A Deep Bayesian Nonparametric Framework for Robust Mutual Information Estimation

Anonymous authors
Paper under double-blind review

Abstract

Mutual Information (MI) is a crucial measure for capturing dependencies between variables, yet exact computation is challenging in high dimensions with intractable likelihoods, impacting accuracy and robustness. One idea is to use an auxiliary neural network to train an MI estimator; however, methods based on the empirical distribution function (EDF) can introduce sharp fluctuations in the MI loss due to poor out-of-sample performance, thereby destabilizing convergence. We present a Bayesian nonparametric (BNP) solution for training an MI estimator by constructing the MI loss with a finite representation of the Dirichlet process posterior to incorporate regularization in the training process. With this regularization, the MI loss integrates both prior knowledge and empirical data to reduce the loss sensitivity to fluctuations and outliers in the sample data, particularly in small sample settings like mini-batches. This approach addresses the challenge of balancing accuracy and low variance by effectively reducing variance, leading to stabilized and robust MI loss gradients during training and enhancing the convergence of the MI approximation while offering stronger theoretical guarantees for convergence. We explore the application of our estimator in maximizing MI between the data space and the latent space of a variational autoencoder. Experimental results demonstrate significant improvements in convergence over EDF-based methods, with applications across synthetic and real datasets, notably in 3D CT image generation—yielding enhanced structure discovery and reduced overfitting in data synthesis. While this paper focuses on generative models in application, the proposed estimator is not restricted to this setting and can be applied more broadly in various BNP learning procedures.

1 Introduction

Mutual information is a statistical measure used to quantify the dependency between random variables with wide applications in machine learning. It plays a crucial role in tasks such as feature selection, representation learning, clustering, and generative modeling. By capturing both linear and nonlinear dependencies, mutual information serves as a versatile measure in understanding and leveraging relationships in complex datasets. However, computing MI in these applications faces significant challenges, particularly in the presence of high-dimensional or complex data structures. Mutual information is defined using the joint distribution of two random variables and the product of their marginal distributions, making its estimation especially difficult when these distributions are analytically intractable. While neural network-based estimators have shown promise in addressing some of these issues (Belghazi et al., 2018; Hjelm et al., 2018), they remain prone to instability and sensitivity to sample variability and outliers, highlighting the need for more flexible and reliable approaches for mutual information estimation in machine learning tasks.

Recently, Bayesian nonparametric (BNP) techniques have proven effective in introducing uncertainty into modeling data distributions with minimal assumptions, leading to enhanced flexibility, robustness, and reduced sensitivity to sample variability in the machine learning community (Fazeli-Asl et al., 2024; Dellaporta et al., 2022; Fazeli-Asl & Zhang, 2023; Bariletto & Ho, 2024). However, BNP approaches for estimating mutual information have received comparatively little attention. In this paper, we introduce a Dirichlet process

(DP) prior on the data distribution and propose a deep BNP mutual information neural estimation (DPMINE) between two variables. This estimator serves as the Bayesian counterpart to the Kullback-Leibler (KL)-based and Jensen-Shannon (JS)-based MINE methods introduced in Belghazi et al. (2018) and Hjelm et al. (2018), respectively. The MINE is computed by training an auxiliary function, parameterized by a neural network, that approximates the intractable likelihoods of the joint distribution of the two variables and the product of their marginal distributions.

Our DP-based approach in the proposed estimator employs a finite DP posterior representation in the loss function of the auxiliary function, making our estimator more robust to sample variability. It introduces distributional regularization through the incorporation of a prior distribution, thereby improving the efficiency and stability of the optimization process. This approach improves estimation stability by enhancing out-of-sample performance, addressing the curse of dimensionality and sensitivity to sample size, and providing robustness guarantees in high-dimensional settings. This makes it well-suited for integration into deep learning models, where a procedure with faster and robust convergence is needed for training the relevant cost functions regularized by this estimator. In contrast, frequentist methods, which rely on the empirical distribution, may exhibit greater sensitivity to sample variability, potentially leading to less stable gradient computations during training, particularly with small batch sizes or when dealing with highly heterogeneous data. This sensitivity can be interpreted as introducing variance in the estimation process. We present experiments comparing our DP approach to EDF-based methods (frequentist approaches) in mutual information estimation, demonstrating improved performance. Furthermore, our experiments highlight the robustness of the DPMINE in high-dimensional settings, showing better convergence and reduced estimation fluctuations. In our results, we demonstrate that the DPMINE is theoretically stronger than MINE counterpart, particularly for the KL-based estimator.

To further motivate the need for our KL-based Bayesian counterpart, we briefly highlight the limitations of existing MI estimators. JS-based estimators ensure stable convergence but often lack the precision needed to capture complex dependencies, while traditional KL-based estimators provide greater accuracy but suffer from poor convergence, limiting their practical use (Song & Ermon, 2020; Poole et al., 2019). Our KL-based Bayesian counterpart addresses these issues by combining the accuracy of KL with improved convergence properties. While its broader impact on tasks such as information bottleneck methods is beyond the scope of this paper, we focus specifically on its effect in improving the performance of a particular family of generative models where representation learning is essential.

We apply our proposed DPMINE to enhance the performance of a BNP VAE-GAN model, a fusion of Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) (Fazeli-Asl & Zhang, 2023). A VAE-GAN model effectively addresses the persistent challenges of mode collapse in GANs and the generation of blurry images in VAEs, striking a balance between diversity and quality in the generated data (Larsen et al., 2016; Donahue et al., 2016; Dumoulin et al., 2016; Rosca et al., 2017; Kwon et al., 2019). This integration combines the high-quality outputs that GANs offer and the diversity of representations that VAEs offer, resulting in a superior generative model. However, to fully unlock the potential of VAE-GAN models, further exploration of the information encoded in the latent space is needed (Belghazi et al., 2018; Hjelm et al., 2018). We aim to maximize the information between original and encoded samples, as well as between generated and encoded samples using our DPMINE. This incorporation of DPMINE in BNP generative modeling improves representation in the code space of the generative model and also helps preserve the information in the code space for generating new samples. Although this paper focuses on DPMINE's application in generative models, we emphasize that its use is not limited to such models and opens new opportunities for researchers interested in applying this estimator to various optimization problems within the BNP learning (BNPL) framework (Lyddon et al., 2018; 2019; Fong et al., 2019).

The paper is structured as follows: Section 2 reviews the basic concepts in mutual information inference, as well as its applications in generative modeling. In Section 3, we present the fundamental concepts of the DP and introduce the DPMINE, along with theoretical results on the properties of our estimator. We then demonstrate the integration of DPMINE into the 3D convolutional architecture of a BNP VAE-GAN model. Section 4 explores the impact of the DPMINE, first demonstrating its improved performance over EDF-based methods in mutual information estimation, with greater robustness in high-dimensional settings. We then apply the DPMINE to train the VAE-GAN model through various experiments, showcasing its

ability to mitigate mode collapse. In particular, we evaluate the proposed procedure using a collection of 3-dimensional (3D) chest CT scans of COVID-19-infected patients. Finally, we conclude the paper in Section 5, where we discuss potential future research directions. All technical proofs, additional experimental details, limitation broad impacts, and safeguards are provided in the Appendix.

2 Fundamentals of Mutual Information Inference and Applications in Generative Modeling

In this section, we introduce the background information on MI and its applications in generative modeling.

2.1 Mutual information

Let $(\mathbf{X}, \mathbf{Y})$ be a pair of random variables with a joint cumulative distribution function (CDF) $F_{\mathbf{X},\mathbf{Y}}$ and marginal CDFs $F_{\mathbf{X}}$ and $F_{\mathbf{Y}}$, defined over the product space $\mathcal{X} \otimes \mathcal{Y}$. The mutual information between $\mathbf{X}$ and $\mathbf{Y}$ is given by:

$$\mathrm{MI}(\mathbf{X}, \mathbf{Y}) = \mathrm{D}_{\mathrm{KL}}(F_{\mathbf{X}\otimes\mathbf{Y}}, F_{\mathbf{X}} \otimes F_{\mathbf{Y}}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}), \tag{1}$$

where $\mathrm{D}_{\mathrm{KL}}(\cdot, \cdot)$ denotes the KL divergence, and $H(\cdot)$ and $H(\cdot|\cdot)$ represent marginal and conditional Shannon entropies, respectively.

Computing Eq. equation 1 can be challenging in many practical applications due to the intractable likelihood of the relevant densities. There are approximate techniques to calculate mutual information. However, many of them suffer from the curse of dimensionality, like in examples where k-nearest neighbors are used to estimate MI. (Al-Labadi et al., 2022; Berrett & Samworth, 2019). Moreover, optimizing k to balance the bias-variance trade-off adds to the complexity (Sugiyama, 2012).

An effective approach for overcoming these obstacles involves the establishment of a variational lower bound (VLB) approximation of the MI through the MINE. For instance, Belghazi et al. (2018) uses the Donsker-Varadhan (DV) representation (Donsker & Varadhan, 1983) of the KL divergence to form a lower bound of equation 1 as:

$$\mathcal{L}^{\mathrm{DV}}_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{F_{\mathbf{XY}}}[T_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{Y})] - \ln \mathbb{E}_{F_{\mathbf{X}}\otimes F_{\mathbf{Y}}}[e^{T_{\boldsymbol{\gamma}}(\mathbf{X},\mathbf{Y})}], \tag{2}$$

where $\{T_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}$ be a set of continuous functions parameterized by a neural network on a compact domain Γ that maps $\mathcal{X}\otimes\mathcal{Y}$ to $\mathbb{R}$ and $\ln(\cdot)$ denotes the natural logarithm. The DV-MINE is then defined as $\mathcal{L}^{\mathrm{DV}}_{\widehat{\boldsymbol{\gamma}}}(\mathbf{X}, \mathbf{Y})$, where $\widehat{\boldsymbol{\gamma}}$ is the optimal parameter that maximizes the VLB equation 2. $T_{\boldsymbol{\gamma}}$, which is an auxiliary statistic, serves as the fundamental component in this procedure, which learns to differentiate between a sample from $F_{\mathbf{XY}}$ and $F_{\mathbf{X}} \otimes F_{\mathbf{Y}}$. In practice, $\mathcal{L}^{\mathrm{DV}}_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{Y})$ in Eq. equation 2 is approximated using the empirical cumulative distribution functions (ECDFs), $F_{\mathbf{X}_{1:n}} := \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{X}_i}$ and $F_{\mathbf{Y}_{1:n}} := \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{Y}_i}$:

$$\mathcal{L}^{\mathrm{DV}}_{\boldsymbol{\gamma}}(\mathbf{X}_{1:n}, \mathbf{Y}_{1:n}) = \sum_{\ell=1}^{n} \frac{1}{n} T_{\boldsymbol{\gamma}}(\mathbf{X}_\ell, \mathbf{Y}_\ell) - \ln \sum_{\ell=1}^{n} \frac{1}{n} e^{T_{\boldsymbol{\gamma}}(\mathbf{X}_\ell, \mathbf{Y}_{\pi(\ell)})}, \tag{3}$$

where $\{\pi(1:n)\}$ denotes a random permutation of $\{1:n\}$, a standard technique to empirically approximate $\mathbb{E}_{F_{\mathbf{X}}\otimes F_{\mathbf{Y}}}(\cdot)$ in Eq. equation 2 (Belghazi et al., 2018; Hjelm et al., 2018).

Belghazi et al. (2018) also considered a divergence representation given in Nguyen et al. (2010) to form another lower bound for Eq. equation 1. However, they discovered that Eq. equation 2 yields a tighter bound than this alternative lower bound and demonstrated numerically that DV-MINE outperforms MINE based on the alternative lower bound. Another MINE, based on providing a lower bound for the JS divergence (Nowozin et al., 2016) between $F_{\mathbf{XY}}$ and $F_{\mathbf{X}} \otimes F_{\mathbf{Y}}$, can also be found in the literature (Hjelm et al., 2018; Jones et al., 2023). It is defined as

$$\mathcal{L}^{\mathrm{JS}}_{\gamma}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{F_{\mathbf{XY}}}[-\zeta(-T_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{Y}))] - \mathbb{E}_{F_{\mathbf{X}}\otimes F_{\mathbf{Y}}}[\zeta(T_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{Y}))],$$

where $\zeta(\cdot) = \ln(1 + \exp(\cdot))$ denotes the softplus function. However, Jones et al. (2023) remarked that JS-MINE may be more appropriate for problems involving the maximization of MI, rather than a precise approximation.

A detailed exploration and comparative assessment of different variational bounds for MI is provided in Poole et al. (2019); Tsai et al. (2021), with much of this work driven by the goal of optimizing MI in deep learning problems (Belghazi et al., 2018; Hjelm et al., 2018; Barber & Agakov, 2004; Lin et al., 2022; Chen et al., 2016). However, recent studies have shown that existing neural MI estimators, whether JS-based (providing stable but imprecise estimates) or KL-based (offering accuracy but poor convergence), struggle to provide accurate and low-variance estimates when the true MI is high, highlighting that stability alone in JS-based estimators and accuracy alone in KL-based estimators are insufficient for effectively capturing information (Song & Ermon, 2020). This limitation can impair the regularization of the objective function in deep learning models, including the encoding-decoding process in VAE-based models. Attempts to stabilize KL-based estimators through various modifications of the DV lower bound to reduce variance while preserving accuracy have been proposed (Mroueh et al., 2021; Wen et al., 2020; Guo et al., 2022). However, these modifications often increase computational complexity and make the training process more difficult to control.

2.2 Interactions between generative models and mutual information

Generative models have been extensively explored in data augmentation and synthesis (Mendes et al., 2023; Menon et al., 2023; Toda et al., 2021; Wolterink et al., 2017; Bu et al., 2021). However, these models are expanded beyond the standard GAN loss function (Goodfellow et al., 2014), which could potentially introduce challenges such as mode collapse–memorizing certain modes of data distribution while overlooking other diversities–and training instability. The standard GAN is constructed based on two sets of functions: $\{G_{\boldsymbol{\omega}}\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ and $\{D_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}$, which are parameterized by neural networks and referred to as the generator and discriminator, respectively (Goodfellow et al., 2014). The generator engages in a game with the discriminator, aiming to deceive it into being unable to distinguish between generated samples and real samples. It is achieved by updating parameters as follows:

$$(\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\Omega}} \max_{\boldsymbol{\Theta}} \mathbb{E}_{F_{\boldsymbol{\xi}}}[\ln(1 - D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi})))] + \mathbb{E}_{F}[\ln(D_{\boldsymbol{\theta}}(\mathbf{X}))].$$

Here, $\mathbf{X} \in \mathbb{R}^d \sim F$ represents a real dataset, and $\boldsymbol{\xi} \in \mathbb{R}^p \sim F_{\boldsymbol{\xi}}$ represents a noise vector, where $d > p$. There are various approaches to address the mode collapse issue, including modifying the loss function (Arjovsky et al., 2017; Nowozin et al., 2016; Li et al., 2015; Dellaporta et al., 2022; Fazeli-Asl et al., 2024) and exploring different architectures (Radford et al., 2015; Zhang et al., 2019).

In contrast, an effective strategy is to employ VAE-GAN models (Larsen et al., 2016). The VAE-GAN employs an encoder network in the VAE component to learn a probabilistic encoding of $\mathbf{X}$ into a compact representation $\boldsymbol{c}$. Subsequently, a decoder is used to generate realistic data from $\boldsymbol{c}$ (Kingma & Welling, 2013). The probabilistic approach involves modeling the variational distribution $F_{E_{\boldsymbol{\eta}}}(\boldsymbol{c}|\mathbf{X})$ using the encoder network $\{E_{\boldsymbol{\eta}}\}_{\boldsymbol{\eta}\in\boldsymbol{\mathcal{H}}}$, which approximates the true posterior distribution of $\boldsymbol{c}$ given $\mathbf{X}$. The VAE aims to minimize the error in reconstructing the original input from the code space, as well as the regularization error–the difference between $F_{E_{\boldsymbol{\eta}}}(\boldsymbol{c}|\mathbf{X})$ and $F_{\mathbf{z}}$.

