in the P&R ratio computation and ii) redundancy in identifying whether a sample is within or outside of a manifold (*e.g.*, synthetic or real image manifold) as follows:

Observation 4.5.1. [Redundancy in Ratio Estimation] As Eq. 4.2 shows, the

Table 4.5: Choice of k for the k-NN algorithm used in our eP&R metric. B.L.: the original P&R metrics as the baseline [78]. eP&R: our efficient P&R metrics. Time (S): serial implementation. Time (P): parallel implementation using CUDA.

k	B.L.				eP&R			
	Precision	Recall	Time (P)	Memory	Precision	Recall	Time (P)	Memory
3	0.716±0.001	0.493±0.001	144s	19.90 GB	0.719±0.001	0.494±0.001	75s	14.21 GB
5	0.814±0.001	0.614±0.001	150s	19.90 GB	0.813±0.002	0.615±0.002	73s	13.71 GB
7	0.865±0.001	0.683±0.001	144s	19.91 GB	0.868±0.004	0.689±0.002	73s	13.51 GB
9	0.893±0.001	0.730±0.001	147s	19.91 GB	0.899±0.006	0.737±0.002	72s	13.39 GB
10	0.907±0.001	0.758±0.001	147s	19.91 GB	0.915±0.006	0.767±0.002	71s	13.39 GB

P&R metrics are essentially ratios of the number of samples in a set $\mathbf{\Phi}$ that lie on a given manifold to the number of all samples in $\mathbf{\Phi}$. Thus, we can obtain similar P&R ratios by using *representative samples* of $\mathbf{\Phi}$ with the rest as redundant.

Observation 4.5.2. [Redundancy in Inside/Outside Manifold Identification] As shown in Eq. 4.3, $f(\phi, \mathbf{\Phi})$ is 1 as long as ϕ is within the k-NN hypersphere of *at least one* sample $\phi' \in \mathbf{\Phi}$. This means that we only need to find one valid ϕ' for each ϕ and all the other ϕ's are redundant.

4.5.2 Redundancy Reduction using Hubness-aware Sampling

Interestingly, we find hubness-aware sampling to be an effective solution for both redundancies. Specifically, for Observation 4.5.1, we find that samples with similar hubness values are effective representative samples of set $\mathbf{\Phi}$ in terms of P&R ratios as they share similar ratios of samples identified as 1 *vs.* 0 by Eq. 4.3 (Fig. 4.1), indicating that we can use a small number of hubs samples to approximate P&R; for Observation 4.5.2, we find that most ϕ with $f(\phi, \mathbf{\Phi}) = 1$ (Eq. 4.3) are included in the k-NN hypersphere of at least one ϕ' with high hubness values, *i.e.*, hubs samples (Fig. 4.2), indicating that we can obtain similar outputs of Eq. 4.3 using a small number of hubs samples. Thus, our efficient P&R metrics can be defined as:

$$\text{precision}^{hub}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_g^{hub}|} \sum_{\phi_g^{hub} \in \mathbf{\Phi}_g^{hub}} f(\phi_g^{hub}, \mathbf{\Phi}_r^{hub}) \tag{4.4}$$

$$\text{recall}^{hub}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_r^{hub}|} \sum_{\phi_r^{hub} \in \mathbf{\Phi}_r^{hub}} f(\phi_r^{hub}, \mathbf{\Phi}_g^{hub}) \tag{4.5}$$

where $\mathbf{\Phi}_g^{hub}$ and $\mathbf{\Phi}_r^{hub}$ are the sets of feature vectors with hubness values $m > t$ corresponding to the generated and real image samples, respectively; t is a threshold hyper-parameter.

Efficient Hubs Sample Identification. Despite their effectiveness, the identification of hub samples is also based on the $O(n^2)$ k-NN algorithm which is expensive in both time and space. Fortunately, such identification is insensitive to exact k-NN results as it only relies on a rough threshold t of the hubness values. Thus, we can use an approximate k-NN algorithm for the identification of hub samples that further improves the efficiency of our metrics.

4.5.3 Computational Complexity Analysis

To provide a clear demonstration of the computational efficiency of our metrics, we conduct a computational complexity analysis as follows. Given two sets $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$ ($|\mathbf{\Phi}_r| = |\mathbf{\Phi}_g| = n$), the calculation of the original P&R [78] can be divided into five stages:

1. [**Distance Matrices of $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$**] Calculating pairwise distances for samples in $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$ respectively, which consumes $O(n^2)$ time and space for each set.

2. [**Sorting**] Sorting the distance matrices as required by the k-NN algorithm, which consumes $O(n^2 \log n)$ time and no extra space.

3. [**Radii**] Recording the distance from each sample to its kth nearest neighbour as the radius of its hypersphere, taking $O(n)$ time and space.

4. [**Distance Matrix between $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$**] Calculating pairwise distances between samples of $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$, which consumes $O(n^2)$ time and space.

5. [**P&R**] Calculating P&R ratios, taking $O(n^2)$ time and no extra space for each metric.

In contrast, the calculation of our efficient P&R metrics can be divided into seven stages:

1. [**Subspace Construction for $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$**] Constructing subspaces of samples for $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$ as required by the approximate k-NN algorithm IVF/PQ [59, 61] and HNSW [92], taking $O(\log n)$ time and $O(n)$ space for each set.

2. [**Approx. Hubs Identification for $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$**] Computing the approximate hubness value for each sample in $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_g$ using the approximate k-NN algorithm and extracting hubs set $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_g^{hub}$ with m_r and m_g ($m_r < n$, $m_g < n$) hubs samples respectively using a user-specified threshold t, taking $O(m_r)$, $O(m_g)$ time and space for each set, respectively.

3. [**Efficient Distance Matrices**] Calculating pairwise distances for samples between $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_r$, and $\mathbf{\Phi}_g^{hub}$ and $\mathbf{\Phi}_g$, which consumes $O(m_r n)$ and $O(m_g n)$ time and space, respectively. Please see Sec. 4.6.10 for an empirical justification of its effectiveness.

4. [**Efficient Sorting**] Sorting the distance matrices as required by the k-NN algorithm, which consumes $O(m_r n \log n)$ and $O(m_g n \log n)$ time respectively and no extra space.

5. [**Radii**] Recording the distance from each sample to its kth nearest neighbour as the radius of its hypersphere, taking $O(m_r)$ and $O(m_g)$ time and space, respectively.

6. [**Efficient Distance Matrix between $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_g^{hub}$**] Calculating pairwise distances between samples of $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_g^{hub}$, which consumes $O(m_r m_g)$ time and space.

7. [**Efficient P&R**] Calculating P&R ratios, taking $O(m_g^2)$ and $O(m_r^2)$ time and no extra space for each metric.

Theoretically, the proposed eP&R metrics run in $\max(O(m_r n \log n), O(m_g n \log n))$ time and consumes $\max(O(m_r n), O(m_g n))$ space while the original P&R metrics run in $O(n^2 \log n)$ time and consumes $O(n^2)$ space. Since $m_r < n$, $m_g < n$, the proposed eP&R metrics are far more efficient than the original P&R metrics.

4.6 Experiments

4.6.1 Experimental Setup

Hardware. We use a PC with an Intel(R) Core(TM) i7-10875H CPU, an NVIDIA RTX 4090 24GB GPU for small datasets and a GPU node with 2 NVIDIA V100 32GB GPUs for large datasets.

Datasets. We use the FFHQ [70] dataset containing 70k portrait images, and the LSUN (Car, Church, Cat, and Horse) dataset [154] containing 550k, 120k, 1.5m and 1.5m images of corresponding categories respectively in our experiments.

Table 4.6: The eP&R scores with different threshold t. Error(%): relative error $\epsilon = \frac{|x-\hat{x}|}{|x|}$

	Hubness			Error(%)		
t	Percent(%)	Precision	Recall	Precision	Recall	Mean
1	72.50±0.01	0.718±0.001	0.494±0.002	0.3	0.0	0.1
2	52.24±0.04	0.718±0.001	0.494±0.002	0.3	0.0	0.1
3	38.21±0.04	0.719±0.002	0.494±0.001	0.4	0.2	0.3
4	28.48±0.04	0.726±0.002	0.496±0.001	1.6	0.6	1.1
5	21.65±0.04	0.730±0.001	0.496±0.002	1.9	0.6	1.3
6	16.64±0.02	0.732±0.002	0.497±0.003	2.2	0.6	1.4
7	12.99±0.02	0.739±0.002	0.498±0.003	3.2	2.6	2.8
8	10.22±0.01	0.747±0.002	0.509±0.003	4.3	4.3	4.3
9	8.15±0.03	0.747±0.003	0.509±0.004	5.5	8.1	6.8
10	6.55±0.01	0.748±0.004	0.517±0.003	9.9	8.4	9.2
B.L.	—	0.716±0.001	0.493±0.001	—	—	—

Generative Models. Following [78], we test our eP&R metrics with StyleGAN2 [68] trained on the FFHQ and LSUN-Car, LSUN-Cat, LSUN-Church and LSUN-Horse datasets mentioned above. To demonstrate the generalizability of our metrics, we further test them with the other members of the StyleGAN family, including StyleGAN3 [65], Projected-GAN [119], VQ-VAE-2 [109] and the Latent Diffusion model [113] trained on the FFHQ dataset.

Hyper-parameters. Unless specified, we follow the original P&R [78] and use $k = 3$ in (approximate) k-NN algorithms for all P&R, eP&R calculations and hubness-aware sampling, and $t = 3$ as the threshold to extract hubs samples, and the FFHQ dataset and a StyleGAN2 model trained on it in our experiments.

4.6.2 Efficient vs. Original Precision and Recall

Approximation Error. Our eP&R is an approximation of the original P&R [78], which inevitably introduces errors. To demonstrate the validity of our approximation, we record the relative errors $\epsilon = \frac{|x-\hat{x}|}{|x|}$ in Table 4.1, where x is the original P&R result and $\hat{x}$ is our approximation. It can be observed that our eP&R metrics share almost identical results to the original P&R with very small relative errors around 1%. Please see sec. 4.6.7 for a comparison with reduced sampling of the original P&R, which further justifies the effectiveness of our metrics.

Table 4.7: Comparison with reduced sampling. # of Spls: number of samples. R.S.: reduced sampling, *i.e.*, instead of using the full dataset, we randomly sample a subset from it and use a reduced number of generated samples to calculate P&R accordingly. eP&R: our efficient P&R metrics. Time (S): serial implementation. Time (P): parallel implementation using CUDA. The last row shows the results of the original P&R as a reference.

# of Spls (%)	R.S.				eP&R			
	Precision	Recall	Time (P)	Memory	Precision	Recall	Time (P)	Memory
72.50	0.724±0.002	0.511±0.002	108s	13.12 GB	0.717±0.001	0.500±0.002	116s	16.34 GB
52.24	0.730±0.002	0.522±0.002	73s	12.23 GB	0.718±0.001	0.501±0.002	95s	15.27 GB
38.21	0.734±0.005	0.533±0.005	49s	8.47 GB	0.719±0.002	0.501±0.001	75s	14.21 GB
28.48	0.742±0.005	0.540±0.005	34s	6.22 GB	0.726±0.002	0.507±0.001	68s	13.10 GB
21.65	0.743±0.005	0.551±0.006	25s	5.01 GB	0.730±0.001	0.515±0.002	63s	12.28 GB
P&R (70k)	0.716±0.001	0.493±0.001	144s	19.90GB	-	-	-	-

Time and Memory Consumption. We profile the running time and memory consumption to compare the computational efficiency of our eP&R and the original P&R metrics. As Table 4.2 shows, our eP&R metrics run significantly faster and consume much less memory than the baseline, which justifies our complexity analysis in Sec. 4.5.3.