The traditional VAE-GAN model comprises of two interconnected networks $E_{\boldsymbol{\eta}}$, $G_{\boldsymbol{\omega}}$, and $D_{\boldsymbol{\theta}}$. The VAE is connected to the GAN through $G_{\boldsymbol{\omega}}$, which acts as a decoder for the VAE. The generator is fed with $(\boldsymbol{\xi}, \boldsymbol{c})$ to generate samples that aim to fool $D_{\boldsymbol{\theta}}$ into distinguishing them from $\mathbf{X}$. Some notable works that employ this strategy include the bi-directional GAN (BiGAN) (Donahue et al., 2016; Dumoulin et al., 2016), the α-GAN (Rosca et al., 2017), the α-WGAN+Gradient penalty (α-WGAN+GP) (Kwon et al., 2019), and the BNPWMMD-GAN (Fazeli-Asl & Zhang, 2023). The main differences between these models lie in their loss functions and network architectures.

Particularly, the α-WGAN+GP serves as the primary frequentist nonparametric (FNP) counterpart to the BNPWMMD, though BNPWMMD has recently proven to be a stronger competitor. It combines α-GAN with WGAN+GP to improve training stability in generating 3D brain MRI data, including sagittal, coronal, and axial representations. Jafari et al. (2023) enhanced the α-WGAN+GP to generate connected 3D volumes, while Ferreira et al. (2022) proposed another enhanced model for generating 3D rat brain MRI

using convolutional data manipulation techniques including zoom, rotation, Gaussian noise, flip, translation and scaling intensity. However, the model may learn to expect these transformations, resulting in generated data that reflects the augmented relationships rather than true variability (correlated outputs). Additionally, the provided results did not show a significant visual improvement.

Alongside advancements in VAE-GAN models for generating samples, it is essential to explore the information encoded in the latent space to fully unlock the potential of these models. Maximizing the MI between the input and output of the encoder can significantly reduce the reconstruction error in VAE-based models (Belghazi et al., 2018; Hjelm et al., 2018). This approach encourages the encoder to retain important details and structure in the input data, ultimately leading to a more informative code representation for the decoder to use. Belghazi et al. (2018) incorporated $\max_{\boldsymbol{\Gamma},\boldsymbol{\mathcal{H}}} \mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(\mathbf{X}, E_{\boldsymbol{\eta}}(\mathbf{X}))$ into the loss function of a BiGAN model to estimate and maximize the MI between $\mathbf{X}$ and $E_{\boldsymbol{\eta}}(\mathbf{X})$. This approach, referred to as BiGAN+MINE model, demonstrated stronger coverage on the training dataset than the original BiGAN, displaying the ability of the MMI procedure to reduce mode collapse.

Moreover, preserving the information contained in codes during the generation process is extremely important. For instance, Chen et al. (2016) attempted to maximize the MI between code $E_{\boldsymbol{\eta}}(\mathbf{X})$ and generated sample $G_{\boldsymbol{\omega}}(\boldsymbol{\xi}, \boldsymbol{E_{\eta}(\mathbf{X})})$ by providing another VLB for standard GANs under appropriate regularity conditions. However, no results were provided to declare the effectiveness of their method in mitigating mode collapse. Alternatively, Belghazi et al. (2018) employed DV representation to maximize $MI(E_{\boldsymbol{\eta}}(\mathbf{X}), G_{\boldsymbol{\omega}}(\boldsymbol{\xi}, E_{\boldsymbol{\eta}}(\mathbf{X})))$ for standard GANs to demonstrate the effectiveness of the MINE-based method in covering the training data for this scenario.

3 A DP-based accurate-stable MI estimator and its application in generative model regularization

Our proposed method of deep mutual information estimation partially relies on the DP as a technique to enhance the robustness of the training process, which we will first introduce here. The proposed approach is easy to implement and, particularly for KL-based estimators, provides a tighter lower bound, ultimately improving the stability of the estimator, which, as demonstrated in the numerical results section, leads to variance reduction.

3.1 Dirichlet process

The DP is an infinite generalization of the Dirichlet distribution that is considered on the sample space denoted as $\mathfrak{X}$, which possesses a σ-algebra $\mathcal{A}$ comprising subsets of $\mathfrak{X}$ (Ferguson, 1973). F follows a DP with parameters (a, H) with the notation $F^{\mathrm{Pri}} := (F \sim DP(a, H))$, if for any measurable partition $A_1, \ldots, A_k$ of $\mathfrak{X}$ with $k \geq 2$, the joint distribution of the vector $(F(A_1), \ldots, F(A_k))$ follows a Dirichlet distribution characterized by parameters $(aH(A_1), \ldots, aH(A_k))$. Moreover, it is assumed that $H(A_j) = 0$ implies $F(A_j) = 0$ with probability one. The base measure H captures the prior knowledge regarding the data distribution, while a signifies the strength or intensity of this knowledge.

As a conjugate prior, the posterior distribution of F also follows a DP, denoted by $F^{\mathrm{Pos}} := (F|\mathbf{X}_{1:n} \sim \mathrm{DP}(a+n, H^*))$, for n independent and identically distributed (IID) draws, $\left(\mathbf{X}_{1:n} \in \mathbb{R}^d\right)$, from the random probability measure F where $H^* = a(a+n)^{-1}H + n(a+n)^{-1}F_{\mathbf{X}_{1:n}}$, and $F_{\mathbf{X}_{1:n}}$ represents the empirical cumulative distribution function of the sample $\mathbf{X}_{1:n}$.

Although the stick-breaking representation is a commonly employed series representation for DP inference (Sethuraman, 1994), it lacks the necessary normalization terms to convert it into a probability measure (Zarepour & Al-Labadi, 2012). Additionally, simulating from an infinite series is only feasible through using a random truncation approach to handle the terms within the series. To address these limitations, Ishwaran & Zarepour (2002) introduced an approximation of the DP in the form of a finite series, which allows for

convenient simulation. In the context of posterior inference, this approximation is given by

$$F_N^{\text{Pos}} := \sum_{i=1}^{N} J_{i,N}^{\text{Pos}} \delta_{\mathbf{X}_i^{\text{Pos}}}, \tag{4}$$

where $\left(J_{1:N,N}^{\text{Pos}}\right) \sim \text{Dirichlet}((a+n)/N, \ldots, (a+n)/N)$, $\left(\mathbf{X}_{1:N}^{\text{Pos}}\right) \overset{\text{IID}}{\sim} H^*$, and $\delta_{\mathbf{X}^{\text{Pos}}}$ is the Dirac delta measure. In this study, the variables $J_{i,N}^{\text{Pos}}$ and $\mathbf{X}_i^{\text{Pos}}$ represent the DP's weight and location, respectively. The sequence $(F_N^{\text{Pos}})_{N\geq 1}$ converges in distribution to F^{Pos}, where F_N^{Pos} and F^{Pos} are random values in $M_1(\mathbb{R}^d)$, the space of probability measures on $\mathbb{R}^d$ endowed with the topology of weak convergence (Ishwaran & Zarepour, 2002). In the subsequent sections, we investigate the efficacy of this approximation within a regularization method in a BNP generative model.

Remark 1. *The prior H in the DP introduces a regularizing effect by smoothing the resulting distribution. Unlike the empirical distribution, which represents the data exactly as observed, the DP with a prior creates a distribution that accounts for both observed data and the prior knowledge. This makes the model less sensitive to fluctuations and variability in the sample data, which can be particularly beneficial for optimization processes in models that rely on loss functions (Bariletto & Ho, 2024). It does so by anchoring the sampled distribution around H, especially when data is sparse or when there are regions of high variability. Choosing an effective prior H for regularization depends on the specific research context and the statistician's knowledge of the subject area. For instance, in genetics, H might reflect biological baselines, while in financial modeling, a prior could incorporate historical market trends. Each choice aligns with the research goals, capturing nuances relevant to the field. A commonly effective choice for general purposes is a multivariate normal distribution with parameters reflecting the sample's mean and covariance. This prior helps anchor the distribution near expected values, but with flexibility that accounts for sample variability, thus providing a stabilizing regularization effect across diverse data points.*

Remark 2. *The DP framework balances between adhering to H and adapting to the data through concentration parameter a. This balance prevents overfitting to the specific data sample and provides a more stable and generalized distribution, especially in small sample scenarios Fong et al. (2019); Lyddon et al. (2018). The DP framework strikes a balance between adhering to H and adapting to the data through the concentration parameter a. This balance mitigates overfitting to specific data samples and ensures a more stable and generalized distribution, particularly in small sample scenarios. In this paper, we follow Fazeli-Asl & Zhang (2023) to employ a Maximum A Posteriori (MAP) estimate for selecting the optimal value of the concentration parameter a. This is achieved by maximizing the log-likelihood of F^{pos}, fitted to the given dataset, over a range of a values.*

Remark 3. *In approximating the DP in Eq. equation 4, it is crucial to determine an optimal number of terms to ensure that the truncated series remains a close representation of the true Dirichlet Process. We address this requirement by applying a random stopping rule as outlined in Zarepour & Al-Labadi (2012). This adaptive rule terminates the approximation when the marginal contribution of the current term falls below a specified significance level relative to the cumulative sum of previous terms. Given a predefined threshold $\epsilon \in (0,1)$, the stopping criterion is defined by: $N = \inf\left\{j :, \frac{\Gamma_{j,j}}{\sum_{i=1}^{j}\Gamma_{i,j}} < \epsilon\right\}$. This approach effectively balances computational efficiency with approximation accuracy.*

3.2 DPMINE: The Dirichlet process mutual information neural estimator

In this section, the DP approximation equation 4 is leveraged to develop two lower bounds for MI, leading to the formulation of DPMINE.

KL-based representation: Let $(\mathbf{X}_{1:n})$ be n IID random variables for $\mathbf{X} \in \mathbb{R}^d \sim F$. For continuous functions $f_i : \mathbb{R}^d \to \mathbb{R}^{d'}$, $i = 1, 2$, let $\mathbf{X}_i' = f_i(\mathbf{X})$. Then, for a fixed value of a and a chosen probability measure H used in the DP approximation equation 4, we propose to estimate the DV lower bound of

$\mathrm{MI}(\mathbf{X}_1', \mathbf{X}_2')$ as:

$$\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) \quad := \quad \sum_{\ell=1}^{N} J_{\ell,N}^{\mathrm{Pos}} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\mathrm{Pos}}), f_2(\mathbf{X}_{\ell}^{\mathrm{Pos}})) \quad - \quad \ln \sum_{\ell=1}^{N} J_{\ell,N}^{\mathrm{Pos}} e^{T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\mathrm{Pos}}), f_2(\mathbf{X}_{\pi(\ell)}^{\mathrm{Pos}}))}. \quad (5)$$

Here, $\{\pi(1:N)\}$ denotes a random permutation of $\{1:N\}$, which is a common technique used to empirically approximate $\mathbb{E}_{F_{\mathbf{X}} \otimes F_{\mathbf{Y}}}(\cdot)$ in equation 2 (Belghazi et al., 2018; Hjelm et al., 2018).

JS-based representation: The JS-based estimation is similarly proposed as:

$$\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DPJS}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) := \sum_{\ell=1}^{N} J_{\ell,N}^{\mathrm{Pos}} \Big[-\zeta(-T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\mathrm{Pos}}), f_2(\mathbf{X}_{\ell}^{\mathrm{Pos}}))) - \zeta(T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\mathrm{Pos}}), f_2(\mathbf{X}_{\pi(\ell)}^{\mathrm{Pos}}))) \Big]. \quad (6)$$

Let i be a label in $\{\mathrm{DV}, \mathrm{JS}\}$ indicating the type of the VLB. The BNP MINEs are consequently considered as:

$$\mathrm{MI}_{\mathrm{DPi}}\big(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}),\, f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})\big) = \max_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} \mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DPi}}\big(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}),\, f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})\big). \quad (7)$$

The following theorem states that, at least in the case of KL-based MINE, $\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DPDV}}(\mathbf{X}_1^{\mathrm{Pos}'}, \mathbf{X}_2^{\mathrm{Pos}'})$ is asymptotically larger than $\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2')$ on average. This results in a tighter VLB on the true MI and allows for a more accurate estimation of MI by maximizing the tighter bound over a compact domain $\boldsymbol{\Gamma}$.

Theorem 1 (Limiting expectation)**.** *Considering DP posterior representations defined in equation 15 and equation 6. Given the DP posterior approximation in equation 4, we have,*

i. $\lim_{n,N\to\infty} \mathbb{E}_{F_N^{Pos}} \left(\mathcal{L}_{\boldsymbol{\gamma}}^{DPDV}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \right) \geq \mathcal{L}_{\boldsymbol{\gamma}}^{DV}(\mathbf{X}_1', \mathbf{X}_2')$, *a.s.,*

ii. $\mathbb{E}_{F_N^{Pos}} \left(\mathcal{L}_{\boldsymbol{\gamma}}^{DPJS}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \right)$ *converges a.s. to* $\mathcal{L}_{\boldsymbol{\gamma}}^{JS}(\mathbf{X}_1', \mathbf{X}_2')$, *as* $n, N \to \infty$,

where "a.s." stands for "almost surely", denoting that the statements hold with probability 1.

The strong consistency of the proposed estimators is also investigated through the next theorem.

Theorem 2 (Consistency)**.** *Considering BNP MINEs given in equation 7. Then, for any label i in* $\{DV, JS\}$, *as* $n, N \to \infty$*:*

i. $MI^{DPi}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \xrightarrow{a.s.} MI^{i}(\mathbf{X}_1', \mathbf{X}_2')$,

ii. There exists a set of neural network $\{T_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}}$ *on some compact domain* $\boldsymbol{\Gamma}$ *such that*

$$MI^{DPi}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \xrightarrow{a.s.} MI(\mathbf{X}_1', \mathbf{X}_2').$$

3.3 Embedding DPMINE in generative models

The BNPWMMD-GAN (Fazeli-Asl & Zhang, 2023) is a VAE-GAN model that places a DP prior on the data distribution. This approach helps prevent overfitting, while also leveraging the advantages of GANs and VAEs in data augmentation. In this study, we aim to refine the BNPWMMD-GAN by using the proposed BNP MINE to further improve its performance. The BNPWMMD-GAN architecture features four networks: The encoder–$\{E_{\boldsymbol{\eta}}\}_{\boldsymbol{\eta} \in \boldsymbol{\mathcal{H}}}$, the generator–$\{G_{\boldsymbol{\omega}}\}_{\boldsymbol{\omega} \in \boldsymbol{\Omega}}$, the code generator–$\{CG_{\boldsymbol{\omega}'}\}_{\boldsymbol{\omega}' \in \boldsymbol{\Omega}'}$, and the discriminator–$\{D_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}}$. The addition of the code generator network sets the BNPWMMD-GAN apart from traditional VAE-GANs. It takes noise from a sub-latent space–the latent space of the code generator–and generates code samples adversarially by using encoder outputs as its training dataset, as described in Eq. equation 8c. The code generator is responsible for exploring uncharted regions of the code space that may have been overlooked by the encoder, thereby enhancing the overall diversity of the generated data.

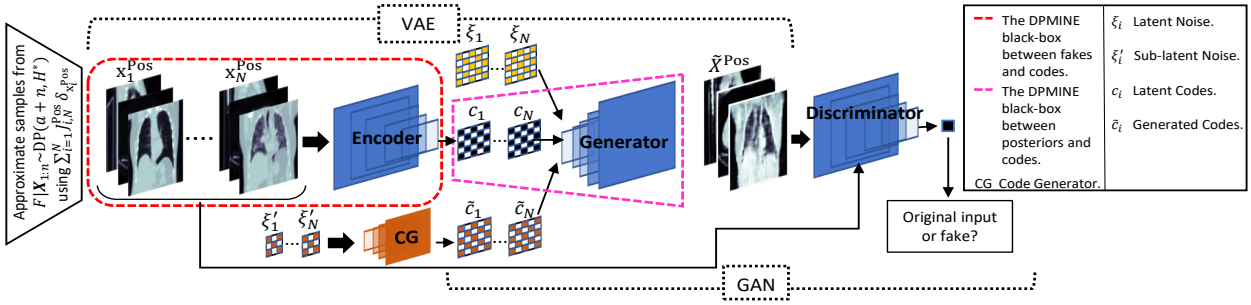


Figure 1: A general diagram of the BNPWMMD model refined by DPMINE in generating 3D images.

Given the codes $\left(\boldsymbol{c}_{1:N} := E\boldsymbol{\eta}\left(\mathbf{X}^{\text{Pos}}_{1:N}\right)\right)$ and generated codes $\left(\widetilde{\boldsymbol{c}}_{1:N} := CG_{\boldsymbol{\omega}'}\left(\boldsymbol{\xi}'_{1:N}\right)\right)$ along with the latent noise $(\boldsymbol{\xi}_{1:N} \in \mathbb{R}^p) \sim F_{\boldsymbol{\xi}}$ and sub-latent noise $(\boldsymbol{\xi}'_{1:N} \in \mathbb{R}^q) \sim F_{\boldsymbol{\xi}'}$, where $q < p$, the BNPWMMD-GAN is trained by updating the networks' parameters according to the following hybrid objective function[1]:

$$(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\eta}})^2 = \underset{\boldsymbol{\Omega},\, \mathcal{H}}{\arg\min} \Bigg\{ \underbrace{-\frac{1}{N}\sum_{i=1}^{N}\Big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}_i)) + D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}_i)) + D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\tilde{\boldsymbol{c}}_i))\Big] + \text{MMD}\big(F^{\text{Pos}}_N, F_{G_{\boldsymbol{\omega}}(\boldsymbol{\xi}_{1:N})}\big)}_{\mathcal{I}_1(\boldsymbol{\omega},\boldsymbol{\eta})}$$

$$\underbrace{\text{-MMD}\big(F^{\text{Pos}}_N, F_{G_{\boldsymbol{\omega}}(\tilde{\boldsymbol{c}}_{1:N})}\big) + \text{MMD}\big(F^{\text{Pos}}_N, F_{G_{\boldsymbol{\omega}}(\boldsymbol{c}_{1:N})}\big)} + \underbrace{\text{MMD}\big(F_{\boldsymbol{\xi}_{1:N}}, F_{\boldsymbol{c}_{1:N}}\big)}_{\mathcal{I}_2(\boldsymbol{\omega},\boldsymbol{\eta})} \Bigg\}, \tag{8a}$$

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\Theta}}{\arg\min} \Bigg\{ \underbrace{\frac{1}{N}\sum_{i=1}^{N}\Big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}_i)) + D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}_i)) + D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\tilde{\boldsymbol{c}}_i)) - 3J^{\text{Pos}}_{i,N} D_{\boldsymbol{\theta}}(\mathbf{X}^{\text{Pos}}_i)\Big]}_{\mathcal{J}_1(\boldsymbol{\theta})} + \lambda \underbrace{L_{\text{GP-D}}}_{\mathcal{J}_2(\boldsymbol{\theta})} \Bigg\}, \tag{8b}$$

$$\hat{\boldsymbol{\omega}}' = \underset{\boldsymbol{\Omega}'}{\arg\min}\, \text{MMD}\big(F_{\boldsymbol{c}_{1:N}}, F_{\tilde{\boldsymbol{c}}_{1:N}}\big). \tag{8c}$$

Here, the generator is fed with $(\boldsymbol{\xi}_{1:N}, \boldsymbol{c}_{1:N}, \widetilde{\boldsymbol{c}}_{1:N})$, and terms $\mathcal{I}_1(\boldsymbol{\omega}, \boldsymbol{\eta})$ and $\mathcal{J}_1(\boldsymbol{\theta})$ refer to minimizing the combined distance given by[3]:

$$d_{\text{WMMD}}(F^{\text{Pos}}, F_{G_{\boldsymbol{\omega}}}) = \text{WS}(F^{\text{Pos}}, F_{G_{\boldsymbol{\omega}}}) + \text{MMD}(F^{\text{Pos}}, F_{G_{\boldsymbol{\omega}}}). \tag{9}$$

The inclusion of distance measurement equation 9 will improve the model training outcomes as it incorporates both the overall distribution comparison and the feature-matching techniques. The idea of minimizing the distance between F^{Pos} and the generator distribution $F_{G_{\boldsymbol{\omega}}}$ was initially introduced independently in the BNP learning approach outlined in Dellaporta et al. (2022); Fazeli-Asl et al. (2024). This approach suggests that such minimization induces a posterior distribution on the generator's parameter space, which, in turn, leads to a posterior approximation for the network parameters. This BNPL strategy effectively handles model misspecification by using a non-parametric prior (Fong et al., 2019; Lyddon et al., 2018; 2019; Lee et al., 2024), which promotes stable learning and improves generalization in generative tasks. Additionally, it mitigates sensitivity to outliers and sample variability through prior regularization, leading to more stable out-of-sample performance compared to conventional learning approaches that depend solely on the empirical data distribution for loss computation (Bariletto & Ho, 2024). A depiction of this strategy is provided by Dellaporta et al. (2022, Figure 1).

[1]Each F indexed by a sample vector in equation 8 indicates the corresponding empirical distribution.

[2]Simultaneous updates of parameters $\boldsymbol{\omega}$ and $\boldsymbol{\eta}$ are facilitated by the essential role played by the generator function, which serves as a decoder during VAE training.

[3]The details for computing the Wasserstein distance (WS) and the maximum mean discrepancy (MMD) are provided in Appendix B.

Since the MMD measure is an L^2-norm distance, the MMD-based terms in $\mathcal{I}_1(\boldsymbol{\omega}, \boldsymbol{\eta})$ serves as the posterior reconstruction errors, while $\mathcal{I}_2(\boldsymbol{\omega}, \boldsymbol{\eta})$ serves as the regularization error for approximating the variational distribution $F_{E_{\boldsymbol{\eta}}}(\boldsymbol{c}_{1:N}|\mathbf{X}_{1:N}^{\text{Pos}})$. Furthermore, the gradient penalty $\mathcal{J}_2(\boldsymbol{\theta})$ with a positive coefficient λ was used to ensure training stability by forcing the 1-Lipschitz constraint on the discriminator (Gulrajani et al., 2017). Specifically, this penalty is given by

$$L_{\text{GP}-\text{D}} = \frac{1}{N}\sum_{i=1}^{N}(\|\nabla_{\widehat{\mathbf{X}}_i^{\text{Pos}}} D_{\boldsymbol{\theta}}(\widehat{\mathbf{X}}_i^{\text{Pos}})\|_2 - 1)^2,$$

where $\widehat{\mathbf{X}}_i^{\text{Pos}} = u\widetilde{\mathbf{X}}_i^{\text{Pos}} + (1-u)\mathbf{X}_i^{\text{Pos}}$ and $\widetilde{\mathbf{X}}_i^{\text{Pos}}$ represents any posterior fake sample generated by the generator.

Now, to maximize the extraction of information during the encoding process and ensure the preservation of meaningful information in the decoding process, we propose refining Eq. equation 8a using the optimization problem in Eq. equation 10a along with incorporating objective functions in Eqs. equation 10b and equation 10c.

$$\begin{aligned}(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\eta}}) = \underset{\boldsymbol{\Omega},\, \mathcal{H}}{\arg\min} \Big\{ & \mathcal{I}_1(\boldsymbol{\omega}, \boldsymbol{\eta}) + \mathcal{I}_2(\boldsymbol{\omega}, \boldsymbol{\eta}) - \mathcal{L}_{\gamma_1}^{\text{DPi}}(\mathbf{X}_{1:N}^{\text{Pos}}, \boldsymbol{c}_{1:N}) - \mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}_{1:N}), \boldsymbol{c}_{1:N}) \\ & - \mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\tilde{\boldsymbol{c}}_{1:N})^4, \boldsymbol{c}_{1:N}) - \mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}_{1:N}), \boldsymbol{c}_{1:N}) \Big\}, \end{aligned} \tag{10a}$$

$$\hat{\gamma}_1 = \underset{\boldsymbol{\Gamma}_1}{\arg\min} \big\{ -\mathcal{L}_{\gamma_1}^{\text{DPi}}(\mathbf{X}_{1:N}^{\text{Pos}}, \boldsymbol{c}_{1:N}) \big\}, \tag{10b}$$

$$\hat{\gamma}_2 = \underset{\boldsymbol{\Gamma}_2}{\arg\min} \big\{ -\mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}_{1:N}), \boldsymbol{c}_{1:N}) - \mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\tilde{\boldsymbol{c}}_{1:N}), \boldsymbol{c}_{1:N}) - \mathcal{L}_{\gamma_2}^{\text{DPi}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}_{1:N}), \boldsymbol{c}_{1:N}) \big\}. \tag{10c}$$

Note that we use the neural network $\{T_{\gamma_1}\}_{\gamma_1 \in \boldsymbol{\Gamma}_1}$ to estimate $\text{MI}(\mathbf{X}, E_{\boldsymbol{\eta}}(\mathbf{X}))$, while we use $\{T_{\gamma_2}\}_{\gamma_2 \in \boldsymbol{\Gamma}_2}$ to estimate $\text{MI}(\widetilde{\mathbf{X}}, E_{\boldsymbol{\eta}}(\mathbf{X}))$. Here, $\widetilde{\mathbf{X}}$ denotes any fake data generated by $G_{\boldsymbol{\omega}}(\cdot)$. The general diagram of the model is visually shown in Figure 1. A detailed description of the model's architecture, accompanied by a flowchart illustrating its operational process, is provided in Appendix E.3.

4 Experimental results

In this section, we evaluate the performance of the procedure using simulation and real datasets. Initially, we need to set the hyperparameters of the DP before placing it on the data distribution F. Following Fazeli-Asl et al. (2024), we consider the base measure to be a multivariate normal distribution, with the concentration parameter estimated using the MAP procedure described in Remark 1 and 2 of Section 3.1. Additionally, the value of N in the DP approximation of Eq. equation 4 is chosen according to the guidance provided in Remark 3 of Section 3.1

We initiate this section by presenting several simulation examples to assess the performance of both KL-based and JS-based DPMINEs in estimating the MI. Subsequently, we will explore the application of the better-performing estimator in refining the training procedure of VAE-GANs.

4.1 MINE: BNP versus FNP

4.1.1 Estimation accuracy

In our experimental evaluation, we must evaluate the accuracy of both the KL-based and JS-based DPMINEs in estimating the MI as it directly impacts the accuracy of refining the desired BNP VAE-GAN model. Additionally, we need to compare the BNP estimators with their frequentist counterparts. To achieve this, we present estimation values of both BNP and FNP procedures through Figure 2 for two cases–when the two random variables X and Y are independent, and when they are dependent. The figure shows that the BNP estimators exhibit better convergence than the FNP estimators as well as lower variance, which stems

[4] $G_{\boldsymbol{\omega}}(\widetilde{\boldsymbol{c}}_{1:N})$ can be expressed as $f_1(\cdot) = G_{\boldsymbol{\omega}}(\widetilde{\boldsymbol{c}}_{1:N}) - \mathbf{X}_{1:N} + \cdot$ in definitions equation 15 and equation 6. Therefore, it is well-defined.

Estimator	Epochs	Dimensions			
		2	**10**	**100**	**1000**
MINE	500	5 sec	5 sec	5 sec	5 sec
	1500	12 sec	12 sec	12 sec	12 sec
DPMINE	500	42 sec	42 sec	42 sec	42 sec
	1500	1 min 25 sec	1 min 25 sec	1 min 25 sec	1 min 25 sec

Table 1: Run time of the MINE and DPMINE on different dimensions of the data and epochs.

from the improved out-of-sample performance of the BNP procedure that reduces sensitivity to sample size and outliers. However, in non-independent cases, the JS-based estimator shows a gap between its estimated value and the true value, unlike the KL-based estimator. Therefore, we decided to continue our investigation in the next experimental part using KL-based estimators.

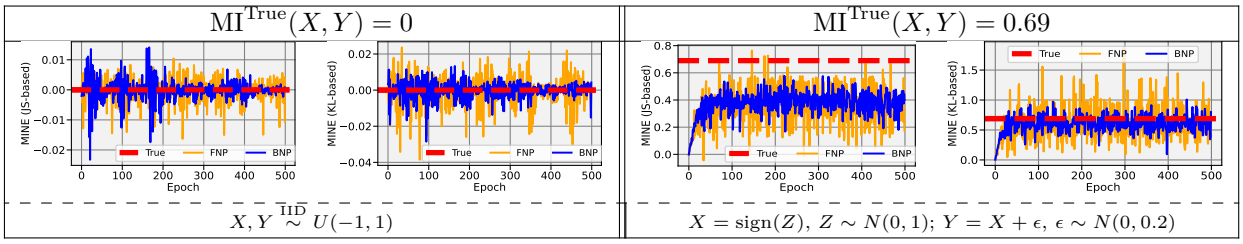


Figure 2: MINE estimation of the MI between two random variables X and Y using both BNP and FNP frameworks, given a sample size of 16 over 500 epochs. The red dashed line represents the true value of MI. The blue line represents the BNP estimation of MI (our method), while the yellow line represents the FNP estimation. The left-hand figure in each experiment represents the JS-based estimator, and the right-hand figure represents the KL-based estimator.

4.1.2 Robustness of the DPMINE to high dimensionality

To assess the robustness of our estimator to high dimensionality, we report the results of two previous examples with varying dimensions of $d = 2, 10, 100, 1000$ (see Figures 3 and 4). The blue color represents the BNP estimation of MI, while the yellow one represents the FNP estimation. These figures demonstrate the robustness of the DPMINE estimator to high dimensionality, indicating better convergence and reduced fluctuations in MI estimation, particularly in cases where the two variables are not independent. This highlights the estimator's ability to control variance, which is a key objective of this paper. In Figure 4(e), we provide results for 1500 epochs, as higher dimensions require more epochs for convergence based on the chosen learning rate, in addition to the results already shown for 500 epochs in Figure 4(d).

4.2 Refining VAE-GAN training via MINE

To implement the BNPWMMD+DPMINE, we consider the Gaussian kernel function, defined as $k_{G_\sigma}(\mathbf{X}, \mathbf{Y}) = \exp(\frac{-||\mathbf{X}-\mathbf{Y}||^2}{2\sigma^2})$ with bandwidth parameter σ, in the MMD distance given in Appendix B. We search for the appropriate bandwidth parameter σ over a fixed grid of values, $\sigma \in \{2, 5, 10, 20, 40, 80\}$. We then compute the mixture of Gaussian kernels, denoted as $k(\cdot,\cdot) = \sum_\sigma k_{G_\sigma}(\cdot,\cdot)$. This selection of kernel function and bandwidth has been shown to yield satisfactory performance in training MMD-based GANs, as mentioned in Li et al. (2015); Fazeli-Asl et al. (2024); Fazeli-Asl & Zhang (2023). Then, we consider next examples to investigate the proposed approach.