4.6.3 Ablation Study

As mentioned in Secs. 4.5.1 and 4.5.2, the proposed eP&R metrics consist of three components addressing Observation 4.5.1, Observation 4.5.2, and "Efficient Hubs Sample Identification" (approx. k-NN) respectively. To show their effectiveness, we conduct an ablation study as shown in Table 4.4. It can be observed that each of the proposed components contributes to the success of our eP&R metrics.

4.6.4 Choice of Hyperparameters

The proposed eP&R metrics have two hyperparameters: i) k used by the (approximate) k-NN algorithm; and ii) threshold t used to identify hubs samples.

Choice of number of nearest neighbours k. As Table 4.5 shows, it can be observed that improvements of our eP&R metrics are stable under different choices of k. Therefore, without loss of generality, we use $k = 3$ following [78].

Choice of threshold t. As Table 4.6 shows, our eP&R metrics introduce a trade-off between error and efficiency with t: the higher t, the more efficient our metrics but at the cost of higher errors. Thus, in our experiments, we strike a balance by using $t = 3$ for (FFHQ, StyleGAN2) combination.

Table 4.8: Time costs when matrix tiling is used. The experiments are conducted using the FFHQ dataset and a StyleGAN2 model trained on it.

# of Tiles	1 (no tiling)	2	5	10	50	100
Time (P)	144s	146s	148s	150s	174s	192s

4.6.5 Robustness against the Truncation Trick

Our eP&R metrics are also robust against the truncation trick, a widely used technique that improves GAN sample quality by truncating the latent vector z [68]. The truncation trick is a widely used technique that improves GAN sample quality by truncating the latent vector z fed into the generator [18, 66, 68]. As Table 4.9 shows, our eP&R metrics are robust against the truncation trick with $\phi = 0.5, 0.7$, where $\phi = 0.7$ is the recommended value.

Table 4.9: Robustness against the truncation trick [66]. We calculate the metrics using StyleGAN2 trained on the FFHQ dataset and $t = 4$. Please note that $\phi = 0.7$ is the recommended value [66, 68] for the truncation trick and $\phi = 1.0$ means no truncation is applied at all. B.L.: the original P&R metrics as the baseline [78]. eP&R: our efficient P&R metrics. We did not include Time and Memory costs are the truncation trick does not affect the number of samples, hence consuming the same amount of time and memory.

	$\phi = 0.5$		$\phi = 0.7$		$\phi = 1.0$	
	Precision	Recall	Precision	Recall	Precision	Recall
eP&R	0.932±0.002	0.089±0.002	0.890±0.002	0.297±0.002	0.714±0.002	0.493±0.001
B.L.	0.935±0.001	0.101±0.001	0.885±0.001	0.308±0.001	0.716±0.001	0.493±0.001

Table 4.10: Time and space consumption of our eP&R metrics compared to the original P&R metrics using StyleGAN2 trained on the LSUN-Horse dataset. B.L.: the original P&R metrics as the baseline [78]. eP&R: our efficient P&R metrics. Time (S): serial implementation. Time (P): parallel implementation using CUDA. The profiling items are in one-to-one correspondence with the stages listed in Sec. 4.5.3.

Profiling	B.L.		Profiling	eP&R	
	Time (S)	Time (P)		Time (S)	Time (P)
			Subspace ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	8min	3min
DMs ($\mathbf{\Phi}_r$, $\mathbf{\Phi}_g$)	12h02min	54min	A. hubs ($\mathbf{\Phi}_r^{hub}$, $\mathbf{\Phi}_g^{hub}$)	2min	1min
			eDMs	5h56min	26min
Sorting	2h56min	22min	eSorting	1h20min	11min
Radii	1min	1min	Radii	50s	50s
DM ($\mathbf{\Phi}_r \leftrightarrow \mathbf{\Phi}_g$)	6h30min	34min	eDM ($\mathbf{\Phi}_r^{hub} \leftrightarrow \mathbf{\Phi}_g^{hub}$)	1h27min	8min
P&R	2h12min	17min	eP&R	56min	4min
Total	23h45min	2h10min	Total	9h50min	**64min**

4.6.6 P&R Curves

We follow [78] and include the original P&R (baseline) and our eP&R curves against the parameter of the truncation trick in Fig. 4.4. The results show that our method approximates the original P&R curves well on both FFHQ and LSUN-Church datasets.

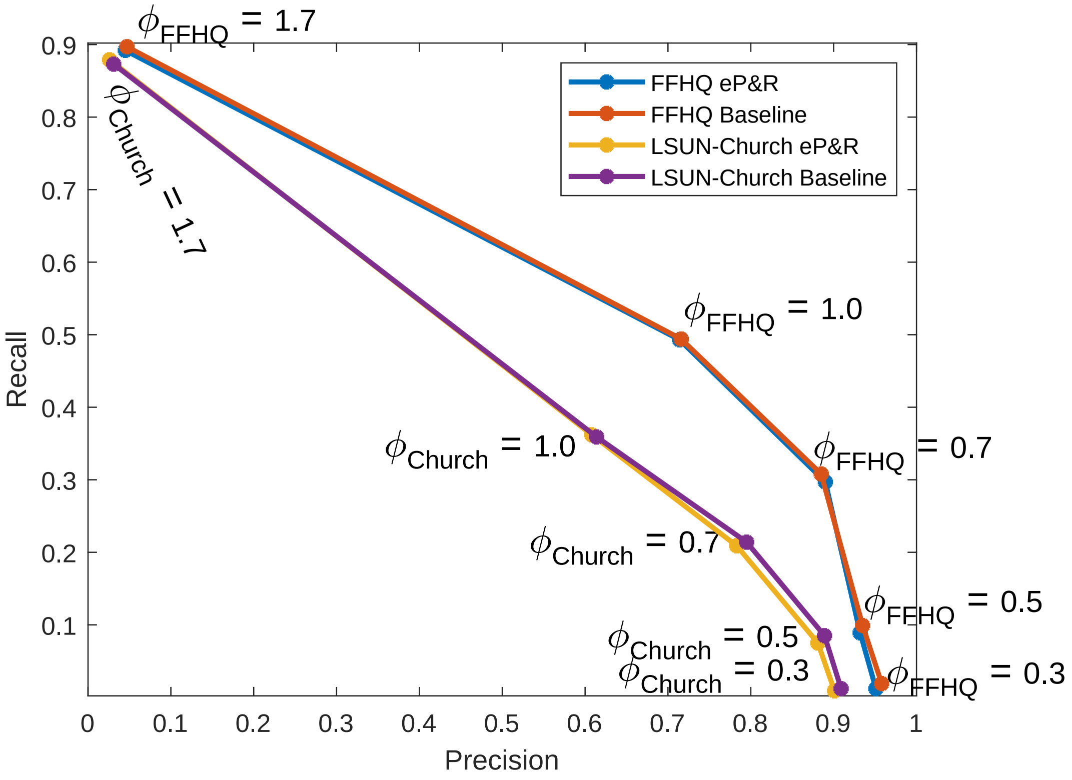


Figure 4.4: Original P&R (baseline) and our eP&R curves on the FFHQ and LSUN-Church datasets.

4.6.7 Comparison with Reduced Sampling

To further demonstrate the superiority of our eP&R metrics, we compare them with another baseline of reduced sampling, *i.e.*, instead of using the full dataset, we randomly sample a subset from it and use a reduced number of generated samples to calculate P&R accordingly. As Table 4.7 shows, our method provides much more accurate P&R results given the same number of samples, demonstrating the superiority of our metrics.

4.6.8 Time and Space Consumption for large datasets

Due to hardware limitations, we have to perform matrix tiling when calculating P&R and eP&R on large datasets which splits a given matrix into tiles (submatrices) that can fit into GPU memory. However, this introduces additional overheads and is not desirable (Table 4.8), which further justifies our motivation to design efficient evaluation metrics for generative models. Nevertheless, we show the results of our eP&R metrics on the LSUN-Horse dataset containing 1.5m images in Table 4.10. It can be observed that our metrics still save a lot of time when matrix tiling is used.

4.6.9 Insensitivity to Group Split Points

As shown in Table 4.11, the ratios of binary scores are similar for each hubness value on the FFHQ dataset, which validates the insensitivity of the choice of group split points for Observation 4.5.1 and Fig. 4.1.

Similarly, as shown in Table 4.12, the ratios of hubness samples increase quickly to 1 with the increase of $|\phi'|$ on the FFHQ dataset, which validates the insensitivity of the choice of group split points for Observation 4.5.2 and Fig. 4.2.

We show the same conclusions hold on the LSUN-Church dataset as well (Table 4.13 and Table 4.14).

4.6.10 Justification of pairwise distance calculation between $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_r$

As Table 4.15 shows, we calculate the pairwise distances between $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_r$ as it provides lower approximation errors than calculating pairwise distances for samples in $\mathbf{\Phi}_r^{hub}$. We conjecture the reason is that $\mathbf{\Phi}_r^{hub}$ is much sparser than $\mathbf{\Phi}_r$ and thus the pairwise distances for samples in it will be much larger than those of the original P&R, resulting in much larger k-NN hyperspheres that increase the approximation error. The same conclusion holds for $\mathbf{\Phi}_g^{hub}$ and $\mathbf{\Phi}_g$.

4.7 t-SNE visualization of the hubness set and the original set

As shown in Fig. 4.5, we included the t-SNE results of:

- (a) Hubness set vs. original set (FFHQ dataset, Φ_r) with thresholds t of 3, 5, 7.
- (b) Hubness set vs. original set (StyleGAN trained on the FFHQ dataset, Φ_g) with thresholds t of 3, 5, 7.

- (c) Hubness set vs. original set (LSUN-Church dataset, Φ_r) with thresholds t of 3, 5, 7.
- (d) Hubness set vs. original set (StyleGAN trained on the LSUN-Church dataset, Φ_g) with thresholds t of 3, 5, 7.

It can be observed that the hubness set approximates the original set well when the threshold $t = 3$, which not only justifies the effectiveness of our approach but also our choice of hyperparameter $t = 3$.

4.8 Conclusion

In conclusion, we have proposed efficient precision and recall (eP&R) metrics that provide almost identical results as the original P&R metrics but with much lower computational costs. By identifying and removing redundancies in P&R computation through hubness-aware sampling and approximate k-NN methods, we have developed a highly efficient yet accurate approach to evaluating generative models. Extensive experiments demonstrate the effectiveness of our eP&R metrics. Going forward, eP&R provides an important step towards feasible and insightful assessment of state-of-the-art generative models trained on massive datasets. We believe eP&R can enable more rapid progress in this exciting field.

Limitations and Future Work. Although effective and efficient, the proposed eP&R metrics are not fully optimized. One area for improvement is in Stage 3 (Efficient Distance Matrices), which currently calculates pairwise distances between hub samples of one set and all samples of the other set to compute radii. A significant amount of time is spent on this step. We could optimize this by utilizing the subspace constructed by the approximate k-NN algorithms. Instead of comparing hubs to the full set, we would only need to calculate distances between hubs and samples within the relevant subspace of the other set. This would allow us to find radii much more quickly. While the current metrics are fast and accurate, optimizations like these could push the efficiency even higher without sacrificing effectiveness. We therefore see continued refinement of the eP&R metrics represents an exciting opportunity for future work.