4.2.1 Synthetic Example

Coil Dataset: We first look at a synthetic example that showcases the effectiveness of the BNPWMMD+DPMINE in mitigating mode collapse. In this example, we simulate 5000 samples in 3D space,

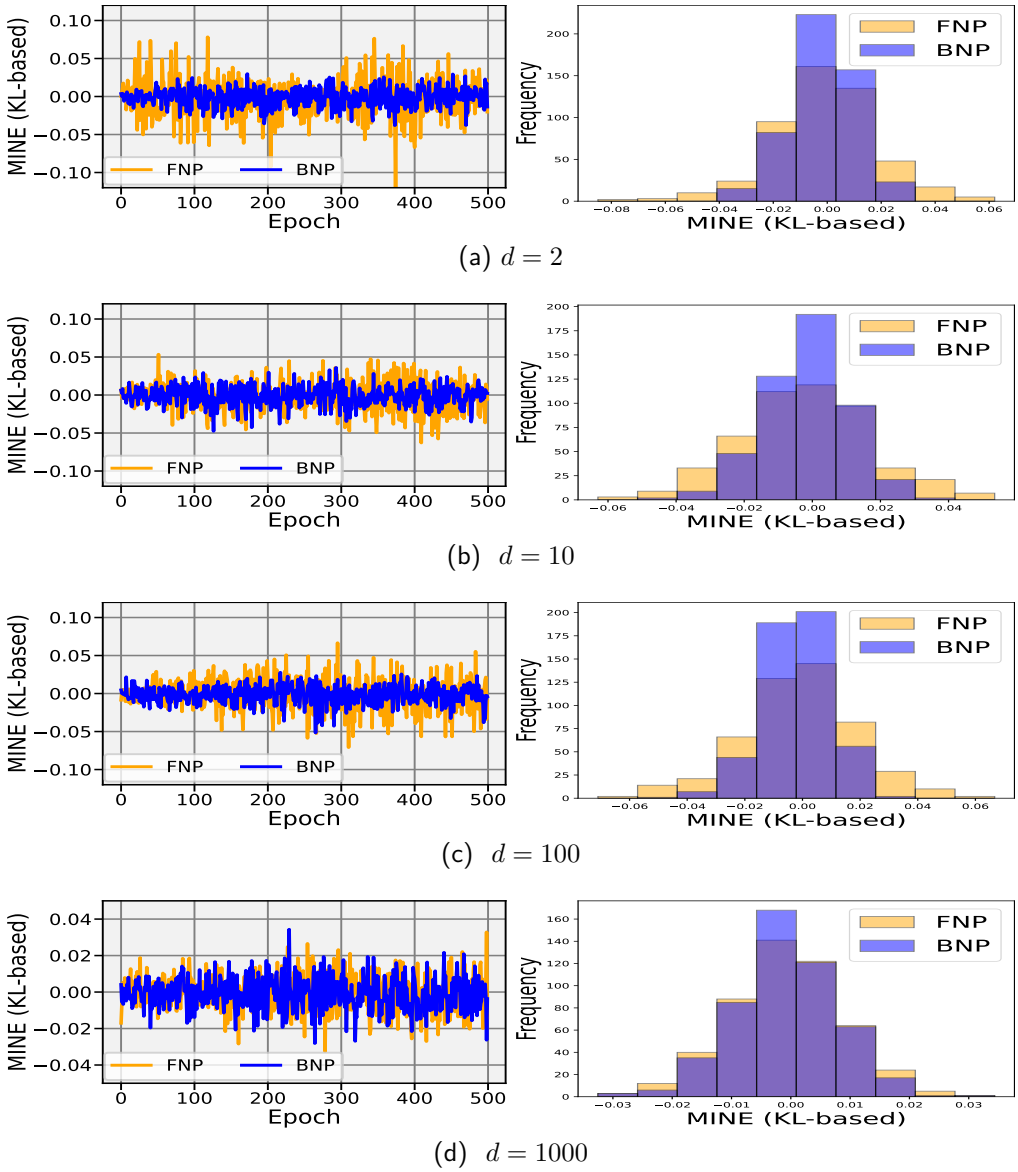
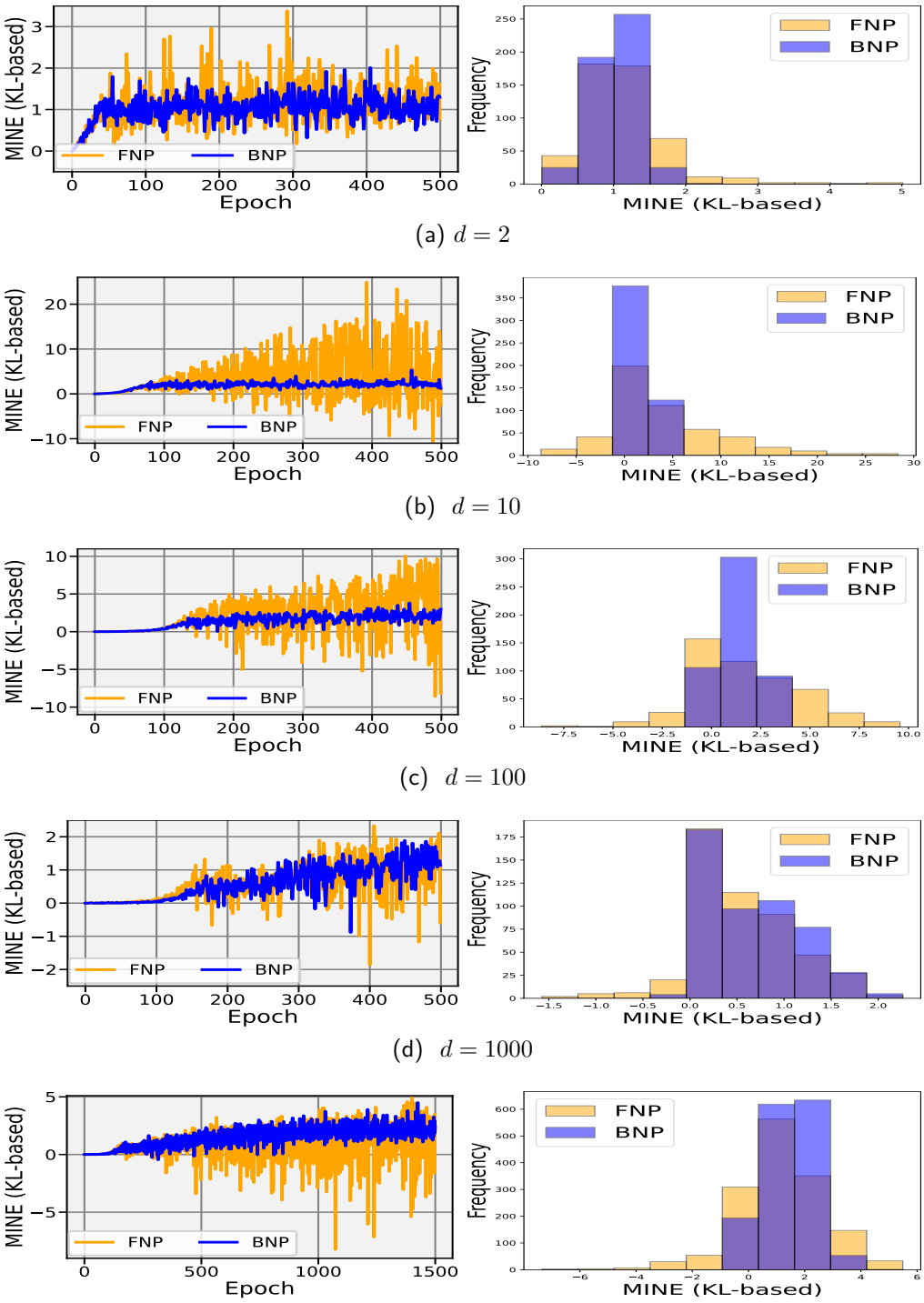


Figure 3: MI estimations between two random variables $\mathbf{X}, \mathbf{Y} \overset{\text{IID}}{\sim} U(-\mathbf{1}, \mathbf{1})$, $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^d$ for various dimension d.

denoted as $(X(t), Y(t), Z(t))$, where $X(t) = 6\cos t$, $Y(t) = 6\sin t$, and $Z(t) = t$, with t ranging from -2π to 4π. We then normalize all datasets to a range between -1 and 1. This normalization ensures compatibility with the hyperbolic tangent activation function, which is used in the generator's last layer in all compared models. We also used a latent dimension of 100 with a sub-latent dimension of 10 in this example. Additionally, we provide the results of the BNPWMMD to display the basic model's performance in covering data space in the absence of DPMINE.

For a comprehensive comparison, we include results from a set of experiments involving three well-known VAE-GAN models commonly used in the literature: BiGAN+MINE, α-WGAN+MINE, and α-WGAN.[5] As demonstrated by Belghazi et al. (2018), adding MINE improves the performance of BiGAN, as evidenced by

[5]In this section, the GP notation is omitted for brevity, as GP is incorporated into the loss functions of both the BNPWMMD and α-WGAN models.



(a) $d = 2$

(b) $d = 10$

(c) $d = 100$

(d) $d = 1000$

(e) $d = 1000$ (epoch=1500)

Figure 4: MI estimations between two random variables $\mathbf{X} = \text{sign}(\mathbf{Z})$, $(\mathbf{Z} \in \mathbb{R}^d) \sim N(\mathbf{0}, I_d)$ and $\mathbf{Y} = \mathbf{X} + \boldsymbol{\epsilon}$, $(\boldsymbol{\epsilon} \in \mathbb{R}^d) \sim N(\mathbf{0}, 0.2I_d)$ for various dimension d.

their comparison between BiGAN and BiGAN+MINE. Thus, we exclude the standalone BiGAN from our experiments.

Notably, BNPWMMD leverages posterior samples generated from F_N^{Pos}, as defined in Eq. equation 4, to optimize its objective functions, rather than relying on direct samples from the empirical distribution used by its FNP counterpart. Consequently, it is essential to employ the BNP version of MINE to regularize BNPWMMD's cost function during network parameter optimization. In contrast, FNP models such as α-WGAN and BiGAN use the empirical distribution to compute sample means in their cost functions, which are compatible with the calculation of MINE in Eq. equation 3 (see Appendix C for further details on the baselines' cost functions). Given these structural differences, applying the MINE designed for FNP models to BNPWMMD (i.e., BNPWMMD+MINE) or using DPMINE with FNP models like α-WGAN and BiGAN (i.e., α-WGAN+DPMINE or BiGAN+DPMINE) is inappropriate. Such combinations are incompatible with the input structures and cost function formulations of these models.

We show 1,000 samples in Figure 5, generated from noise input (random samples) and reconstructed from encoded and decoded real input. This figure clearly demonstrates the impact that the DPMINE has in generative modeling, as it allows the BNPWMMD to cover the entire data space. Although applying the MINE to α-WGAN improves the convergence of random samples compared to the absence of these refinements, it still struggles to effectively cover the data space even with these modifications.

Moreover, the BNPWMMD+DPMINE maintains a good balance in coverage between random and reconstructed samples while the α-WGAN model cannot provide a reasonable balance between these two cases. The BiGAN+MINE performs poorly, in comparison, after 5000 epochs and requires a significantly larger number of epochs to converge. However, we found that even with these extended epochs, it cannot achieve better quality than the BNP counterpart provided in this paper. An additional synthetic example in Appendix E.1 further demonstrates the effectiveness of our model.

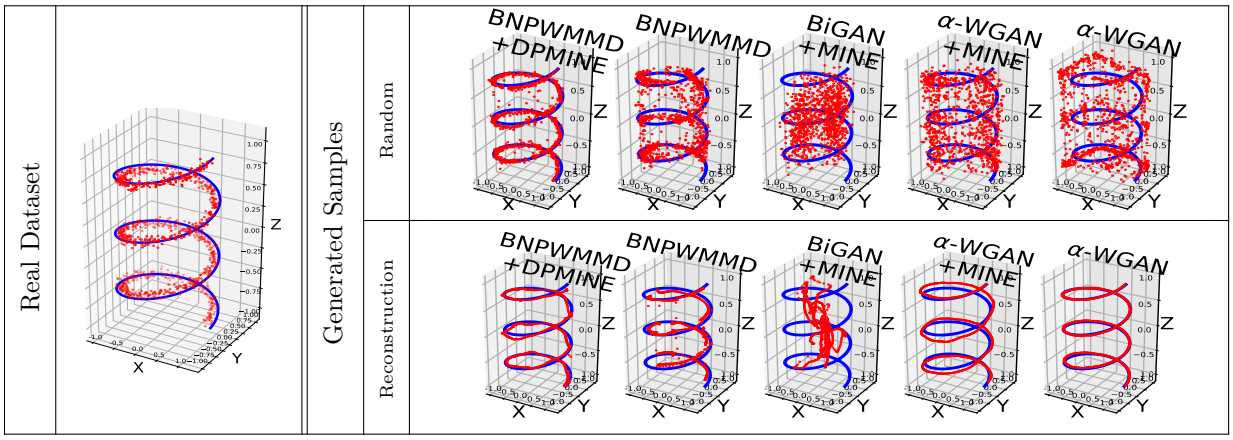


Figure 5: 1000 randomly generated and reconstructed samples for the coil example.

4.2.2 Real Example

COVID-19 Dataset: This dataset comprises of 3D chest CT images of 1000 subjects diagnosed with lung infections after testing positive for COVID-19[6]. All images in this dataset are stored in the DICOM format and have a resolution of 16 bits per pixel, with dimensions of 512×512 pixels in grayscale. We randomly selected 200 patients, resulting in a total of 91,960 images. As part of the preprocessing step, we first stored each patient's data in a Neuroimaging Informatics Technology Initiative (NIFTI) format file. Then, we converted each NIFTI file into a 3D image with a dimension of $64 \times 64 \times 64$. Each dimension represents the axial, sagittal, and coronal views of the lungs. After normalizing the dataset, we used a mini-batch size of 16 and trained all compared models for 7500 epochs. We also used a latent dimension of 1000 as suggested

[6]The dataset is freely available online at `https://doi.org/10.7910/DVN/6ACUZJ` (license: CC0 1.0).

in Kwon et al. (2019) for high-dimensional cases, with a sub-latent dimension of 100 as used in Fazeli-Asl & Zhang (2023).

To showcase the image generation ability of BNPWMMD+DPMINE compared to other models, we provide a 3D visualization of real and random samples of chest CT images in Figure 6. This indicates the exceptional capability of the model to generate sharp images. For a more comprehensive view of the generated samples, we included additional slices from each dimension representation in Appendix E.2 We also compared the reconstruction capability of different models by displaying slices of a reconstructed sample from each model in the Appendix. The BNPWMMD+DPMINE excels at reconstructing the high-dimensional dataset, further validating the effectiveness of our proposed model.