Table 4.11: Insensitivity to group split points for Observation 4.5.1 and Fig. 4.1 (FFHQ). Hub. Value: hubness value, B.S.: binary score.

(a) All 70k images in the FFHQ dataset

Hub. Value	1	2	3	4	5	6	7	8
B.S. = 1	9038	7489	5795	4579	3525	2741	2188	1842
All Samples	13382	11292	8615	6815	5250	4158	3235	2664
Ratio	0.675	0.663	0.673	0.672	0.671	0.659	0.676	0.691

Hub. Value	9	10	11	12	13	14	15	16
B.S. = 1	1474	1244	931	818	679	593	508	425
All Samples	2190	1791	1389	1231	1046	883	776	640
Ratio	0.673	0.695	0.67	0.665	0.649	0.672	0.655	0.664

Hub. Value	17	18	19	20	21	22	23	24
B.S. = 1	361	305	272	246	200	177	153	160
All Samples	543	469	409	361	291	264	232	230
Ratio	0.665	0.65	0.665	0.681	0.687	0.67	0.659	0.696

(b) 70k images generated by StyleGAN2

Hub. Value	1	2	3	4	5	6	7	8
B.S.e = 1	10030	9150	7375	5858	4555	3664	2952	2266
All Samples	12083	10988	8833	7015	5452	4402	3561	2743
Ratio	0.83	0.833	0.835	0.835	0.835	0.832	0.829	0.826

Hub. Value	9	10	11	12	13	14	15	16
B.S.e = 1	1848	1515	1358	1120	911	800	646	522
All Samples	2237	1835	1624	1344	1117	992	762	635
Ratio	0.826	0.826	0.836	0.833	0.816	0.806	0.848	0.822

Hub. Value	17	18	19	20	21	22	23	24
B.S.e = 1	456	374	350	306	261	228	187	178
All Samples	539	467	420	379	317	277	217	208
Ratio	0.846	0.801	0.833	0.807	0.823	0.823	0.862	0.856

Table 4.12: Insensitivity to group split points for Observation 4.5.2 and Fig. 4.2 (FFHQ).

(a) All 70k images in the FFHQ dataset

ϕ'	0	1	2	3	4	5	≥6
Hubness	11922	8625	6290	4540	3516	2741	10818
Non-hubness	13254	9089	6519	4598	3537	2749	10818
Ratio	0.9	0.949	0.965	0.987	0.994	0.997	1.000

(b) 70k images generated by StyleGAN2

ϕ'	0	1	2	3	4	≥5
Hubness	11792	6596	4136	2745	1860	5531
Non-hubness	13482	7210	4328	2788	1874	5531
Ratio	0.875	0.915	0.956	0.985	0.992	1.000

Table 4.13: Insensitivity to group split points for Observation 4.5.1 and Fig. 4.1 (LSUN-Church). Hub. Value: hubness value, B.S.: binary score.

(a) All 120k images in the LSUN-Church dataset

Hubness Value	1	2	3	4	5	6	7	8	9
Binary Score = 1	51045	17414	6653	2906	1413	658	355	202	108
All Samples	75650	25836	9866	4296	2074	1006	562	291	153
Ratio	0.675	0.674	0.674	0.676	0.681	0.654	0.631	0.694	0.708

(b) 100k images generated by StyleGAN2

Hubness Value	1	2	3	4	5	6	7	8	9	10
Binary Score = 1	24459	4658	1081	405	161	71	35	19	9	6
All Samples	55401	10564	2385	957	371	154	81	38	21	12
Ratio	0.441	0.441	0.453	0.423	0.434	0.461	0.432	0.5	0.429	0.5

Table 4.14: Insensitivity to group split points for Observation 4.5.2 and Fig. 4.2 (LSUN-Church).

(a) All 120k images in the LSUN-Church dataset

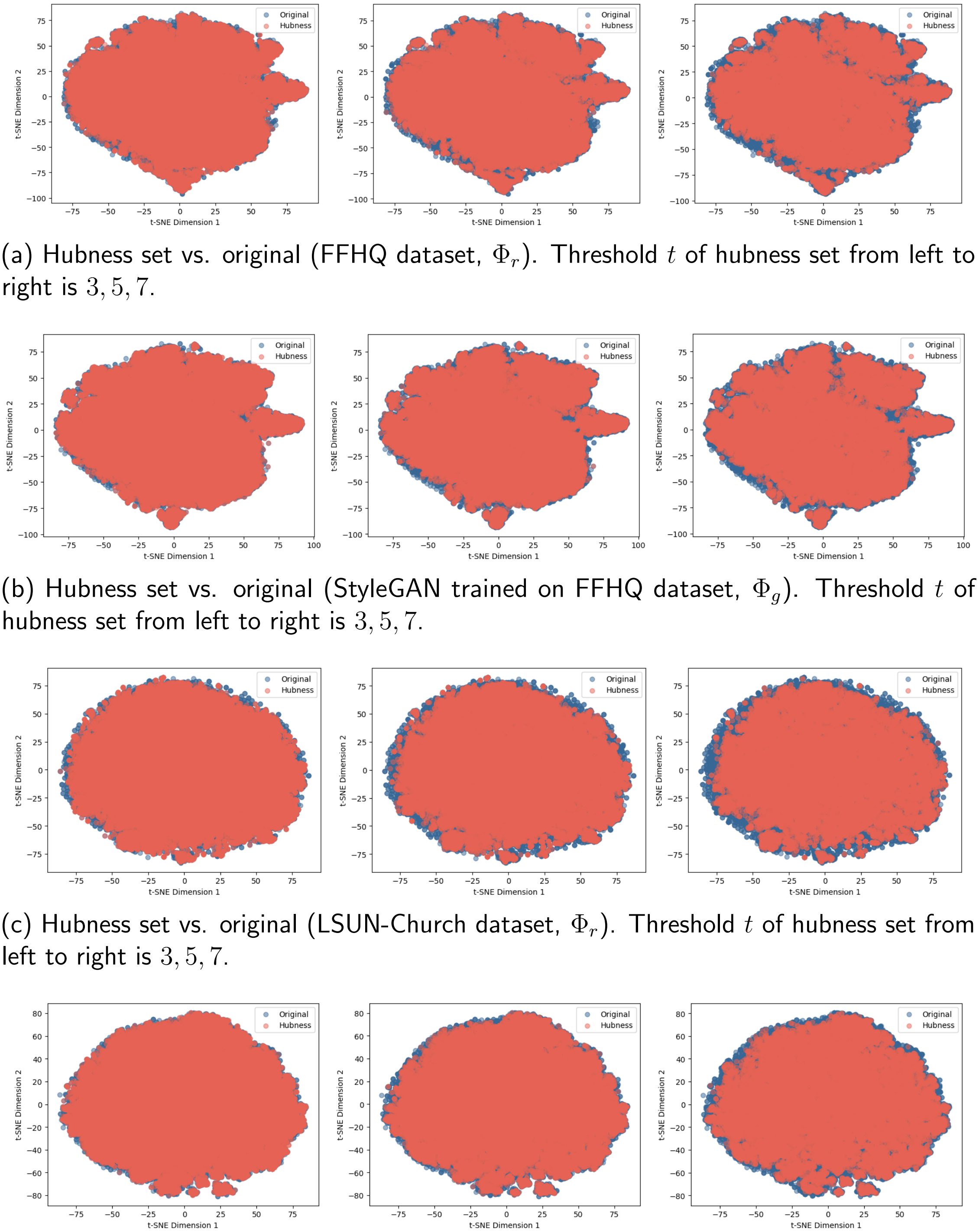
ϕ'	0	1	2	3	4	5	≥6
Hubness	29986	18482	12819	9106	6749	5037	20441
Non-hubness	32669	19615	13346	9213	6773	5040	20441
Ratio	0.918	0.942	0.961	0.988	0.996	0.999	1

(b) 100k images generated by StyleGAN2

ϕ'	0	1	2	3	4	5	6	≥7
Hubness	15731	7063	3984	2405	1505	963	728	2007
Non-hubness	17590	7821	4158	2443	1513	965	729	2007
Ratio	0.894	0.903	0.958	0.984	0.995	0.998	0.999	1

Table 4.15: Justification of pairwise distance calculation between $\mathbf{\Phi}_r^{hub}$ and $\mathbf{\Phi}_r$. P: Precision; R: Recall. Error(%): relative error $\epsilon = \frac{|x-\hat{x}|}{|x|}$, where x denotes the baseline precision 0.716 and recall 0.493 of the original P&R metric.

t	Pairwise distance between $\mathbf{\Phi}_r^{hub}$, $\mathbf{\Phi}_r^{hub}$				Pairwise distance between $\mathbf{\Phi}_r^{hub}$, $\mathbf{\Phi}_r$			
	P	R	P Error(%)	R Error(%)	P	R	P Error(%)	R Error(%)
1	0.713	0.484	0.4	1.8	0.717	0.500	0.3	0.0
2	0.723	0.506	1.0	2.6	0.718	0.501	0.3	0.0
3	0.746	0.534	4.2	8.3	0.719	0.501	0.4	0.2
4	0.768	0.562	7.3	14.0	0.726	0.507	1.6	0.6
5	0.787	0.588	9.9	19.3	0.730	0.515	1.9	0.6



(a) Hubness set vs. original (FFHQ dataset, Φ_r). Threshold t of hubness set from left to right is $3, 5, 7$.

(b) Hubness set vs. original (StyleGAN trained on FFHQ dataset, Φ_g). Threshold t of hubness set from left to right is $3, 5, 7$.

(c) Hubness set vs. original (LSUN-Church dataset, Φ_r). Threshold t of hubness set from left to right is $3, 5, 7$.

(d) Hubness set vs. original (StyleGAN trained on LSUN-Church dataset, Φ_g). Threshold t of hubness set from left to right is $3, 5, 7$.

Figure 4.5: t-SNE visualization of the hubness set and the original set.

Chapter 5

Hubness Sampling to Train GAN Models for Mitigating Dataset Bias

This Chapter addresses the critical issue of dataset bias and synthetic fairness in generative models, focusing on achieving balanced training and equal representation across various categories. Recent advancements in GANs have highlighted the need for methodologies that enhance fairness without extensive labeling and computational resources. This study introduces hubness sampling as an unsupervised, pre-training-free approach to training fair generative models. Statistical analyses reveal that the likelihood of latents with high hubness values mapping to minority classes is initially low but significantly improved with hubness sampling. Furthermore, the method shows substantial improvements in fairness across different categories, including ethnicity, gender, make-up, and age. This chapter applies hubness sampling to train several state-of-the-art GAN models, including StyleGAN, Diffusion-GAN, and GANFormer, and evaluates the quality and diversity of synthetic images using established metrics such as Precision & Recall and FID. Experimental results demonstrate that hubness sampling significantly enhances both balance and diversity in generated outputs, effectively mitigating dataset-induced bias. Overall, the hubness sampling method presents a novel approach to improving diversity and fairness in GAN training, reducing bias, and enhancing minor class representation.

5.1 Introduction

In recent years, substantial efforts have been made to address dataset bias and enhance synthetic fairness, a topic that remains central to ongoing research. The objective of fairness in generative models is to achieve balanced training, also known as equal representation, which entails generating samples that adhere to a uniform distribution across categories [118, 134, 151]. Recent research indicates that the fairness of GAN models can be enhanced by pre-learning the feature distribution through methodologies such as weak supervision and transfer learning [25, 134]. However, these methods necessitate substantial effort to label diverse datasets, identify minor and major features, and develop a new balanced supervised model with weak supervision or adapt to a class-balanced set of the real images after pre-trained with the full dataset, resulting in significant time and computational resource expenditure. In this chapter, we present the first study to investigate the potential of hubness sampling as an unsupervised and pre-training-free approach for training fair generative models. Our method aims to reduce computational overhead and resource waste while promoting equitable representation in the generated output.

As shown in Table 5.1, the statistic reveals a clear imbalance in the ethnic distribution within the FFHQ dataset, where the proportion of white individuals is significantly higher than that of other ethnic groups. This disparity may lead to biased representations and performance inconsistencies in generative models trained on such datasets. To address this issue, we explore the use of hubness sampling as a strategic data selection method during the training process. In this chapter, we apply hubness sampling to train several popular and representative Generative Adversarial Network (GAN) models, including StyleGAN, Diffusion-GAN, and GANFormer. These state-of-the-art GAN architectures are recognized for their robust and stable training behavior across diverse image datasets, which minimizes the influence of incidental factors that often hinder training outcomes. By integrating hubness-aware sampling with these models, we aim to promote more balanced data utilization, enhance the fairness of the generated outputs, and provide a principled approach to mitigating dataset-induced bias in generative image modeling. In addition, we

Ethnicity	East-asian	Black	White	Other
ratio(%)	17.943	5.911	56.190	19.956

Table 5.1: The ethnicity statistic analysis for the FFHQ dataset. The highest ratio, white, exhibits the highest degree of feature concentration.

will employ established generative model evaluation metrics, Precision & Recall and Fréchet Inception Distance (FID), to quantitatively assess the quality and diversity of the synthetic images produced.

Our main contributions can be summarized as follows:

1. We introduce a hubness-aware sampling method for training GAN-based models, demonstrating its effectiveness in enhancing both balance and diversity in the generated outputs.

2. Comprehensive experimental evaluations across multiple GAN architectures and datasets validate the effectiveness and robustness of our proposed sampling method.

5.2 Balance affected with batch size

The selection of batch size in unconditional generative modeling introduces a crucial empirical trade-off between gradient stability and optimal feature distribution balance [18]. Larger batch sizes typically increase statistical precision and reduce gradient variance, and our experimental analysis demonstrates that this high variance severely hinders overall feature balance, particularly for complex and subtle distributions (e.g., Age). As shown in the tables 5.2, the moderately larger batch (Batch = 48) consistently provides sufficient **statistical precision** to capture the feature manifold more comprehensively, resulting in a better balance across Ethnicity, Gender, and Age distributions. The Batch = 48 setting achieves superior FD_o values (as low as $\mathbf{0.034}$ for Ethnicity and $\mathbf{0.036}$ for Age), confirming that a critical level of statistical stability is necessary to avoid mode collapse and capture underrepresented features effectively. This observation confirms that the choice of batch size profoundly impacts the optimization path, directly determining the model's capacity to achieve

Table 5.2: The balance illustration in training StyleGAN-2 across attributes such as ethnicity, gender and age with different batch size. The fairness discrepancy (FD) metric measures the difference between the expected classifier output and a fairness probability, with FD$_f$ set to a uniform distribution and FD$_o$ set to the original ratio of attributes, ensuring a structured evaluation of fairness in generative models.

(a) Ethnicity

	East-asian	Black	South-asian	Lat-hisp	Mid-easten	White	FD$_o$ ↓	FD$_f$ ↓
Original	17.943%	5.911%	3.050%	10.741%	6.164%	56.190%	—	0.448
Batch(4)	20.555%	3.110%	1.415%	9.410%	5.200%	60.310%	0.061	0.502
Batch(48)	18.960%	4.985%	2.569%	9.466%	5.204%	58.815%	0.034	0.480

(b) Gender and Age

	Female	Male	FD$_o$ ↓	FD$_f$ ↓	(0-18)	(19-36)	(36-54)	(54+)	FD$_o$ ↓	FD$_f$ ↓
Original	55.263%	44.737%	—	0.074	18.134%	50.167%	23.321%	8.377%	—	0.310
Batch(4)	61.025%	38.975%	0.081	0.156	1.665%	74.750%	22.410%	1.185%	0.304	0.599
Batch(48)	58.793%	41.207%	0.067	0.141	17.007%	53.161%	23.898%	6.925%	0.036	0.344

feature distribution balance and influencing the fairness of generated samples. The results confirm that prioritizing statistical precision, as demonstrated by the superior performance of Batch = 48, is crucial for avoiding collapse into majority modes and ensuring robust representativeness.

To overcome the instability of small-batch stochasticity while maintaining comprehensive mode coverage, Hubness sampling offers a systematic solution. This approach targets the geometric bias in the feature space where majority features form stable hubs and minority features reside in sparse, underrepresented regions. The model's natural tendency is to reinforce the hubs, leading to feature imbalance. Hubness sampling addresses this by implementing a sampling strategy that actively identifies and increases the sampling probability of data points located in these sparse regions (i.e., points rarely selected as nearest neighbors). By intentionally exposing the Discriminator to these underrepresented feature modes, hubness sampling compels the Generator to synthesize diverse samples. This method replaces unreliable random stochasticity with controlled, targeted sampling, thereby effectively stabilizing the optimization process and significantly improving feature distribution balance across the entire manifold.

5.3 Hubness Sampling Method

Algorithm 3 Latent Hubness Sampling for GAN training

Input: a set of GAN latents $S = \{z_1, z_2, ..., z_n\}$ sampled from a standard normal distribution $\mathcal{N}(0, I)$, a hyper-parameter k, the batch size x

Output: discriminator D and generator G

```
  # Step 1
  $m_{1,2,...,n} \leftarrow$HubnessValue($S$)
  # Step 2
  Rank($m_{1,2,...,n}$) mapping $S$
  $Z_h = S_{1 \rightarrow x}$
  # Step 3
  $X = G(Z_h)$
  minmax $\log D(X) + \log(1 - D(G(Z_h)))$
```

The hubness sampling algorithm for training GAN models, to be different with the previous hub algorithm (Alg 1), has been updated to ensure that a fixed number of top hub-value latents are selected from a pool of random samples. This adjustment helps maintain consistency in the batch size during GAN training, which is crucial for stable model performance and effective optimization. The updated approach is outlined in detail in Alg 3.

Instead of applying a threshold to select samples, which may lead to an unpredictable batch size, we rank the hubness values from high to low and select the top-x latents, where x corresponds to the batch size. This ensures that the batch size remains fixed throughout training.

Moreover, high hub-value latents represent regions of the latent space with higher sampling density, which are more likely to contribute to the generation of high-quality images. The hypothesis here is that training GANs with latent inputs that have higher hubness scores could provide more informative gradients during back-propagation, thereby improving the model's ability to approximate the real data distribution.

5.4 Experiment

In our experiments, we investigate how different latent sampling strategies influence GAN training. In particular, we evaluate a hubness-aware sampling method, and compare it against Gaussian and normalization-based sampling baselines. The experiment focuses primarily on the training of Diffusion-GAN [145], Style-GAN2 [68] and GANFormer [57] recent state-of-the-art generative models, using the FFHQ [66], LSUN (Bedroom, Church, Cat and Horse)[154], and Landscape[125] datasets.

In this section, we will examine how training latents using the hubness sampling method enhances the fairness of GANs based on the FFHQ dataset and we apply random latents as inference to generate images after training. The performance of the models is evaluated using metrics such as Precision & Recall and the Fréchet Inception Distance (FID) to assess the quality and diversity of generated images. Additionally, we employ the CLEAM score [132] and the Fairness Discrepancy (FD) [133] to measure the fairness of the GANs. Our proposed method demonstrates significant improvements in both the diversity and fairness of GAN-generated content.

In this study, the hubness sampling method is also compared with Gaussian and normalization sampling strategies to evaluate its effectiveness in improving GAN training outcomes and normalization sampling is established as the baseline. Note that we apply the different sampling methods to train the GANs, but the latents to test are applied with random samplings. But we also discuss GANs trained with the hubness sampling to test with the hubness latents.

The experiments are conducted on a PC equipped with an Intel(R) Core(TM) i7-10875H CPU and an NVIDIA RTX 4090 GPU with 24GB of VRAM, ensuring sufficient computational resources for training the models effectively.

5.4.1 Hubness Fairness statistic

In this section, we will present the statistical results for different categories based on ethnicity, gender, make-up, and age using Diffusion-GAN, StyleGAN-2, and GANFormer. These models were trained with Gaussian sampling latents, normalization sampling latents, and hubness sampling latents. From these results, it can be argued that our method significantly mitigates dataset bias and enhances fairness.

Moreover, the research by [156] addresses the issue of bias in generative models, which arises due to imbalances in training datasets, leading to under-representation of minority groups, highlighting the problem of data coverage.

Table 5.3: The balance illustration in training Diffusion-GAN, StyleGAN-2, and GANFormer across attributes such as ethnicity, age, gender, and makeup. Highlight how hubness sampling effectively enhances tail data for study. The fairness discrepancy (FD) metric measures the difference between the expected classifier output and a fairness probability, with FD_f set to a uniform distribution and FD_o set to the original ratio of attributes, ensuring a structured evaluation of fairness in generative models.

Ethnicity	East-asian	Black	South-asian	Lat-hisp	Mid-easten	White	$\text{FD}_o \downarrow$	$\text{FD}_f \downarrow$
Original	17.943%	5.911%	3.050%	10.741%	6.164%	56.190%	—	0.448
				Diffusion-GAN				
Hubness	17.379%	7.405%	3.173%	10.920%	6.786%	54.336%	**0.025**	**0.426**
Gaussian	19.014%	5.891%	2.883%	8.562%	5.657%	57.992%	0.031	0.470
Normal	19.009%	5.973%	2.670%	8.546%	5.762%	58.041%	0.031	0.470
				StyleGAN-2				
Hubness	17.621%	6.885%	3.329%	10.920%	6.341%	54.903%	**0.017**	**0.433**
Gaussian	18.689%	4.935%	2.963%	9.682%	5.146%	58.585%	0.031	0.476
Normal	18.960%	4.985%	2.569%	9.466%	5.204%	58.815%	0.034	0.480
				GANFormer				
Hubness	17.249%	6.726%	3.721%	11.338%	6.857%	54.113%	**0.026**	**0.423**
Gaussian	18.617%	5.689%	2.972%	10.239%	4.291%	58.192%	0.029	0.472
Normal	19.098%	5.693%	2.909%	10.766%	4.025%	57.509%	0.028	0.467