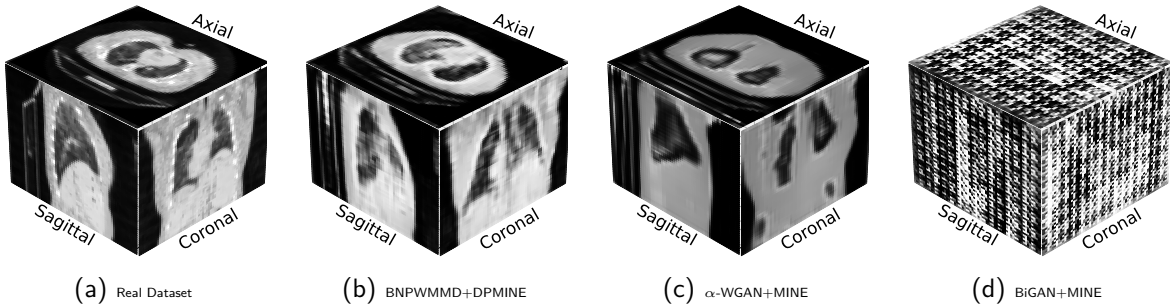
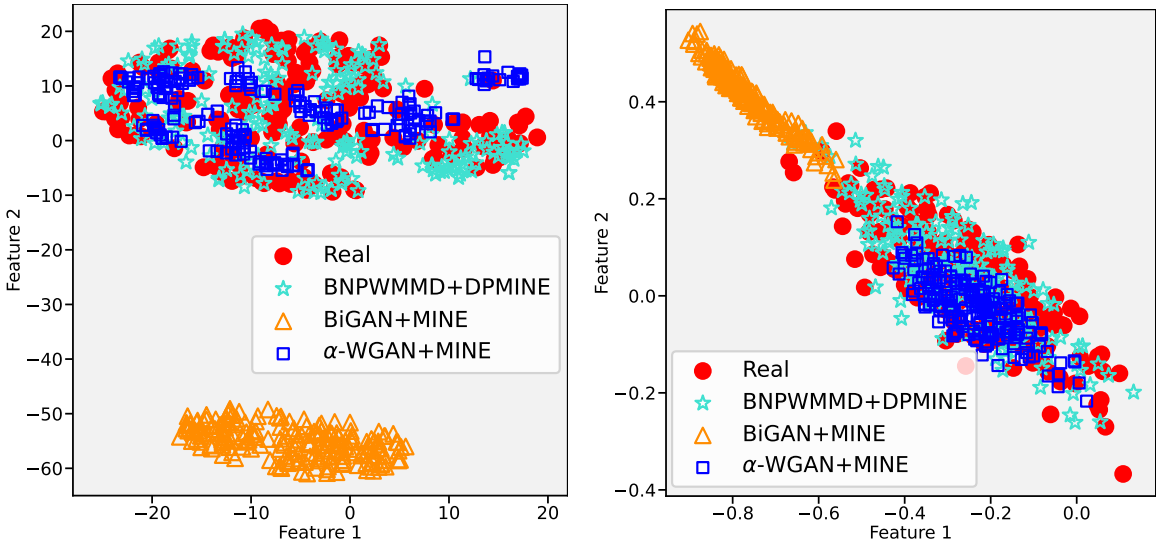


Figure 6: A 3D visualization of a real sample and a randomly generated sample in COVID-19 example.

However, additional quantitative tools are essential to assess the similarity between real and fake samples. Reducing the data dimensionality allows for better visualization to understand the similarities between data points. In Figure 11a(a), we present the t-SNE (Van der Maaten & Hinton, 2008) mapping of the real dataset and 200 random samples to 2D points which clearly shows the extensive coverage of samples generated from the BNPWMMD+DPMINE on the real dataset. Conversely, the α-WGAN+MINE exhibits mode collapse, and BiGAN+MINE demonstrates poor performance. We also use a custom encoder, a 2-layer linear neural network, to map samples to the 2D space displayed by Figure 11a(b), which supports a similar conclusion.

To quantify the dissimilarity in the 2D space between real and generated features, denoted as $\boldsymbol{f}_r := (\text{Feature1}_{1:200,r}, \text{Feature2}_{1:200,r})$ and $\boldsymbol{f}_g := (\text{Feature1}_{1:200,g}, \text{Feature2}_{1:200,g})$, we calculate the average of Fréchet inception distance (FID) and the Kernel inception distance (KID) metrics over 100 replications. Results for the empirical MMD metric between real and generated scans in the 2D feature space, computed with the same kernel and bandwidth parameters used in the BNPWMMD model, are also presented in Table 2. In this table, we deliberately avoid computing MMD in the data space, as our proposed model is already optimized by minimizing MMD between real and fake scans in the data space as a part of training process. As a result, our model is expected to achieve better MMD scores in the data space compared to others, rendering such a comparison unfair. Smaller values in the reported metrics indicate lower dissimilarity and better performance, thereby confirming the significant role of the proposed model in avoiding mode collapse. Additionally, we use the Multi-Scale Structural Similarity Index (MS-SSIM) (Wang et al., 2003) to evaluate the perceived visual quality of images. MS-SSIM captures both structural information and visual quality by performing a multi-scale decomposition and comparing luminance, contrast, and structure across various scales between real and synthetic samples. As an instance-level similarity metric, MS-SSIM assumes a one-to-one mapping between real and generated samples. While the methods being compared are unconditional generators, using MS-SSIM is justified because the axial, sagittal, and coronal views of lung scans are generally well-aligned across patients. This alignment facilitates meaningful comparisons between synthetic and corresponding real views, enabling an evaluation that effectively captures structural fidelity. A higher MS-SSIM score suggests better quality, while a lower score indicates lower quality. For further details on the calculations, refer to Appendix E.3. Additionally, Appendix E.2 presents another real-world example using a brain MRI dataset of patients with tumors, extending the evaluation to a broader range of datasets from

different domains. This inclusion provides a more comprehensive validation of the model's effectiveness and highlights its versatility across various types of data.



(a) t-SNE (b) Custom encoder

Figure 7: Scatter plot of 2D features in COVID-19 example.

Table 2: Comparison of statistical scores for COVID-19 example.

Evaluator		BNPWMMD +DPMINE	α-WGAN +MINE	BiGAN +MINE
FID	Custom Encoder	**0.00063**	0.02177	0.25729
	t-SNE	**9.48447**	13.3682	6298.69
KID	Custom Encoder	**0.00069**	0.02281	0.46684
	t-SNE	**0.99858**	2.31426	3107.53
MMD	Custom Encoder	**0.00902**	0.05131	0.92849
	t-SNE	**0.00074**	0.05742	0.34447
MS-SSIM		**0.43465**	0.39756	0.02251

5 Concluding remarks

We introduced a Bayesian nonparametric (BNP) framework for mutual information estimation using a Dirichlet process-based neural estimator (DPMINE). The core theoretical contribution of this work lies in the construction of a tighter lower bound for mutual information using a finite representation of the Dirichlet process posterior, which was established specifically for KL-based estimators. This tighter bound enhances the informativeness of the MI loss function and introduces a form of regularization that reduces sensitivity to sample variability and outliers, leading to improved convergence during training. The combination of prior regularization and empirical data allows DPMINE to address the trade-off between variance and accuracy, which is a known limitation of existing KL-based estimators. Our experimental results demonstrate that DPMINE outperforms empirical distribution function-based methods (EDF-based) in terms of convergence and variance reduction, particularly in complex high-dimensional generative modeling tasks such as 3D CT image generation.

The proposed estimator effectively enhances the training of deep generative models by improving the reliability and informativeness of the learned representations. Notably, the demonstrated improvements in convergence and variance control make DPMINE highly suitable for handling challenging high-dimensional datasets where accurate mutual information estimation is critical. Although this paper focuses on the application of DPMINE in generative models, the underlying BNP framework extends beyond this setting

and opens new opportunities for optimizing mutual information in other complex learning tasks, including representation learning, reinforcement learning, and Bayesian decision-making.

An important future direction involves adapting the DPMINE framework to address emerging challenges in training large language models (LLMs). Mutual information estimation is increasingly recognized as a key tool for improving representation quality and guiding information flow in transformer-based architectures. The ability of DPMINE to provide stable and accurate estimates of information content could be leveraged to refine LLM pretraining objectives, enhance token embedding strategies, and improve generalization in downstream tasks. Furthermore, by reducing variance and stabilizing training, the BNP framework offers a promising solution for mitigating the sensitivity of LLM training to noise and sample heterogeneity, particularly in few-shot and domain-adaptive settings.

Additionally, the regularization effect introduced by the BNP framework provides a principled approach for reducing overfitting and improving generalization in settings with limited data, such as low-resource languages and domain-specific corpora. Incorporating DPMINE into federated learning contexts could further strengthen privacy-preserving data generation and cross-device model training by enhancing the robustness of information flow across non-IID datasets. However, this will pose significant challenges since the procedure developed in this paper is based on the IID assumption, and extending it to non-IID. settings will require additional methodological adjustments.

References

Luai Al-Labadi, Forough Fazeli Asl, and Zahra Saberi. A test for independence via Bayesian nonparametric estimation of mutual information. *Canadian Journal of Statistics*, 50(3):1047–1070, 2022.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

Nicola Bariletto and Nhat Ho. Bayesian nonparametrics meets data-driven distributionally robust optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=8CguPoe3TP`.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540. PMLR, 2018.

Thomas B Berrett and Richard J Samworth. Nonparametric independence testing via mutual information. *Biometrika*, 106(3):547–566, 2019.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Tian Bu, Zhiyong Yang, Shan Jiang, Guobin Zhang, Hongyun Zhang, and Lin Wei. 3D conditional generative adversarial network-based synthetic medical image augmentation for lung nodule detection. *International Journal of Imaging Systems and Technology*, 31(2):670–681, 2021.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 29, 2016.

Charita Dellaporta, Jeremias Knoblauch, Theodoros Damoulas, and François-Xavier Briol. Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. In *International Conference on Artificial Intelligence and Statistics*, pp. 943–970. PMLR, 2022.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. IV. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Forough Fazeli-Asl and Michael Minyi Zhang. A Bayesian non-parametric approach to generative models: Integrating variational autoencoder and generative adversarial networks using Wasserstein and maximum mean discrepancy. *arXiv preprint arXiv:2308.14048*, 2023.

Forough Fazeli-Asl, Michael Minyi Zhang, and Lizhen Lin. A semi-bayesian nonparametric estimator of the maximum mean discrepancy measure: Applications in goodness-of-fit testing and generative adversarial networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=lUnlHS1FYT`.

Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2): 209–230, 1973.

André Ferreira, Ricardo Magalhães, Sébastien Mériaux, and Victor Alves. Generation of synthetic rat brain MRI scans with a 3D enhanced alpha generative adversarial network. *Applied Sciences*, 12(10):4844, 2022.

Edwin Fong, Simon Lyddon, and Chris Holmes. Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. In *International Conference on Machine Learning*, pp. 1952–1962. PMLR, 2019.

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*, 27:2672–2680, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

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.

Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Jing Huang, Larry Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *Advances in Neural Information Processing Systems*, 35:28319–28334, 2022.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Hemant Ishwaran and Mahmoud Zarepour. Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.

Sanaz Mohammad Jafari, Mucahit Cevik, and Ayse Basar. Improved α-GAN architecture for generating 3D connected volumes with an application to radiosurgery treatment planning. *Applied Intelligence*, pp. 1–27, 2023.

Daniel C Jones, Patrick Danaher, Youngmi Kim, Joseph M Beechem, Raphael Gottardo, and Evan W Newell. An information theoretic approach to detecting spatially varying genes. *Cell Reports Methods*, 3(6), 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Gihyun Kwon, Chihye Han, and Dae-Shik Kim. Generation of 3D brain MRI using auto-encoding generative adversarial networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pp. 118–126. Springer, 2019.

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*, pp. 1558–1566. PMLR, 2016.

Hyungi Lee, Giung Nam, Edwin Fong, and Juho Lee. Enhancing transfer learning with flexible nonparametric posterior sampling. *arXiv preprint arXiv:2403.07282*, 2024.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727. PMLR, 2015.

Chengde Lin, Shengwu Xiong, and Yaxiong Chen. Mutual information maximizing gan inversion for real face with identity preservation. *Journal of Visual Communication and Image Representation*, 87:103566, 2022.

Simon Lyddon, Stephen Walker, and Chris C Holmes. Nonparametric learning from Bayesian models with randomized objective functions. *Advances in Neural Information Processing Systems*, 31, 2018.

Simon P Lyddon, CC Holmes, and SG Walker. General Bayesian updating and the loss-likelihood bootstrap. *Biometrika*, 106(2):465–478, 2019.

José Mendes, Tania Pereira, Francisco Silva, Julieta Frade, Joana Morgado, Cláudia Freitas, Eduardo Negrão, Beatriz Flor de Lima, Miguel Correia da Silva, António J Madureira, et al. Lung CT image synthesis using GANs. *Expert Systems with Applications*, 215:119350, 2023.

Sumeet Menon, Jayalakshmi Mangalagiri, Josh Galita, Michael Morris, Babak Saboury, Yaacov Yesha, Yelena Yesha, Phuong Nguyen, Aryya Gangopadhyay, and David Chapman. CCS-GAN: COVID-19 CT scan generation and classification with very few positive training images. *Journal of Digital Imaging*, pp. 1–14, 2023.

Youssef Mroueh, Igor Melnyk, Pierre Dognin, Jarret Ross, and Tom Sercu. Improved mutual information estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9009–9017, 2021.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11): 5847–5861, 2010.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in Neural Information Processing Systems*, 29, 2016.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

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

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, pp. 639–650, 1994.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations (ICLR)*, 2020.

Masashi Sugiyama. Machine learning with squared-loss mutual information. *Entropy*, 15(1):80–112, 2012.

Ryo Toda, Atsushi Teramoto, Masakazu Tsujimoto, Hiroshi Toyama, Kazuyoshi Imaizumi, Kuniaki Saito, and Hiroshi Fujita. Synthetic CT image generation of shape-controlled lung cancer using semi-conditional InfoGAN and its applicability for type classification. *International Journal of Computer Assisted Radiology and Surgery*, 16:241–251, 2021.

Yao-Hung Hubert Tsai, Martin Q Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. *arXiv preprint arXiv:2103.11275*, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

C Villani. Optimal transport, old and new. *Grundlehren der mathematischen Wissenschaften*, 3, 2008.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. Ieee, 2003.

Liangjian Wen, Yiji Zhou, Lirong He, Mingyuan Zhou, and Zenglin Xu. Mutual information gradient estimation for representation learning. In *International Conference on Learning Representations*, 2020.

Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum. Generative adversarial networks for noise reduction in low-dose CT. *IEEE Transactions on Medical Imaging*, 36(12):2536–2545, 2017.

Mahmoud Zarepour and Luai Al-Labadi. On a rapid simulation of the Dirichlet process. *Statistics & Probability Letters*, 82(5):916–924, 2012.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363. PMLR, 2019.

Appendix

A Notations

- F: Real data distribution.
- $(\mathbf{X}_{1:n})$: A sample of n independent and identically distributed random variables generated from F.
- $\mathrm{DP}(a+n, H^*)$: The Dirichlet process (DP) posterior with concentration parameter $a+n$, $a>0$, and base measure H^*.
- H^*: A mixture probability measure given by $\frac{a}{a+n}H + \frac{n}{a+n}F_{\mathbf{x}_{1:n}}$.
- H: The base measure of the DP prior $\mathrm{DP}(a, H)$.
- $F_{\mathbf{x}_{1:n}}$: The empirical probability measure based on the sample data defined by $\frac{1}{n}\sum_{i=1}^{n}\delta_{\mathbf{X}_i}$.
- F_N^{Pos} : The probability measure of the DP posterior approximation defined by $\sum_{i=1}^{N} J_{i,N}^{\mathrm{Pos}}\delta_{\mathbf{X}_i^{\mathrm{Pos}}}$ with $\left(\mathbf{X}_{1:N}^{\mathrm{Pos}}\right) \sim H^*$ and $\left(J_{1:N,N}^{\mathrm{Pos}}\right) \sim \mathrm{Dirichlet}((a+n)/N, \ldots, (a+n)/N)$ Ishwaran & Zarepour (2002).
- $f_i(\mathbf{X})$: A continuous function of a random variable $\mathbf{X} \sim F$, denoted as $\mathbf{X}_i'$, for $i = 1, 2$.
- $\{T_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}$: A set of continuous functions parameterized by a neural network on a compact domain $\boldsymbol{\Gamma}$.
- I_d: Identity matrix of size $d \times d$.
- "a.s.": Standing for "almost surely" and indicates that the statements hold with probability 1.