Gender	Female	Male	$\text{FD}_o \downarrow$	$\text{FD}_f \downarrow$	Make-Up	Yes	No	$\text{FD}_o \downarrow$	$\text{FD}_f \downarrow$
Original	55.263%	44.737%	—	0.074	Original	54.503%	45.497%	—	0.064
				Diffusion-GAN					
Hubness	55.476%	44.524%	**0.003**	**0.077**	Hubness	50.467%	49.533%	**0.057**	**0.007**
Gaussian	57.375%	42.625%	0.030	0.104	Gaussian	59.625%	40.375%	0.072	0.136
Normal	58.121%	41.879%	0.040	0.115	Normal	59.504%	40.496%	0.071	0.134
				StyleGAN-2					
Hubness	55.476%	44.524%	**0.003**	**0.077**	Hubness	54.622%	45.378%	**0.002**	**0.456**
Gaussian	59.803%	40.198%	0.064	0.139	Gaussian	59.504%	41.496%	0.064	0.128
Normal	59.973%	40.027%	0.067	0.141	Gaussian	59.504%	41.496%	0.064	0.128
				GANFormer					
Hubness	55.527%	44.473%	**0.004**	**0.078**	Hubness	54.563%	45.437%	**0.001**	**0.065**
Gaussian	57.375%	42.625%	0.030	0.104	Gaussian	57.408%	42.592%	0.041	0.105
Normal	57.121%	42.879%	0.026	0.101	Normal	57.427%	42.573%	0.041	0.105

Age	(0-18)	(19-36)	(36-54)	(54+)	$\text{FD}_o \downarrow$	$\text{FD}_f \downarrow$
Original	18.134%	50.167%	23.321%	8.377%	—	0.310
			Diffusion-GAN			
Hubness	18.032%	50.147%	23.382%	8.439%	**0.001**	**0.309**
Gaussian	16.756%	52.423%	24.094%	6.727%	0.032	0.340
Normal	16.316%	53.161%	24.098%	6.425%	0.041	0.348
			StyleGAN-2			
Hubness	18.034%	50.146%	23.381%	8.439%	**0.001**	**0.309**
Gaussian	16.276%	53.090%	23.929%	6.705%	0.039	0.347
Normal	17.007%	53.161%	23.898%	6.925%	0.036	0.344
			GANFormer			
Hubness	18.017%	50.200%	23.173%	8.610%	**0.003**	**0.309**
Gaussian	18.276%	51.403%	24.094%	6.227%	0.026	0.331
Normal	17.507%	51.699%	24.398%	6.396%	0.028	0.334

Table 5.4: The radium of features with different samplings. Explore that the hubness sampling can make the space fairness to have the similar radium and the random sampling (gaussian and normal) will lead to the bias, because of the dataset non-fairness.

(a) The latents radium in different Ethnicity

Ethnicity	East-asian	Black	South-asian	Lat-Hisp	Mid-East	White
Hubness	25.155	25.076	25.264	25.181	25.345	25.404
Gaussian	25.293	24.541	24.424	24.308	24.450	25.954
Normal	25.063	24.451	24.477	24.557	24.554	25.881

(b) The latents radium in make-up and gender

Make-up	Yes	No	Gender	Yes	No
Hubness	25.404	25.361	Hubness	25.587	25.474
Gaussian	25.317	25.956	Gaussian	25.768	25.577
Normal	25.378	25.586	Normal	25.678	25.400

(c) The latents radium in different age periods

Age	(0-18)	(19-36)	(36-54)	(54+)
Hubness	25.228	25.396	25.271	25.051
Gaussian	24.519	25.955	24.979	24.403
Normal	25.074	25.479	25.573	24.208

Table 5.5: Diffusion-GAN with hubness sampling, gaussian sampling and normalization sampling trained on FFHQ, LSUN-(Church, Cat, Horse, Bedroom) and Landscape. The hubness sampling, having the best recall score, can be helpful to improve the diversity of the model. (1), (2) and (3) are the models trained with different sampling, (1) Norm. (2) Gauss. (3) Hubness. (4) is model trained with hubness samplings and test with hubness latents.

	FFHQ			LSUN-Church			LSUN-Cat		
	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID
(1)	**0.708**	0.419	**3.280**	0.603	0.357	3.950	0.574	0.251	8.746
(2)	0.698	0.430	3.661	**0.606**	0.393	4.031	0.571	0.234	9.306
(3)	0.683	**0.440**	3.539	0.604	**0.400**	**3.659**	**0.576**	**0.265**	**8.720**
(4)	0.693	0.427	3.623	0.627	0.388	3.828	0.634	0.236	9.232

	LSUN-Horse			LSUN-Bedroom			Landscape		
	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID
(1)	0.644	0.357	**3.145**	0.585	0.240	4.649	**0.748**	0.478	**2.872**
(2)	0.641	0.352	3.342	0.578	0.130	5.669	0.747	0.473	2.918
(3)	0.644	**0.367**	3.292	**0.587**	**0.254**	**4.113**	0.742	**0.485**	2.894
(4)	0.656	0.359	3.871	0.615	0.243	4.362	0.785	0.406	4.001

Table 5.6: StyleGan-2 and GANFormer with hubness sampling, gaussian sampling and normalization sampling trained on FFHQ, and LSUN-(Church, Bedroom). The hubness sampling, having the best recall score, can be helpful to improve the diversity of the model. (1), (2) and (3) are the models trained with different sampling, (1) Norm. (2) Gauss. (3) Hubness. (4) is model trained with hubness samplings and test with hubness latents.

(a) StyleGAN-2.

	FFHQ			LSUN-Church			LSUN-Bedroom		
	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID
(1)	0.680	0.439	3.583	0.594	0.467	2.680	0.565	0.368	2.813
(2)	**0.685**	0.437	3.587	0.595	0.480	**2.502**	0.565	0.353	2.836
(3)	0.668	**0.464**	**3.570**	0.595	**0.497**	2.626	0.563	**0.373**	**2.755**
(4)	0.683	0.457	3.578	0.612	0.484	2.726	0.578	0.361	3.008

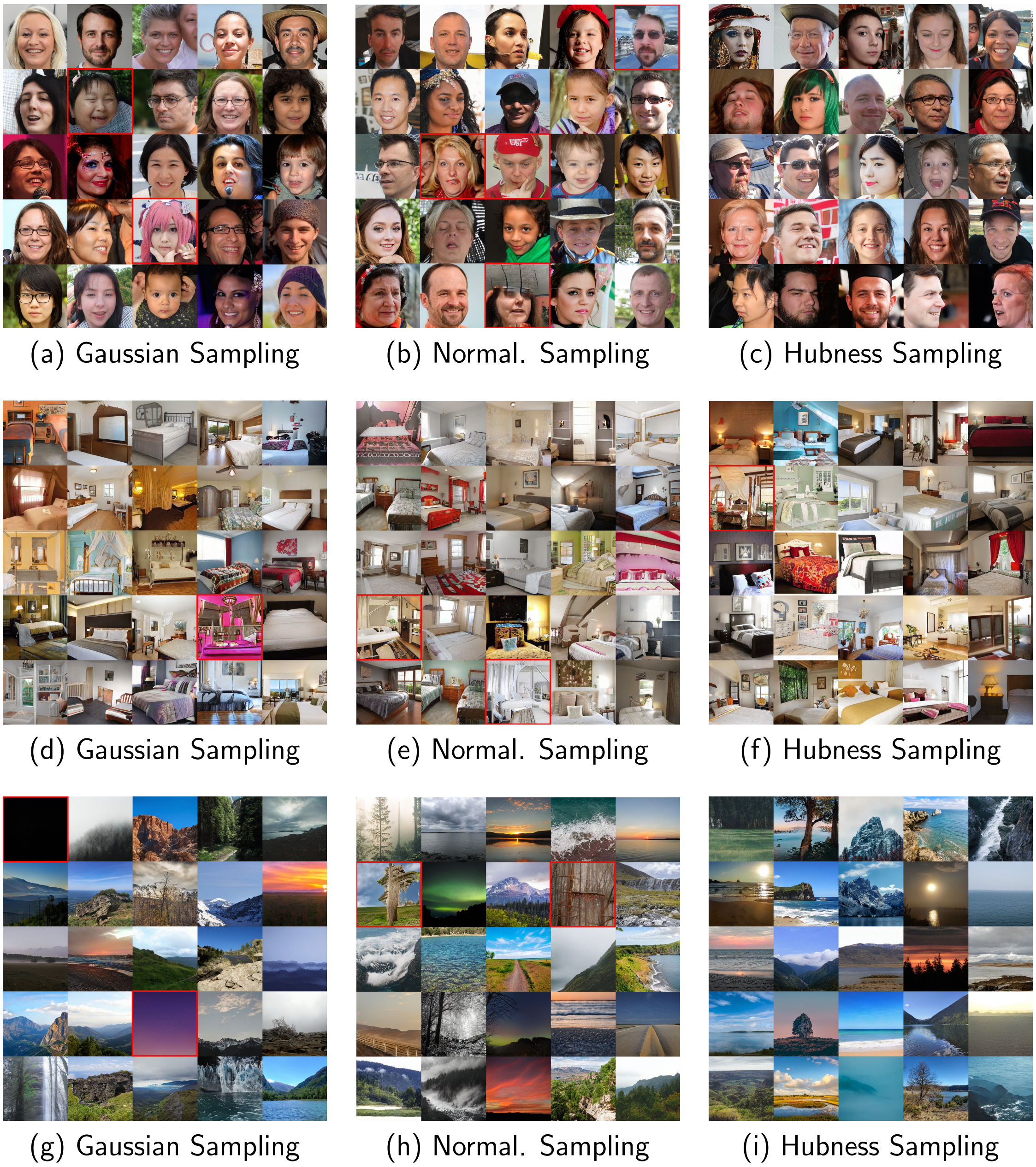
(b) GANFormer.

	FFHQ			LSUN-Church			LSUN-Bedroom		
	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID
(1)	0.616	0.245	9.484	0.354	0.192	12.771	0.502	0.225	5.256
(2)	0.689	0.272	6.184	0.495	0.223	7.334	0.530	0.218	5.072
(3)	**0.691**	**0.307**	**5.055**	**0.610**	**0.293**	**4.040**	**0.532**	**0.254**	**5.021**
(4)	0.685	0.285	4.867	0.602	0.274	4.095	0.514	0.224	5.651

Balance training with hubness samplings

As presented in Table 5.4 and Table 5.3, we analyze the coverage of various classes within the FFHQ image dataset, which highlight the coverage of real images by generative models. The results demonstrate that our hubness sampling method effectively mitigates convergence issues caused by class imbalance in large-scale datasets. This approach helps maintain a more consistent distribution of feature radii within the latent space. Furthermore, from the statistics shown in Table 5.3, it is evident that the ratio of minority (tailed) classes increases significantly, while the dominance of majority classes decreases. These findings strongly indicate that hubness sampling enhances the fairness of model training.

Hubness sampling may also enhance the discriminator's ability to distinguish between real and generated samples. By providing a more concentrated and representative set of latent variables, hubness sampling allows the generator to focus on the discriminator's "hard" regions—those that closely resemble real data. This targeted learning helps the generator better approximate the real data manifold, ultimately improving its performance in generating realistic samples.