B Definition of MMD and Wasserstein distances: DP and empirical representations

Selecting an appropriate statistical distance is crucial for effective generative model training. Here, we focus on the DP representation of two popular distances used in BNP deep learning and we will briefly mention their frequentist counterparts.

B.1 Maximum mean discrepancy distance (feature-matching comparison)

The MMD distance was initially introduced in Gretton et al. (2012) for frequentist two-sample comparisons. Recently, a DP-based version of this distance has been proposed for BNP hypothesis testing Fazeli-Asl et al. (2024). Consider a set of functions $\{G_{\boldsymbol{\omega}}\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ parameterized by a neural network that can generate n IID random variables $(\mathbf{Y}_{1:n})$, where the likelihood function is intractable and not accessible. Let $k(\cdot,\cdot)$ be a continuous kernel function with a feature space corresponding to a universal reproducing kernel Hilbert space Gretton et al. (2012) defined on a compact sample space $\mathfrak{X}$ Gretton et al. (2012). Given a sample $(\mathbf{X}_{1:n}) \overset{\text{IID}}{\sim} F$, the MMD distance between F^{Pos} and $F_{\mathbf{Y}_{1:n}}$ is approximated as:

$$\text{MMD}^2(F_N^{\text{Pos}}, F_{\mathbf{Y}_{1:n}}) := \sum_{\ell,t=1}^{N} J_{\ell,N}^{\text{Pos}} J_{t,N}^{\text{Pos}} k(\mathbf{X}_\ell^{\text{Pos}}, \mathbf{X}_t^{\text{Pos}}) - \frac{2}{n}\sum_{\ell=1}^{N}\sum_{t=1}^{n} J_{\ell,N}^{\text{Pos}} k(\mathbf{X}_\ell^{\text{Pos}}, \mathbf{Y}_t) + \frac{1}{n^2}\sum_{\ell,t=1}^{n} k(\mathbf{Y}_\ell, \mathbf{Y}_t). \quad (11)$$

In the frequentist version of the MMD distance, as defined in Gretton et al. (2012), the empirical distribution $F_{\mathbf{X}_{1:n}}$ is considered. This is denoted as $\text{MMD}^2(F_{\mathbf{X}_{1:n}}, F_{\mathbf{Y}_{1:n}})$, which is obtained by replacing N, $J_{1:N,N}^{\text{Pos}}$, and $\mathbf{X}_{1:N}^{\text{Pos}}$ with n, $1/n$, and $\mathbf{X}_{1:n}$, respectively, in Eq. equation 11.

B.2 Wasserstein distance (overall distribution comparison)

The frequentist version of the Wasserstein distance is completely discussed in (Villani, 2008, Part I6). Fazeli-Asl et al. Fazeli-Asl & Zhang (2023) proposed a BNP version of this distance through its Kantorovich-Rubinstein dual representation. Let $\{D_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ be a parametrized family of continuous functions that all are 1-Lipschitz. Then the Wasserstein distance between F^{Pos} and $F_{\mathbf{Y}_{1:n}}$ is approximated as:

$$\text{WS}(F_N^{\text{Pos}}, F_{\mathbf{Y}_{1:n}}) := \max_{\Theta} \sum_{i=1}^{N} \left(J_{i,N}^{\text{Pos}} D_\theta(\mathbf{X}_i^{\text{Pos}}) - \frac{D_{\boldsymbol{\theta}}(\mathbf{Y}_i)}{n} \right). \quad (12)$$

By utilizing modifications in equation 12 similar to those described for the MMD distance, the empirical representation of the Wasserstein distance can also be obtained. In this section, we propose two novel representations for the MINE using the DP. These representations are based on the KL and JS divergences will be used in our BNP learning framework to maximize information during the training process.

C Baseline Models

In this section, we present details of the two baseline models used in our experiments. Their cost functions are defined in terms of the general expectation of random variables. However, in practice, all expectations appearing in their objective functions are approximated using empirical distributions.

C.1 α-WGAN+Gradient penalty

The α-WGAN+GP, introduced by Kwon et al. (2019), is a VAE-GAN–a hybrid generative model that combines Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs)–specifically designed for data synthesis. It incorporates the encoder–$\{E_{\boldsymbol{\eta}}\}_{\boldsymbol{\eta}\in\boldsymbol{\mathcal{H}}}$, the generator–$\{G_{\boldsymbol{\omega}}\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$, the code discriminator–$\{CD_{\boldsymbol{\theta}'}\}_{\boldsymbol{\theta}'\in\boldsymbol{\Theta}'}$, the discriminator–$\{D_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}$, and the Wasserstein GAN with gradient penalty framework. For training dataset $\mathbf{X} \sim F$, the model is trained by updating the networks' parameters according to the following

hybrid objective function:

$$(\hat{\boldsymbol{\omega}},\, \hat{\boldsymbol{\eta}}) = \arg\min_{\boldsymbol{\Omega},\, \mathcal{H}} \Big\{ -\mathbb{E}_{F_{\boldsymbol{c}}}\big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}))\big] - \mathbb{E}_{F_{\boldsymbol{\xi}}}\big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}))\big] + \lambda_1\, \mathbb{E}_{F_{\boldsymbol{c}}}\big[\|\mathbf{X} - G_{\boldsymbol{\omega}}(\boldsymbol{c})\|_1\big] \Big\}$$

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\Theta}} \Big\{ \mathbb{E}_{F_{\boldsymbol{c}}}\big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{c}))\big] + \mathbb{E}_{F_{\boldsymbol{\xi}}}\big[D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}))\big] - 2\,\mathbb{E}_{F}\big[D_{\boldsymbol{\theta}}(\mathbf{x})\big] + \lambda_2\, L_{\text{GP-D}} \Big\} \tag{13}$$

$$\hat{\boldsymbol{\theta}}' = \arg\min_{\boldsymbol{\Theta}'} \Big\{ \mathbb{E}_{F_{\boldsymbol{c}}}\big[C\, D_{\boldsymbol{\theta}'}(\boldsymbol{c})\big] - \mathbb{E}_{F_{\boldsymbol{\xi}}}\big[C\, D_{\boldsymbol{\theta}'}(\boldsymbol{\xi})\big] + \lambda_2\, L_{\text{GP-CD}} \Big\}. \tag{14}$$

where $\boldsymbol{\xi}$ is a noise vector sampled from the distribution $F_{\boldsymbol{\xi}}$, and $\boldsymbol{c} := E_{\boldsymbol{\eta}}(\mathbf{X})$ represents the latent representation of $\mathbf{X}$. The term $\lambda_1 \mathbb{E}_{F_{\boldsymbol{c}}} \|\mathbf{X} - G_{\boldsymbol{\omega}}(\boldsymbol{c})\|_1$ denotes the reconstruction loss, inferred by modeling the data distribution with a Laplace distribution. Additionally, the gradient penalty terms $L_{\text{GP-D}}$ and $L_{\text{GP-CD}}$, scaled by the coefficient λ_2, are added to Eqs. equation 13 and equation 14, respectively, to enforce the 1-Lipschitz constraint on the discriminators.

C.2 BiGAN

BiGAN, independently introduced by Donahue et al. (2016) and Dumoulin et al. (2016), is a variant of the VAE-GAN framework designed for data generation. In BiGAN, the discriminator distinguishes between pairs of data and latent codes—either real data paired with its encoded representation or generated data paired with its sampled latent vector. This bidirectional structure enables the model to simultaneously learn a mapping from data to the latent space and from the latent space back to the data. The training of BiGAN involves optimizing the following objectives:

$$(\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\eta}}) = \arg\min_{\boldsymbol{\Omega}, \boldsymbol{\mathcal{H}}} \{ -\mathbb{E}_{F_{\boldsymbol{\xi}}}[\ln D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}), \boldsymbol{\xi})] - \mathbb{E}_{F}[\mathbb{E}_{F_{\boldsymbol{c}}}[1 - \ln D_{\boldsymbol{\theta}}(\mathbf{X}, \boldsymbol{c})]] \},$$

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\Theta}} \{ -\mathbb{E}_{F}[\mathbb{E}_{F_{\boldsymbol{c}}}[\ln D_{\boldsymbol{\theta}}(\mathbf{X}, \boldsymbol{c})]] - \mathbb{E}_{F_{\boldsymbol{\xi}}}[\ln (1 - D_{\boldsymbol{\theta}}(G_{\boldsymbol{\omega}}(\boldsymbol{\xi}), \boldsymbol{\xi}))] \}.$$

D Theoretical proofs

D.1 Proof of Theorem 1

Theorem 3 (Limiting expectation)**.** *Considering DP posterior representations of MINE presented in the main paper. Given the DP posterior approximation F_N^{Pos}, we have,*

i. $\lim_{n,N\to\infty} \mathbb{E}_{F_N^{Pos}} \left(\mathcal{L}_{\boldsymbol{\gamma}}^{DPDV}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos}))\right) \geq \mathcal{L}_{\boldsymbol{\gamma}}^{DV}(\mathbf{X}_1', \mathbf{X}_2')$*, a.s.,*

ii. $\mathbb{E}_{F_N^{Pos}} \left(\mathcal{L}_{\boldsymbol{\gamma}}^{DPJS}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos}))\right)$ *converges a.s. to* $\mathcal{L}_{\boldsymbol{\gamma}}^{JS}(\mathbf{X}_1', \mathbf{X}_2')$*, as* $n, N \to \infty$*,*

Proof. Recall that

$$\mathcal{L}_{\boldsymbol{\gamma}}^{\text{DPDV}}(f_1(\mathbf{X}_{1:N}^{\text{Pos}}), f_2(\mathbf{X}_{1:N}^{\text{Pos}})) \quad := \quad \sum_{\ell=1}^{N} J_{\ell,N}^{\text{Pos}} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\text{Pos}}), f_2(\mathbf{X}_{\ell}^{\text{Pos}})) \;-\; \ln \sum_{\ell=1}^{N} J_{\ell,N}^{\text{Pos}} e^{T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\text{Pos}}), f_2(\mathbf{X}_{\pi(\ell)}^{\text{Pos}}))}. \tag{15}$$

Considering the property of Dirichlet distribution, $\mathbb{E}_{F^{\text{Pos}}}(J_{\ell,N}^{\text{Pos}}) = 1/N$, and then applying Jensen's inequality in the above equation implies

$$\mathbb{E}_{F^{\text{Pos}}} \left(\mathcal{L}_{\boldsymbol{\gamma}}^{\text{DPDV}}(f_1(\mathbf{X}_{1:N}^{\text{Pos}}), f_2(\mathbf{X}_{1:N}^{\text{Pos}}))\right) \geq \sum_{\ell=1}^{N} \frac{1}{N} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\text{Pos}}), f_2(\mathbf{X}_{\ell}^{\text{Pos}})) - \ln \sum_{\ell=1}^{N} \frac{1}{N} e^{T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_{\ell}^{\text{Pos}}), f_2(\mathbf{X}_{\pi(\ell)}^{\text{Pos}}))}$$
$$= I. \tag{16}$$

As n approaches infinity, the Glivenko-Cantelli theorem implies that $F_{\mathbf{x}_{1:n}} \xrightarrow{a.s.} F$, and subsequently, $H^* \xrightarrow{a.s.} F$. This implies that $\left(\mathbf{X}_{1:N}^{\text{Pos}}\right)$ converges to $(\mathbf{X}_{1:N})$, a sample of N random variables generated from

F. Therefore, using the continuous mapping theorem as $n \to \infty$, we have the convergence:

$$I \quad \xrightarrow{a.s.} \quad \sum_{\ell=1}^{N} \frac{1}{N} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell), f_2(\mathbf{X}_\ell)) \quad - \quad \ln \sum_{\ell=1}^{N} \frac{1}{N} e^{T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell), f_2(\mathbf{X}_{\pi(\ell)}))} \quad = \quad \mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(f_1(\mathbf{X}_{1:N}), f_2(\mathbf{X}_{1:N})).$$

Here, we use $\mathcal{L}_{\boldsymbol{\gamma}}^{DV}(f_1(\mathbf{X}_{1:N}), f_2(\mathbf{X}_{1:N}))$ to indicate the empirical representation of the DV lower bound based on the N samples $(\mathbf{X}_{1:N})$. On the other hand, the law of the large number implies

$$\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(f_1(\mathbf{X}_{1:N}), f_2(\mathbf{X}_{1:N})) \xrightarrow{a.s.} \mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2'), \tag{17}$$

as $N \to \infty$.

Now, subtract $\mathcal{L}\boldsymbol{\gamma}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2')$ from both sides of equation 16 and take the limit as n and N approach infinity:

$$\lim_{n,N\to\infty} \left(\mathbb{E}_{F^{\mathrm{Pos}}}\left(\mathcal{L}\boldsymbol{\gamma}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}}))\right) - \mathcal{L}\boldsymbol{\gamma}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2')\right) \geq 0,$$

which completes the proof of (i). A similar method is used to prove (ii) and it is then omitted. □

D.2 Proof of Theorem 2

Theorem 4 (Consistency)**.** *Considering BNP MINEs given in the main paper. Then, for any label i in $\{DV, JS\}$, as $n, N \to \infty$:*

i. $MI^{DPi}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \xrightarrow{a.s.} MI^{i}(\mathbf{X}_1', \mathbf{X}_2')$,

ii. There exists a set of neural network $\{T_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}$ on some compact domain $\boldsymbol{\Gamma}$ such that

$$MI^{DPi}(f_1(\mathbf{X}_{1:N}^{Pos}), f_2(\mathbf{X}_{1:N}^{Pos})) \xrightarrow{a.s.} MI(\mathbf{X}_1', \mathbf{X}_2'). \tag{18}$$

Proof. We will only provide the proof for the DPDV estimator of the MI. The proof for the DPJS estimator is similar and therefore omitted.

To prove (i), following the proof of Theorem 1, we have $\left(\mathbf{X}_{1:N}^{\mathrm{Pos}}\right) \xrightarrow{a.s.} (\mathbf{X}_{1:N})$ as $n \to \infty$. Then, for any $\ell \in \{1, \ldots, N\}$, the continuous mapping theorem implies:

$$T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell^{\mathrm{Pos}}), f_2(\mathbf{X}_\ell^{\mathrm{Pos}})) \xrightarrow{a.s.} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell), f_2(\mathbf{X}_\ell)), \tag{19}$$

$$T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell^{\mathrm{Pos}}), f_2(\mathbf{X}_{\pi(\ell)}^{\mathrm{Pos}})) \xrightarrow{a.s.} T_{\boldsymbol{\gamma}}(f_1(\mathbf{X}_\ell), f_2(\mathbf{X}_{\pi(\ell)})), \tag{20}$$

as $n \to \infty$. On the other hand, following Fazeli-Asl et al. (Fazeli-Asl & Zhang, 2023, Theorem 1), as $n \to \infty$, we have:

$$J_{\ell,N}^{\mathrm{Pos}} \xrightarrow{a.s.} \frac{1}{N}. \tag{21}$$

Now, considering results equation 19, equation 20, and equation 21 in the DP representation of the DV lower bound given in equation 15 when $n \to \infty$, we can imply:

$$\mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) \xrightarrow{a.s.} \mathcal{L}_{\boldsymbol{\gamma}}^{\mathrm{DV}}(f_1(\mathbf{X}_{1:N}), f_2(\mathbf{X}_{1:N})). \tag{22}$$

Finally, considering the convergence in equation 17 on the right-hand side of Eq. equation 22 when $N \to \infty$, since $\max(\cdot)$ is a continuous function, the continuous mapping theorem concludes the proof of (i).

To prove (ii), the triangular inequality implies:

$$\begin{aligned}\Big|\mathrm{MI}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) - \mathrm{MI}(\mathbf{X}_1', \mathbf{X}_2')\Big| \\ \leq \Big|\mathrm{MI}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) - \mathrm{MI}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2')\Big| \\ + \Big|\mathrm{MI}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2') - \mathrm{MI}(\mathbf{X}_1', \mathbf{X}_2')\Big|.\end{aligned} \tag{23}$$

Regarding almost surely convergence in part (i), given $\epsilon_1 > 0$, there exists $N_0 \in \mathbb{N}$ such that for all $n, N > N_0$,

$$\left|\mathrm{MI}^{\mathrm{DPDV}}(f_1(\mathbf{X}_{1:N}^{\mathrm{Pos}}), f_2(\mathbf{X}_{1:N}^{\mathrm{Pos}})) - \mathrm{MI}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2')\right| < \epsilon_1, \text{ a.s.} \tag{24}$$

On the other hand, following Belghazi et al. (Belghazi et al., 2018, Lemma 1) for given $\epsilon_2 > 0$, there exists a set of neural network $\{T_\gamma\}_{\gamma \in \mathbf{\Gamma}}$ on some compact domain $\mathbf{\Gamma}$ such that

$$\left|\mathrm{MI}^{\mathrm{DV}}(\mathbf{X}_1', \mathbf{X}_2') - \mathrm{MI}(\mathbf{X}_1', \mathbf{X}_2')\right| < \epsilon_2, \text{ a.s.} \tag{25}$$

Now using equation 24 and equation 25 in equation 23 with choosing $\epsilon_1 = \epsilon_2 = \epsilon/2$ completes the proof. □

E Additional results and implementing details

E.1 synthetic example

Stanford Bunny Dataset: We use the Stanford Bunny dataset, available at `https://graphics.stanford.edu/data/3Dscanrep/`, to investigate the impact of DPMINE on an additional synthetic example. The dataset consists of point clouds representing the Stanford Bunny, a renowned 3D model provided by the Stanford University Computer Graphics Laboratory. These point clouds, captured with the Cyberware 3030 MS scanner and stored in PLY files (Polygon File Format) developed at Stanford, represent spatial locations on the object's surface. Point clouds provide a 3D spatial representation of the object, enabling detailed visualization and analysis. For our study, we use three point clouds from this dataset and apply filtering with the *pyoints* Python library to extract points in 3D space.

We randomly sampled 5000 points from the available 43,188 points as the training dataset and implemented our model on them by feeding the model with noise inputs to generate 2500 random samples and with encoded real inputs to obtain reconstruction samples. Figure 8 illustrates the significant impact of DPMINE on the performance of the BNP VAE-GAN in data generation.

E.2 Real examples

Covid-19 dataset: The red border in Figure 9 indicates the corresponding slices depicted in Figure 6 of the experimental findings discussed in the main paper. This clearly demonstrates the excellent performance of BNPWMMD+DPMINE in displaying sharp and diverse slices in a 3D randomly generated sample, surpassing the performance of other methods. Additionally, Figure 10 includes reconstructed samples that exhibit the highest similarity to the training dataset for the BNPWMMD+DPMINE method.

BraTS 2018 Dataset: The BRATS 2018 dataset, a benchmark resource in medical imaging available at `https://www.med.upenn.edu/sbia/brats2018/data.html`, is employed for training models in the generation of brain tumor MRI scans. For the experiments, data from 210 subjects labeled as "HGG" (High-Grade Glioma) are used, focusing on patients with aggressive brain tumors. Each subject's MRI data includes four distinct imaging modalities: T1-weighted (T1), T1-weighted with contrast enhancement (T1ce), T2-weighted (T2), and Fluid Attenuated Inversion Recovery (FLAIR). Notably, the FLAIR modality, which is particularly effective for highlighting edema and tumor boundaries, is used for the experiments, providing essential insights for synthetic MRI generation.

The results presented in Figures 11-14 and Table 3 reinforce our previous findings on the lung dataset, demonstrating similarly strong performance when applying the methodology to the BraTS dataset. This cross-domain evaluation further validates the model's robustness and adaptability to diverse data types.

E.3 Implementing details

E.3.1 Calculation of evaluation scores

Considering the real and generated features, $\boldsymbol{f}_r := (\mathrm{Feature1}_{1:200,r}, \mathrm{Feature2}_{1:200,r})$ and $\boldsymbol{f}_g := (\mathrm{Feature1}_{1:200,g}, \mathrm{Feature2}_{1:200,g})$, the FID and KID metrics are calculated using the following Eqs., respec-

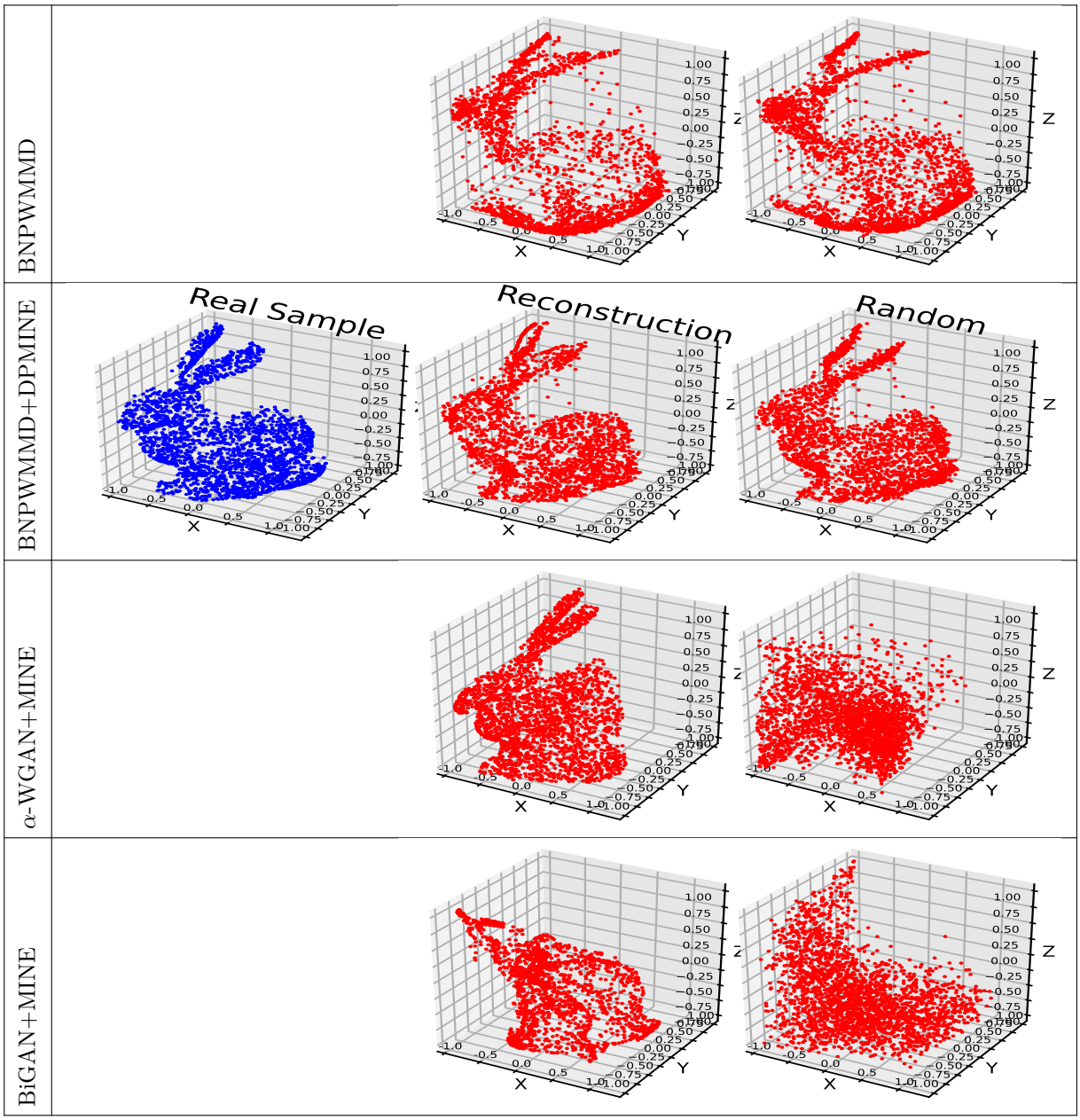


Figure 8: 2500 samples for the Stanford Bunny using BNPWMMD+DPMINE, α-WGAN+MINE, and BiGAN+MINE after 5000 epochs.

tively Bińkowski et al. (2018):

$$\mathrm{FID}(\boldsymbol{f}_r, \boldsymbol{f}_g) = ||\boldsymbol{\mu}_{\boldsymbol{f}_r} - \boldsymbol{\mu}_{\boldsymbol{f}_g}||^2 + \mathrm{Tr}\Big(\Sigma_{\boldsymbol{f}_r} + \Sigma_{\boldsymbol{f}_g} - 2\sqrt{\Sigma_{\boldsymbol{f}_r}\Sigma_{\boldsymbol{f}_g}}\Big),$$
$$\mathrm{KID}(\boldsymbol{f}_r, \boldsymbol{f}_g) = \mathrm{MMD}^2(\boldsymbol{f}_r, \boldsymbol{f}_g),$$

where $\boldsymbol{\mu}_{\boldsymbol{f}_r}$ and $\boldsymbol{\mu}_{\boldsymbol{f}_g}$, and $\Sigma_{\boldsymbol{f}_r}$ and $\Sigma_{\boldsymbol{g}_r}$ are the mean vector and covariance matrix of $\boldsymbol{f}_r$ and $\boldsymbol{f}_g$, respectively. The empirical MMD metric in Gretton et al. (2012) is used to compute $\mathrm{MMD}^2(\cdot,\cdot)$, with a polynomial kernel $k(\boldsymbol{f}_r, \boldsymbol{f}_g) = (0.5\boldsymbol{f}_r^{\mathrm{T}}\boldsymbol{f}_g + 1)^{\nu}$ and degree ν. The "Tr" denotes matrix trace, "T" denotes matrix transpose, and "$||\cdot||$" denotes Euclidean norm. We use the provided code at `https://torchmetrics.readthedocs.io/en/v0.8.2/image/kernel_inception_distance.html` to compute KID with $\nu = 3$, and also use the provided code at `https://pytorch.org/ignite/generated/ignite.metrics.FID.html` to calculate FID.

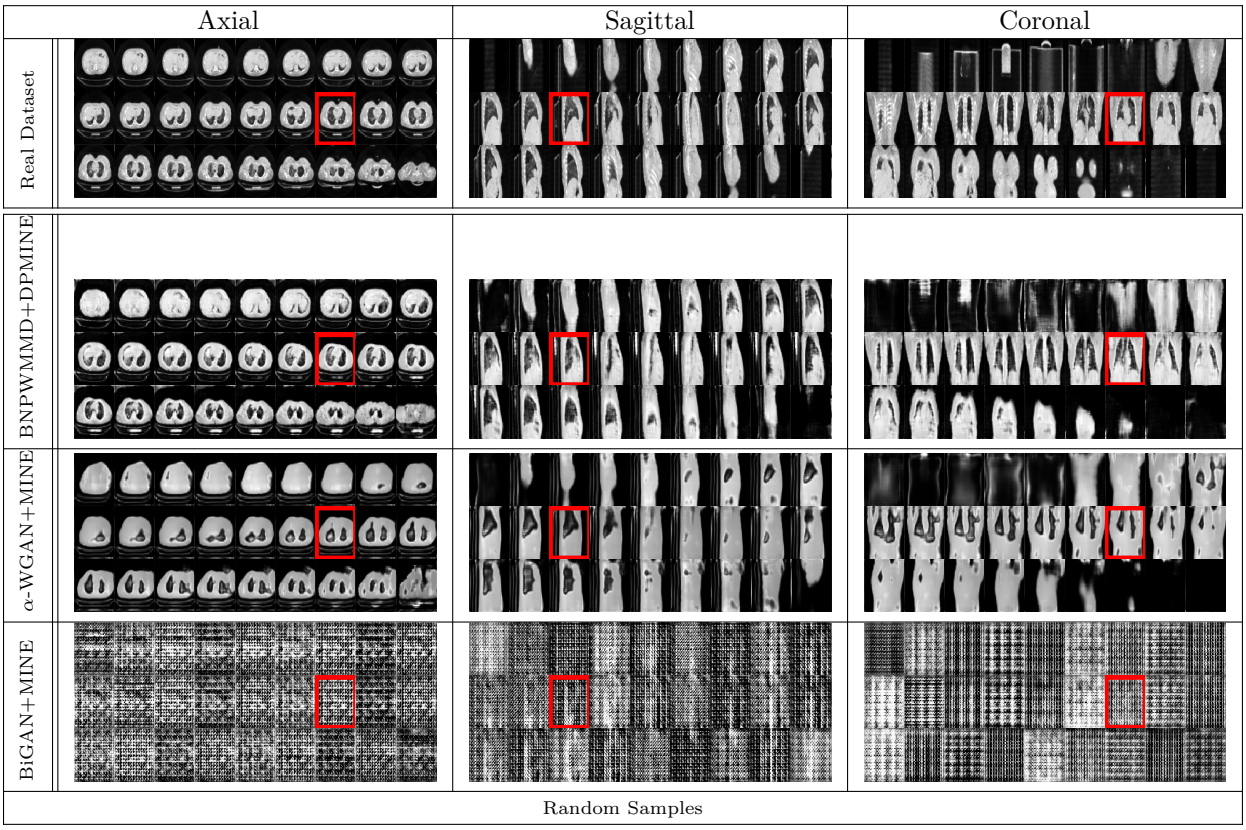


Figure 9: 27 slices of each side of a 3D sample randomly generated from BNPWMMD+DPMINE, BiGAN+MINE, and α-WGAN+MINE after 7500 epochs for COVID-19 example.

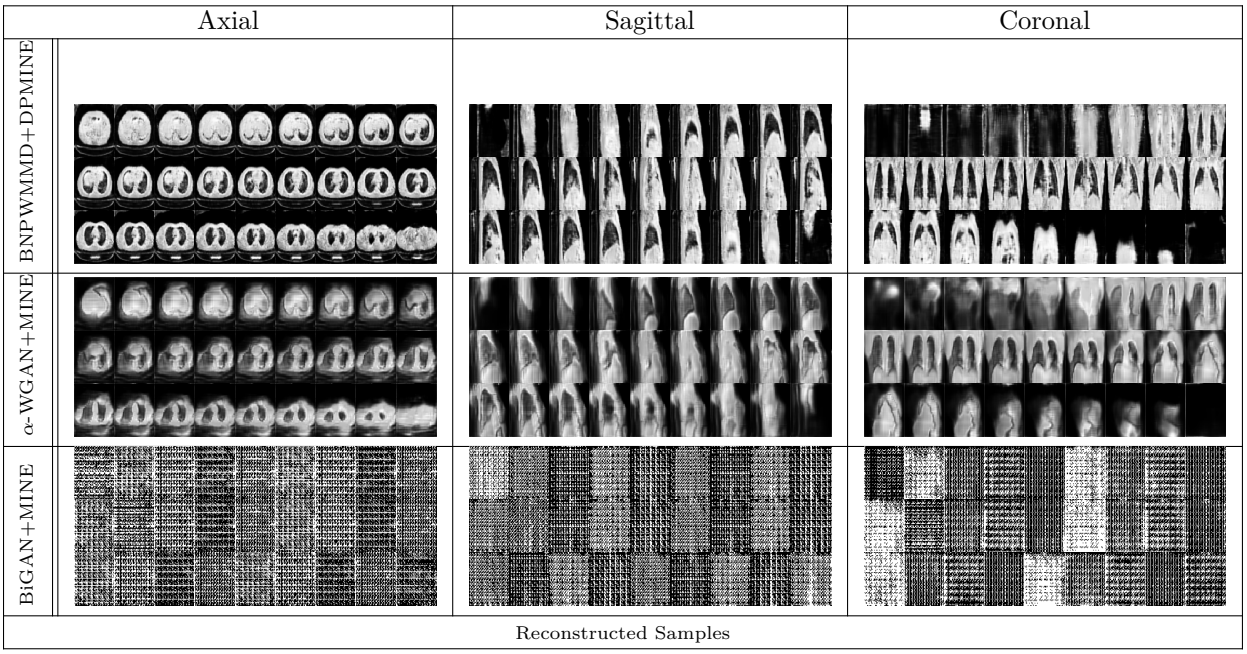


Figure 10: 27 slices of each side of a 3D sample reconstructed by BNPWMMD+DPMINE, BiGAN+MINE, and α-WGAN+MINE after 7500 epochs for COVID-19 example.