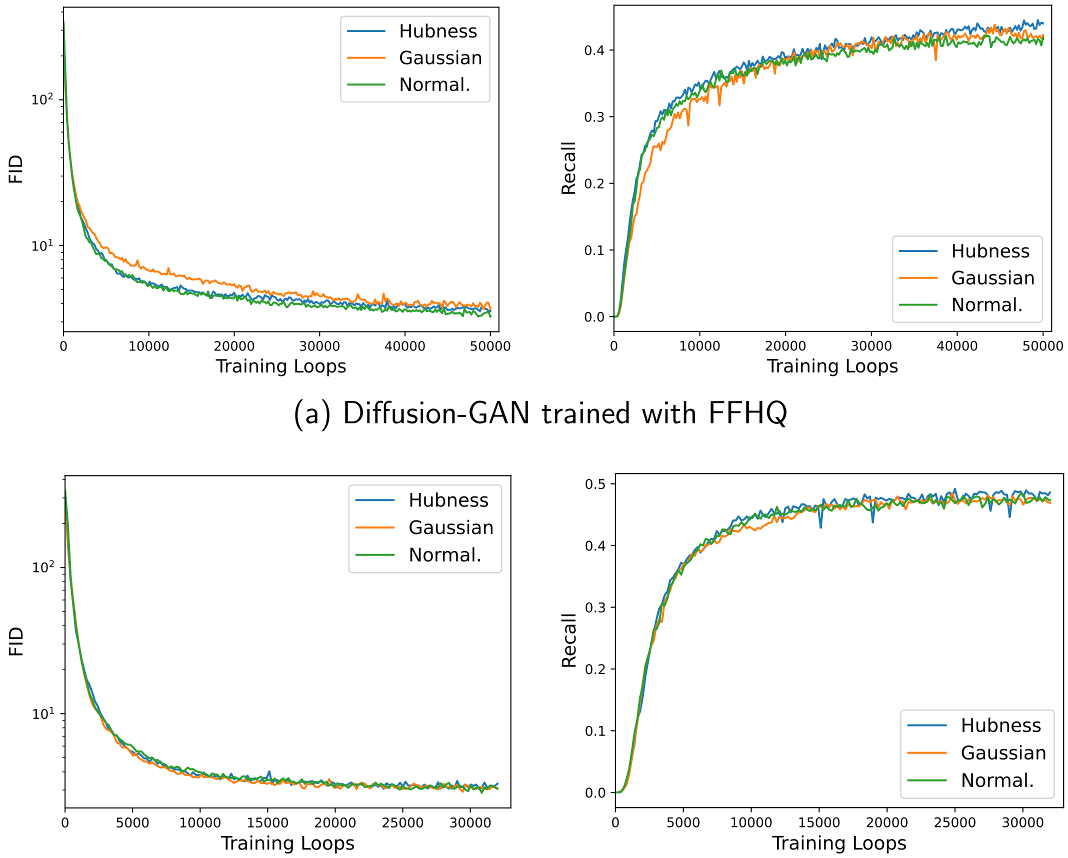


(a) Gaussian Sampling (b) Normal. Sampling (c) Hubness Sampling

(d) Gaussian Sampling (e) Normal. Sampling (f) Hubness Sampling

(g) Gaussian Sampling (h) Normal. Sampling (i) Hubness Sampling

Figure 5.1: The example results of trained on Gaussian sampling, Normalization sampling and Hubness sampling with FFHQ (a,b,c), Lsun-bedroom (d,e,f) and Landscape (g,h,i) datasets. The input latents of these trained models are random, where the red box is to mark the low quality generative images.



(a) Diffusion-GAN trained with FFHQ

(b) Diffusion-GAN trained with Landscape

Figure 5.2: The examples show Diffusion-GAN trained on the FFHQ and Landscape datasets. As demonstrated in the images, the FID scores are comparable across Hubness sampling, Gaussian sampling, and Normalization sampling. However, Hubness sampling consistently outperforms the other two methods in terms of Recall scores, particularly in the tail regions.

5.4.2 Test with different dataset.

Evaluating the quality of GANs generated images is essential, as precision, recall, and FID are frequently used metrics to quantify GAN output. As shown in Table 5.5, Diffusion-GAN [145] was trained on various datasets, including FFHQ, LSUN (Church, Cat, Horse, Bedroom), and Landscape. The results indicate that the proposed sampling method achieves higher recall and improved image quality compared to random sampling. Furthermore, our method demonstrates superior performance, particularly when training on the LSUN-Church, Cat, and Bedroom datasets. These findings suggest that hubness sampling may enhance the diversity of generated images while maintaining a quality level comparable to the baseline normalization sampling approach.

As shown in Table 5.6a and Table 5.6b, we evaluate the performance of different sampling methods on StyleGAN2 [68] and GANFormer [57] using the FFHQ and LSUN datasets (Church and Bedroom). The results indicate that StyleGAN2 achieves similar outcomes to Diffusion-GAN, with hubness sampling yielding the highest recall. Additionally, GANFormer demonstrates improved diversity with hubness sampling, along with better precision and FID scores, signifying that this sampling method enhances both the quality and diversity of the generated images.

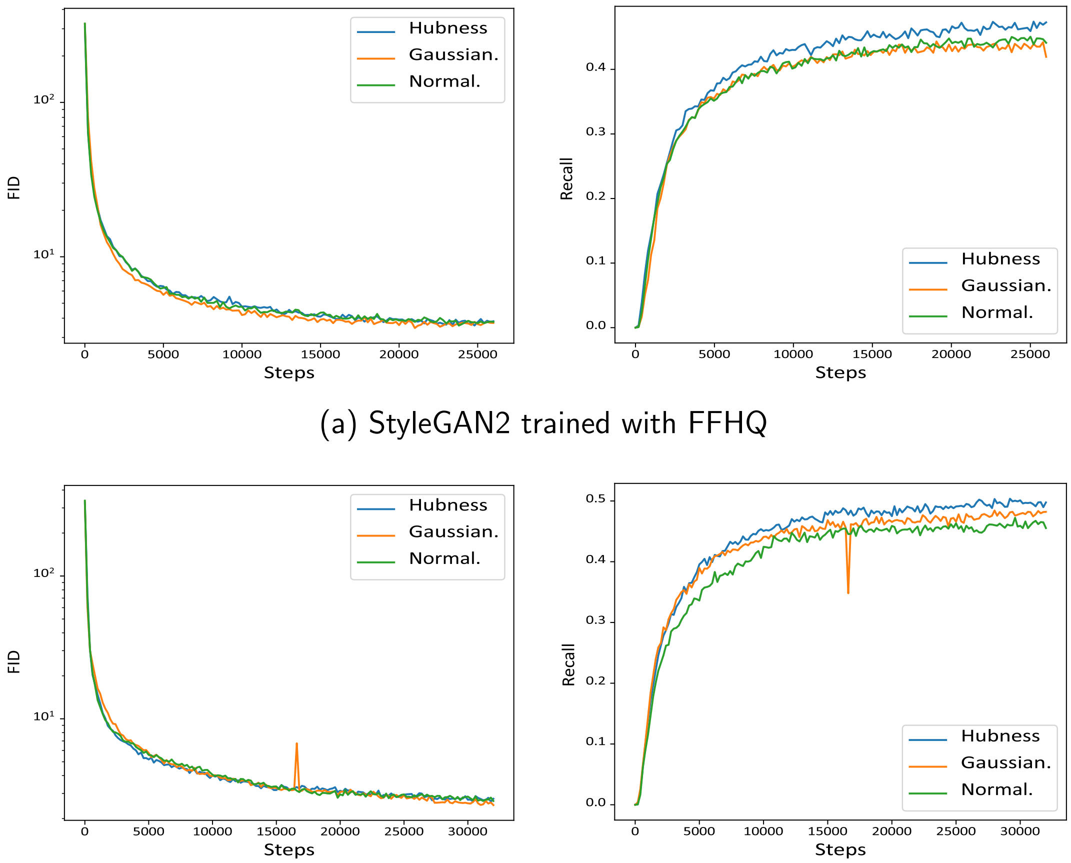
We also evaluate the hubness latents in three GANs trained using hubness sampling. It can be observed that the hubness sampling used for training and testing with hubness latents differs from the random sampling used for training and testing with random latents. Moreover, training and testing with hubness latents through hubness sampling results in higher precision and lower recall. This aligns with the previously established trend: while hubness latents produce high-quality images, they come at the cost of reduced diversity.

5.4.3 Example results from the different dataset

As Fig. 5.1 shows, example training results from the FFHQ, LSUN-Bedroom, and Landscape datasets are displayed, with low-quality images highlighted by red boxes. Additionally, in the case of the Landscape dataset, models trained with Gaussian sampling occasionally generate blank images, also indicated by red boxes. This further underscores the advantage of hubness sampling in generating more reliable and higher-quality outputs.

5.4.4 Evaluating on fairness measurement

To evaluate the fairness of generative models, this study employs the CLEAM score [132] and the Fairness Discrepancy (FD) [133]. The CLEAM score, a Boolean-fairness metric designed for binary demographic attributes (*e.g.*male/female, young/not young), quantifies fairness on a scale where values approaching 0.5 indicate greater parity. The Fairness Discrepancy (FD) metric assesses the deviation between a prede-



(a) StyleGAN2 trained with FFHQ

(b) StyleGAN2 trained with LSUN-Church

Figure 5.3: The examples show StyleGAN2 trained on the FFHQ and Landscape datasets. As demonstrated in the images, the FID scores are comparable across Hubness sampling, Gaussian sampling, and Normalization sampling. However, Hubness sampling consistently outperforms the other two methods in terms of Recall scores, particularly in the tail regions.

fined fairness probability and the expected output of an attribute classifier. Formally, FD $= \|\hat{p} - \mathbb{E}_{x\sim q}[C(x)]\|_2$, where for an observed instance x sampled from the data distribution q, the attribute classifier C yields a probabilistic output $p(x) = C(x)$, and $\hat{p}$ represents the fairness probability vector for the binary feature, typically set as $\hat{p} = [\frac{1}{2}, \frac{1}{2}]$. Table 5.7 and 5.8 present the CLEAM scores and FD values for StyleGAN-2, DiffusionGAN, and GANFormer with respect to gender (male/female) and age (young/not young). The reported results suggest that the application of hubness sampling leads to CLEAM scores closer to the ideal value of 0.5 and achieves the lowest FD, thereby indicating that hubness sampling contributes to improved fairness and a reduction in dataset bias.

Table 5.7: CLEAM score and FD for StyleGAN-2, DiffusionGAN, FormerGAN trained on FFHQ based on gender (male/female).

(a) StyleGAN-2

	CLEAM	FD ↓
Norm.	0.471	0.041
Gauss.	0.478	0.031
Hubness	0.487	0.018

(b) DiffusionGAN

	CLEAM	FD ↓
Norm.	0.486	0.020
Gauss.	0.480	0.028
Hubness	0.492	0.011

(c) FormerGAN

	CLEAM	FD ↓
Norm.	0.473	0.038
Gauss.	0.463	0.052
Hubness	0.478	0.031

Table 5.8: CLEAM score and FD for StyleGAN-2, DiffusionGAN, FormerGAN trained on FFHQ based on age (young/not young).

(a) StyleGAN2

	CLEAM	FD ↓
Norm.	0.889	0.550
Gauss.	0.887	0.547
Hubness	0.873	0.527

(b) DiffusionGAN

	CLEAM	FD ↓
Norm.	0.885	0.544
Gauss.	0.881	0.539
Hubness	0.880	0.537

(c) FormerGAN

	CLEAM	FD ↓
Norm.	0.885	0.544
Gauss.	0.882	0.540
Hubness	0.880	0.537

5.4.5 Models with truncation trick

The truncation trick is a widely adopted technique in high-quality GAN image generation. Consequently, we will dedicate further discussion to its application and impact on GANs trained using our method. As shown in Tables 5.9, 5.11 and 5.10, our sampling method follows the previously established truncation trick. Specifically, as the truncation threshold decreases, diversity decreases while quality improves. Notably, our method achieves the highest diversity (as indicated by the highest recall) across various threshold levels. Meanwhile, Tables 5.9b, 5.11a and 5.10c reveal that our method does not achieve the best FID score with random latents. However, once the truncation threshold reaches 0.9, our method attains the best FID scores. Following that result, it demonstrates that our method enables training with a better balance between diversity and quality.

5.5 Conclusion and Limitation

In conclusion, we present a novel hubness latent sampling method to train the GAN-series models, replacing the traditional Gaussian samplings and the normalization samplings. Under our method, it is significant to improve the diversity of the GAN

Table 5.9: Recall and FID for different truncation trick $(1.0, 0.9, 0.7, 0.5)$ and different dataset on StyleGAN-2. The hubness sampling is helpful to improve the diversity on the different level of the truncation with pre-trained StyleGAN-2 on FFHQ, LSUN-Church and LSUN-Bedroom.

(a) StyleGAN-2 trained with FFHQ testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	FFHQ-1.0		FFHQ-0.9		FFHQ-0.7		FFHQ-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.439	3.583	0.358	5.740	0.203	19.874	0.052	54.013
Gauss.	0.437	3.587	0.375	5.458	0.200	20.408	0.041	53.077
Hubness	**0.470**	**3.568**	**0.415**	**4.456**	**0.245**	**17.588**	**0.063**	**50.998**

(b) StyleGAN-2 trained with Lsun-Church testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	Church-1.0		Church-0.9		Church-0.7		Church-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.467	2.680	0.390	3.601	0.191	9.939	0.034	22.206
Gauss.	0.480	**2.502**	0.411	3.625	0.229	10.880	0.029	24.409
Hubness	**0.497**	2.626	**0.425**	**3.406**	**0.236**	**9.769**	**0.038**	**22.016**

(c) StyleGAN-2 trained with Lsun-Bedroom testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	Bedroom-1.0		Bedroom-0.9		Bedroom-0.7		Bedroom-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.368	2.813	0.327	3.639	0.173	12.059	0.031	31.701
Gauss.	0.353	2.836	0.298	3.896	0.151	12.039	0.024	27.645
Hubness	**0.373**	**2.755**	**0.332**	**3.308**	**0.190**	**9.952**	**0.032**	**24.479**

space and enhance the tail classes to study. To claim the reliability of our method, we did the statistic of the feature-studies situation based on FFHQ and tested our method on different models and dataset, all of the results show hubness latent sampling method can notably decrease the bias to study the feature of the minors.

Although our method effectively enhances minor classes and reduces bias during training, it comes with the drawback of requiring considerable computational time to identify hubness samples. As table 5.12 shown, this limitation highlights the need for further improvements in efficiency. In future work, a more computationally effective approach should be explored to accurately and efficiently capture latent hubness while maintaining the benefits of bias reduction and class enhancement.

Table 5.10: Recall and FID for different truncation trick $(1.0, 0.9, 0.7, 0.5)$ and different dataset (FFHQ, LSUN-Church and LSUN-Bedroom) on FormerGAN. The hubness sampling is helpful to improve the diversity on the different level of the truncation with pretrained FormerGAN on FFHQ, LSUN-Church and LSUN-Bedroom.

(a) FormerGAN trained with FFHQ testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	FFHQ-1.0		FFHQ-0.9		FFHQ-0.7		FFHQ-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.245	9.484	0.196	10.622	0.136	21.101	0.062	53.009
Gauss.	0.272	6.184	0.234	6.455	0.172	12.796	0.078	28.753
Hubness	**0.307**	**5.055**	**0.257**	**5.667**	**0.187**	**11.995**	**0.097**	**27.457**

(b) FormerGAN trained with Lsun-Church testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	Church-1.0		Church-0.9		Church-0.7		Church-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.192	12.771	0.154	13.092	0.103	14.000	0.054	22.718
Gauss.	0.223	7.334	0.182	8.319	0.137	9.588	0.080	18.350
Hubness	**0.293**	**4.040**	**0.250**	**4.289**	**0.182**	**7.729**	**0.104**	**15.667**

(c) FormerGAN trained with Lsun-Bedroom testing on the truncation level to be $1.0, 0.9, 0.7, 0.5$.

	Bedroom-1.0		Bedroom-0.9		Bedroom-0.7		Bedroom-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.225	**4.656**	0.185	5.721	0.120	8.230	0.056	16.398
Gauss.	0.221	5.072	0.179	5.603	0.115	8.216	0.061	16.120
Hubness	**0.254**	5.021	**0.197**	**5.544**	**0.125**	**7.756**	**0.087**	**14.873**

Table 5.11: Recall and FID for different truncation trick $(1.0, 0.9, 0.7, 0.5)$ and different dataset (FFHQ, LSUN-Church and LSUN-Bedroom) on Diffusion-GAN. The hubness sampling is helpful to improve the diversity on the different level of the truncation with pretrained Diffusion-GAN on FFHQ, LSUN-Church and LSUN-Bedroom.

(a) Diffusion-GAN trained with FFHQ testing on the truncation level to be $1.0, 0.9, 0.70.5$.

	FFHQ-1.0		FFHQ-0.9		FFHQ-0.7		FFHQ-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.419	3.280	0.360	5.430	0.189	20.986	0.048	55.503
Gauss.	0.430	3.661	0.389	5.114	0.272	16.726	0.112	48.015
Hubness	0.440	3.539	0.399	3.891	0.285	11.012	0.133	31.651

(b) Diffusion-GAN trained with Lsun-Church testing on the truncation level to be $1.0, 0.9, 0.70.5$.

	Church-1.0		Church-0.9		Church-0.7		Church-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.357	3.950	0.278	5.306	0.111	13.023	0.018	28.079
Gauss.	0.393	4.031	0.316	5.601	0.139	13.616	0.024	30.334
Hubness	**0.400**	**3.659**	**0.326**	4.814	0.150	11.697	**0.028**	23.794

(c) Diffusion-GAN trained with Lsun-Bedroom testing on the truncation level to be $1.0, 0.9, 0.70.5$.

	Bedroom-1.0		Bedroom-0.9		Bedroom-0.7		Bedroom-0.5	
	Recall	FID	Recall	FID	Recall	FID	Recall	FID
Norm.	0.240	4.649	0.042	13.721	0.020	21.030	0.001	34.710
Gauss.	0.130	5.669	0.098	6.595	0.036	14.076	0.006	26.951
Hubness	0.254	4.113	0.211	4.847	0.098	13.089	0.017	24.130

Table 5.12: Hubness sampling was performed by selecting the top $10\%, 15\%, 20\%, 25\%, 30\%$ of latent vectors based on their hubness values. According to Algorithm 3, the number of latent vectors used for training was held constant. Consequently, the original latent space needed to contain $\frac{x}{10\%}, \frac{x}{15\%}, \frac{x}{20\%}, \frac{x}{25\%}, \frac{x}{30\%}$ latent vectors, respectively, to yield the fixed training batch set size x after hubness-based selection.

	Top-10%	Top-15%	Top-20%	Top-25%	Top-30%	Random
Time (CUDA)	5.820s	3.144s	2.215s	1.415s	1.406s	> 10ms

Chapter 6

Conclusions, and future work

This concluding chapter synthesizes the results of the present study, focusing on the theoretical understanding and practical implementation of the hubness phenomenon in the context of generative modeling, particularly within the architecture of Generative Adversarial Networks (GANs). The findings and key takeaways are presented in Section 6.1. Section 6.2 focuses on highlighting the main contributions of this thesis, while recommendations for future research directions are discussed in Section 6.3.

6.1 Summary

This thesis aims to deepen the understanding of the hubness phenomenon in the latent space of generative models, with a focus on improving generative diversity and image quality. In Section 3, we explore how hubness latents can be explored to generate high-quality synthetic images and investigate the relationship between hubness and the truncation trick, offering potential explanations for its effectiveness. In Section 4, we extend the hubness phenomenon to the non-random feature space and demonstrate its application in the precision and recall metrics for generative model assessment, aiming to enhance evaluation efficiency. Finally, we apply hubness sampling to train state-of-the-art GAN models. The training results reveal an intriguing finding: the hubness sampling method not only improves the diversity of GAN-generated outputs but also helps reduce the bias inherent in the dataset. This work offers valuable insights into the use of hubness to optimize the performance of generative models in various tasks.

Overall, this work makes a significant contribution to understanding and leveraging the hubness phenomenon in GANs, offering a novel perspective on high-dimensional latent space properties and their implications for generative model performance.

6.2 Contributions

In this thesis, there are three main contributions, investigating the presence of the hubness phenomenon in generative latent spaces and synthesizing high-quality images by hubness-aware latents; applying hubness vectors improve the efficiency of evaluation metrics in feature spaces; and exploiting hubness-aware sampling techniques to improve generative diversity while mitigating dataset bias, leading to more balanced and robust generative models.

For the hubness phenomenon in generative latent spaces, we analyze the hubness phenomenon with the high-dimensional vectors by hubness value and discuss the possible effect for the generative spaces, and the contributions of this area are summarised as:

- We uncover the existence of *hubness* phenomenon in the GAN latent space, which has a significant correlation with the quality of GAN synthesized images, *i.e.* the proposed *hubness priors* and propose a novel GAN latent sampling algorithm that identifies high-quality *hub* latents based on our *hubness priors*, which allows efficient and high-quality image synthesis for GANs.

For improving the efficiency of precision and recall using hubness features, we analyze redundancy in the computation algorithm and uncover the relationship between the hubness phenomenon and redundancy. The key aspect of this area is:

- We propose *efficient precision and recall* (eP&R) metrics for assessing generative models, which give almost identical results as the original P&R [78] but consume much less time and space. Theoretically, our eP&R run in $O(mn \log n)$ time and consume $O(mn)$ space (m is the number of of hubs samples and $m < n$), which are much more efficient than the original P&R metrics that run in $O(n^2 \log n)$ time and consumes $O(n^2)$ space.

To enhance generative diversity, building upon previous research [84] on distribution-aware sampling for improving GAN training, we propose a novel hubness-aware sampling method. This approach suggests hubness sampling latents can improve the diversity of GAN-generated outputs while mitigating dataset bias. By incorporating hubness-aware sampling into the training process, we aim to achieve a more balanced and representative generative model, ultimately enhancing both the quality and variability of the generated data, as shown in the following.

- We introduce a hubness-aware sampling method for training GAN-based models, demonstrating its effectiveness in enhancing both balance and diversity in the generated outputs and emphasize the crucial role of sampling strategies in GAN training, particularly their impact on latent space density and model performance.

6.3 Future work

The research presented in this thesis has provided valuable insights into the role of the hubness phenomenon in GANs, shedding light on its potential to enhance generative model performance. However, it is still a main challenge to solve the diversity estimation and distribution fitting in high dimensinal space and there are also several works remain in this area, warranting further investigation. Future research directions include:

- Going a step further and investigating the distribution problem in higher dimensional spaces and how this adoption of density imbalances can be used to improve existing models.

- Conducting user studies to assess human preferences for generated images, complementing metric-based evaluations with subjective quality assessmen

- Exploring more effective and efficient methods for identifying true hubness vectors, optimizing their selection for improved generative outcomes.

- Expanding the study of hubness beyond GANs to explore its applicability in other generative models, such as Variational Autoencoders (VAEs) and diffusion models, to further enhance their quality, diversity, and efficiency.

Bibliography

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the styleGAN latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020.

[3] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[4] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings 8*, pages 420–434. Springer, 2001.

[5] Dongsheng An, Jianwen Xie, and Ping Li. Learning deep latent variable models by short-run mcmc inference with optimal transport correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15415–15424, June 2021.

[6] Michael Arbel, Danica J. Sutherland, Mikoł aj Bińkowski, and Arthur Gretton. On gradient regularizers for mmd gans. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative

adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[8] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In *International conference on similarity search and applications*, pages 34–49. Springer, 2017.

[9] Arturs Backurs, Yihe Dong, Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. Scalable nearest neighbor search for optimal transport. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 497–506. PMLR, 13–18 Jul 2020.

[10] Leemon C Baird and A Harry Klopf. Reinforcement learning with high-dimensional, continuous actions. *Wright Laboratory, Wright-Patterson Air Force Base, Tech. Rep. WL-TR-93-1147*, 15, 1993.

[11] Dmitry Baranchuk, Artem Babenko, and Yury Malkov. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 202–216, 2018.

[12] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022.

[13] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the 14th international conference on World Wide Web*, pages 651–660, 2005.

[14] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[15] R. Bellman, R.E. Bellman, and Rand Corporation. *Dynamic Programming.* Rand Corporation research study. Princeton University Press, 1957.

[16] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[17] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

[18] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[19] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[20] Krisztian Buza. Classification of gene expression data: a hubness-aware semi-supervised approach. *Computer methods and programs in biomedicine*, 127:105–113, 2016.

[21] Krisztian Buza. Semi-supervised naive hubness Bayesian k-nearest neighbor for gene expression data. In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, pages 101–110. Springer, 2016.

[22] Krisztián Antal Buza and Júlia Koller. Classification of electroencephalograph data: A hubness-aware approach. *Acta Polytechnica Hungarica*, 13(2):27–46, 2016.

[23] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[24] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19830–19843, 2023.

[25] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1887–1898. PMLR, 13–18 Jul 2020.

[26] Min Jin Chong and David Forsyth. Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079, 2020.

[27] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[29] Adolfo Crespo Márquez. *The Curse of Dimensionality*, pages 67–86. Springer International Publishing, Cham, 2022.

[30] Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard Hovy, and Aaron Courville. Calibrating energy-based generative adversarial networks. In *International Conference on Learning Representations*, 2022.

[31] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[32] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[33] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, volume abs/1605.08803, 2017.

[34] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[35] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.

[36] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[37] Damien François, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.

[38] Felix Friedrich, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Patrick Schramowski, Sasha Luccioni, and Kristian Kersting. Fair diffusion: Instructing text-to-image generation models on fairness. *arXiv preprint arXiv:2302.10893*, 2023.

[39] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *arXiv preprint arXiv:1707.00143*, 2017.

[40] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations*, 2020.

[41] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models

for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[44] Will Sussman Grathwohl, Jacob Jin Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No mcmc for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*.

[45] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[46] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. *Advances in neural information processing systems*, 30, 2017.

[47] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, page 47–57, New York, NY, USA, 1984. Association for Computing Machinery.

[48] Pengxiao Han, Changkun Ye, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. Latent-based diffusion model for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2639–2648, June 2024.

[49] Kazuo Hara, Ikumi Suzuki, Kei Kobayashi, Kenji Fukumizu, and Miloš Radovanović. Flattening the density gradient for eliminating spatial centrality to reduce hubness. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 1659–1665. AAAI Press, 2016.

[50] Ben Harwood and Tom Drummond. Fanng: Fast approximate nearest neighbour graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5713–5722, 2016.

[51] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[52] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[53] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[54] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[55] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[56] John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088–3092, 1984.

[57] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021.

[58] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[59] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[60] Steffen Jung and Margret Keuper. Internalized biases in fréchet inception distance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[61] Hervé Jégou, Romain Tavenard, Matthijs Douze, and Laurent Amsaleg. Searching in one billion vectors: Re-rank with source coding. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 861–864, 2011.

[62] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[63] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[64] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[65] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[66] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 4401–4410, 2019.

[67] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[68] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[69] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[70] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[71] Thomas A Keller, Jorn WT Peters, Priyank Jaini, Emiel Hoogeboom, Patrick Forré, and Max Welling. Self normalizing flows. In *International Conference on Machine Learning*, pages 5378–5387. PMLR, 2021.

[72] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

[73] Yeongmin Kim, Byeonghu Na, Minsang Park, JoonHo Jang, Dongjun Kim, Wanmo Kang, and Il chul Moon. Training unbiased diffusion models from biased dataset. In *The Twelfth International Conference on Learning Representations*, 2024.

[74] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[75] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[76] Flip Korn, B-U Pagel, and Christos Faloutsos. On the" dimensionality curse" and the" self-similarity blessing". *IEEE Transactions on Knowledge and Data Engineering*, 13(1):96–111, 2002.

[77] Konstantinos Koutroumbas and Sergios Theodoridis. *Pattern recognition*. Academic Press, 2008.

[78] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[79] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[80] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[81] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[82] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[83] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1475–1488, 2019.

[84] Yang Li, Yichuan Mo, Liangliang Shi, and Junchi Yan. Improving generative adversarial networks via adversarial learning in latent space. *Advances in neural information processing systems*, 35:8868–8881, 2022.

[85] Yuanbang Liang, Jing Wu, Yu-Kun Lai, and Yipeng Qin. Exploring and exploiting hubness priors for high-quality GAN latent sampling. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13271–13284. PMLR, 17–23 Jul 2022.

[86] King Ip Lin, H. V. Jagadish, and Christos Faloutsos. The tv-tree: an index structure for high-dimensional data. 3(4):517–542, October 1994.

[87] Li Lin, Santosh, Mingyang Wu, Xin Wang, and Shu Hu. Ai-face: A million-scale demographically annotated ai-generated face dataset and fairness benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[88] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[89] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *International Conference on Machine Learning*, 2023.

[90] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1680–1691, 2023.

[91] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. A time-

space efficient locality sensitive hashing method for similarity search in high dimensions. *Technical report, Tech. Rep.*, 2006.

[92] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

[93] Marco Marchesi. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082*, 2017.

[94] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

[95] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*, pages 2391–2400. PMLR, 2017.

[96] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[97] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[98] Charles M Newman and Yosef Rinott. Nearest neighbors and Voronoi volumes in high-dimensional point processes with various distance functions. *Advances in Applied Probability*, 17(4):794–809, 1985.

[99] Charles M Newman, Yosef Rinott, and Amos Tversky. Nearest neighbors and Voronoi regions in certain point processes. *Advances in Applied Probability*, 15(4):726–751, 1983.

[100] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[101] Haechan Noh, Taeho Kim, and Jae-Pil Heo. Product quantizer aware inverted index for scalable nearest neighbor search. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12190–12198, 2021.

[102] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[103] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[104] Yuchen Pu, Weiyao Wang, Ricardo Henao, Liqun Chen, Zhe Gan, Chunyuan Li, and Lawrence Carin. Adversarial symmetric variational autoencoder. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[105] Yipeng Qin, Niloy Mitra, and Peter Wonka. How does Lipschitz regularization influence GAN training? In *European Conference on Computer Vision*, pages 310–326. Springer, 2020.

[106] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[107] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept):2487–2531, 2010.

[108] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[109] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[110] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[111] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021.

[112] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[113] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and BjÃ¶rn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[114] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[115] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[116] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

[117] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

[118] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3:1–3:9, 2019.

[119] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

[120] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. volume abs/2201.00273, 2022.

[121] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[122] Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Ridge regression, hubness, and zero-shot learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*, pages 135–151. Springer, 2015.

[123] Larissa C Shimomura, Rafael Seidi Oyamada, Marcos R Vieira, and Daniel S Kaster. A survey on graph-based methods for similarity searches in metric spaces. *Information Systems*, 95:101507, 2021.

[124] Loïc Simon, Ryan Webster, and Julien Rabin. Revisiting precision and recall definition for generative model evaluation. *arXiv preprint arXiv:1905.05441*, 2019.

[125] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. *arXiv preprint arXiv:2104.06954*, 2021.

[126] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022.

[127] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[128] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[129] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[130] Yufei Tao, Jimeng Sun, and Dimitris Papadias. Analysis of predictive spatio-temporal queries. *ACM Transactions on Database Systems (TODS)*, 28(4):295–336, 2003.

[131] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

[132] Christopher Teo, Milad Abdollahzadeh, and Ngai-Man (Man) Cheung. On measuring fairness in generative models. In *Advances in Neural Information Processing Systems*, volume 36, pages 10644–10656. Curran Associates, Inc., 2023.

[133] Christopher T. H. Teo and Ngai-Man Cheung. Measuring fairness in generative models. *ArXiv*, abs/2107.07754, 2021.

[134] Christopher TH Teo, Milad Abdollahzadeh, and Ngai-Man Cheung. Fair generative models via transfer learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2429–2437, 2023.

[135] Nenad Tomašev, Krisztian Buza, Kristóf Marussy, and Piroska B Kis. Hubness-aware classification, instance selection and feature construction: Survey and extensions to time-series. In *Feature selection for data and pattern recognition*, pages 231–262. Springer, 2015.

[136] Nenad Tomašev and Dunja Mladenić. Hub co-occurrence modeling for robust high-dimensional kNN classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 643–659. Springer, 2013.

[137] Nenad Tomašev, Miloš Radovanović, Dunja Mladenić, and Mirjana Ivanović. Hubness-based fuzzy measures for high-dimensional k-nearest neighbor classification. *International Journal of Machine Learning and Cybernetics*, 5(3):445–458, 2014.

[138] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

[139] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.

[140] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[141] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[142] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[143] Jinqi Wang, Yunfei Fu, Zhangcan Ding, Bailin Deng, Yu-Kun Lai, and Yipeng Qin. Training-free editioning of text-to-image models. *arXiv preprint arXiv:2405.17069*, 2024.

[144] Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *arXiv preprint arXiv:2101.12631*, 2021.

[145] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[146] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and

performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.

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

[148] Yankun Wu, Yuta Nakashima, and Noa Garcia. Stable diffusion exposed: Gender bias from prompt to image. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 1648–1659, 2024.

[149] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13095–13105, 2023.

[150] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *International Conference on Learning Representations*, 2022.

[151] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE international conference on big data (big data)*, pages 570–575. IEEE, 2018.

[152] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023.

[153] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18938–18949, 2023.

[154] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[155] Lantao Yu, Yang Song, Jiaming Song, and Stefano Ermon. Training deep energy-based models with f-divergence minimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10957–10967. PMLR, 13–18 Jul 2020.

[156] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 377–393. Springer, 2020.

[157] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. Vaegan: A collaborative filtering framework based on adversarial variational autoencoders. In *IJCAI*, volume 19, pages 4206–4212, 2019.

[158] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36, 2024.

[159] Cheng Zhang, Xuanbai Chen, Siqi Chai, Henry Chen Wu, Dmitry Lagun, Thabo Beeler, and Fernando De la Torre. ITI-GEN: Inclusive text-to-image generation. In *ICCV*, 2023.

[160] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030, 2017.

[161] Peitian Zhang, Zheng Liu, Shitao Xiao, Zhicheng Dou, and Jing Yao. Hybrid inverted index is a robust accelerator for dense retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1877–1888, 2023.

[162] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6027–6037, 2023.

[163] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *International Conference on Learning Representations*, 2022.

[164] Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, and Zhihua Zhang. Lipschitz generative adversarial nets. In *International Conference on Machine Learning*, pages 7584–7593. PMLR, 2019.

[165] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[166] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.

[167] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1219–1229, June 2023.