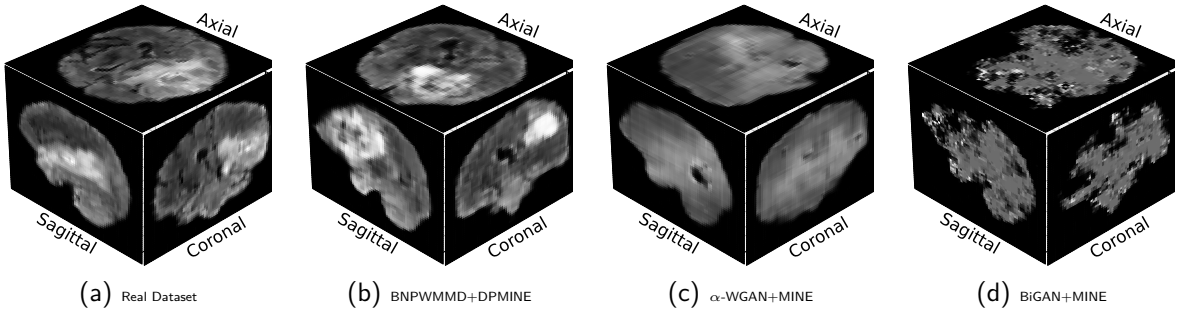


Figure 11: A 3D visualization of a real sample and a randomly generated sample in BraTS18 example.

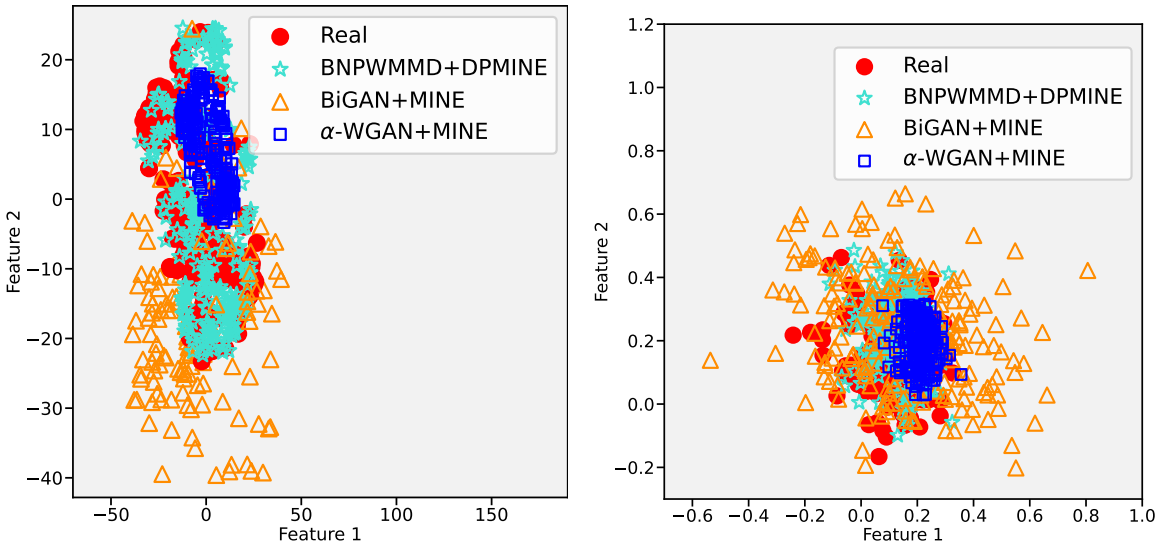
Additionally, we calculate MS-SSIM using the available codes at `https://torchmetrics.readthedocs.io/en/v0.8.2/image/multi_scale_structural_similarity.html`.

E.3.2 Network setting

In this section, we provide comprehensive information regarding the architectures of all networks used in the experimental results. The encoder network, generator network, and discriminator network architectures, as presented in Tables 4, 6, and 7 respectively, were implemented based on Kwon et al. (2019) for the α-WGAN model. The relevant codes[7] are available at `https://github.com/cyclomon/3dbraingen.git`. Moreover, the BNPWMMD model proposed in Fazeli-Asl & Zhang (2023) incorporates these architectures, alongside an additional code generator outlined in Table 5. Figure 15 provides a flowchart illustrating the operation process of BNPWMMD+DPMINE.

On the other hand, we have attempted to adapt the generator and discriminator architectures given in Belghazi et al. (2018) for generating 3D chest CT images using BiGAN. The generator architecture can be found in Table 8, and the discriminator architecture can be found in Table 9. We have also considered the

[7]It is licensed under the MIT License.



(a) t-SNE (b) Custom encoder

Figure 12: Scatter plot of 2D features in BraTS18 example.

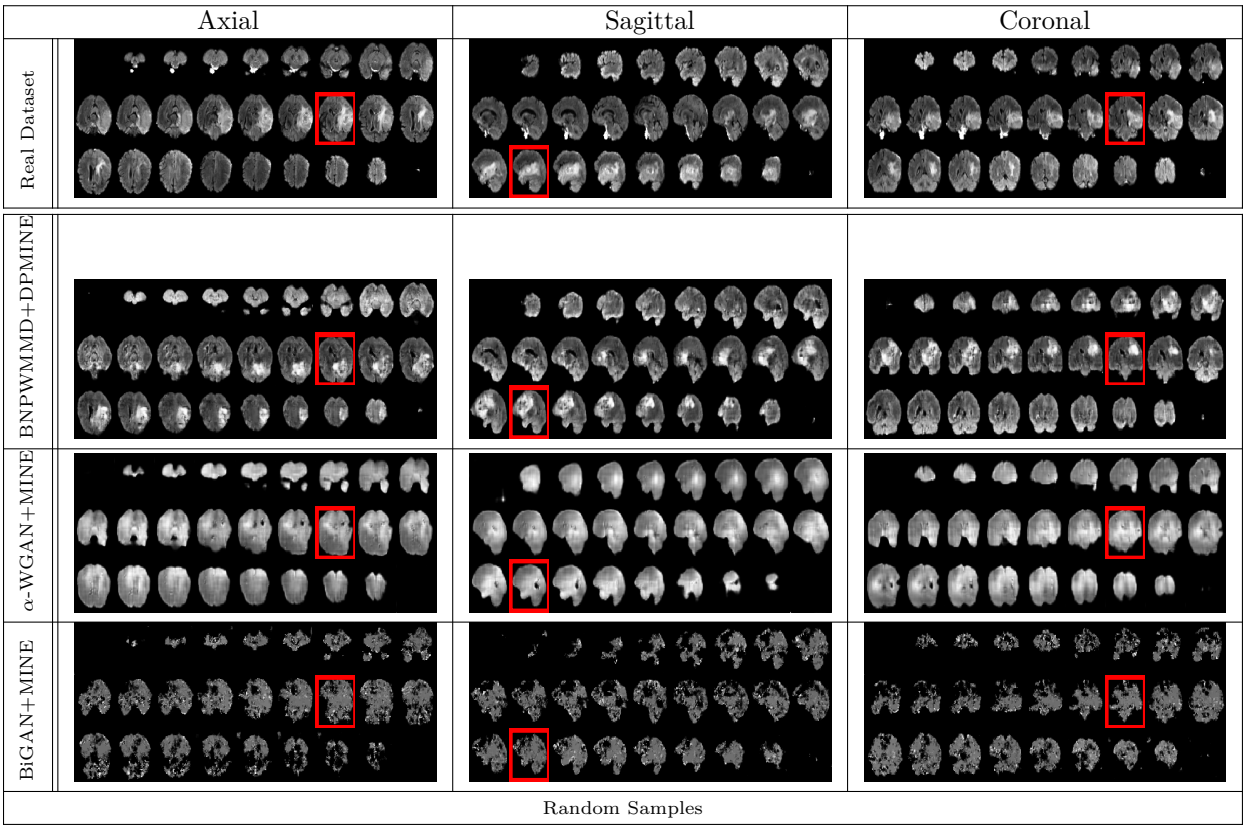


Figure 13: 27 slices of each side of a 3D sample randomly generated from BNPWMMD+DPMINE, BiGAN+MINE, and α-WGAN+MINE after 7500 epochs for BraTS18 example.

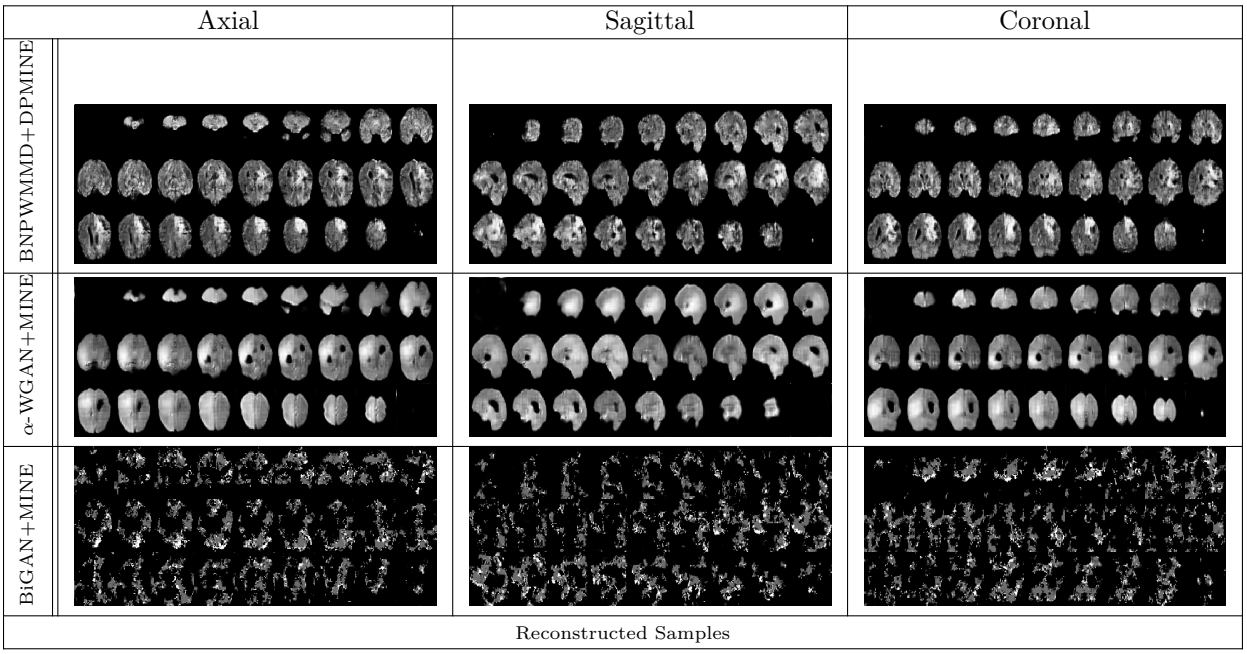


Figure 14: 27 slices of each side of a 3D sample reconstructed by BNPWMMD+DPMINE, BiGAN+MINE, and α-WGAN+MINE after 7500 epochs for BraTS18 example.

Table 3: Comparison of statistical scores for BraTS18 example.

Evaluator		BNPWMMD +DPMINE	α-WGAN +MINE	BiGAN +MINE
FID	Custom Encoder	**0.00197**	0.00692	0.04000
	t-SNE	**25.9317**	59.5596	5572.95
KID	Custom Encoder	**0.00038**	0.00229	0.1589
	t-SNE	**8.8100**	15.1563	2723.97
MMD	Custom Encoder	**0.00075**	0.00410	0.02799
	t-SNE	**0.1988**	0.8150	2.4023
MS-SSIM		**0.75007**	0.74799	0.20349

encoder architecture provided in Table 4 for BiGAN. Additionally, Table 10 presents a simple architecture used for updating the parameters $\{T_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}$ in the DPMINE and MINE calculationsBelghazi et al. (2018). The basic codes[8] of BiGAN+MINE can be found in `https://github.com/gtegner/mine-pytorch.git`.

All architectures have been implemented in PyTorch using the Adam optimizer and a learning rate of 0.0002 on an NVIDIA Tesla V100-SXM2 with 4 GPUs, each with 32GB of RAM. Our code requires 48-72 hours to provide results for the COVID-19 example. For the coil experiments, our code takes about 4-6 hours.

F Limitations

The BNP generative model proposed in this paper assumes that the training data follows the IID assumption. However, this assumption restricts the model's applicability to certain privacy-preserving techniques, such as federated learning, which often rely on non-IID assumptions at the local device level. While this limitation is acknowledged, it is beyond the scope of the current work to address it. In future research, we plan to extend the proposed BNP model to incorporate non-IID assumptions and apply it to federated learning. This extension aims to mitigate issues such as slow convergence and unfair predictions that may arise when applying a global model to unseen data.

[8]It is licensed under the MIT License.

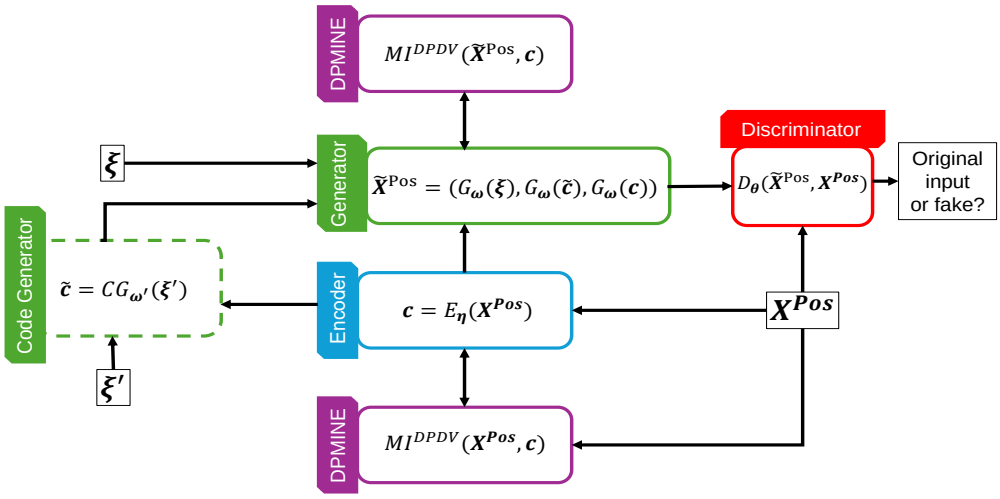


Figure 15: A general flowchart to illustrate the actual operation process of BNPWMMD+DPMINE.

Table 4: Encoder: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BNPWMMD-GAN and α-WGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Convolution	$1 \times 64 \times 64 \times 64$ (Data dimension)	$64 \times 32 \times 32 \times 32$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
Convolution	$64 \times 32 \times 32 \times 32$	$128 \times 16 \times 16 \times 16$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$128 \times 16 \times 16 \times 16$	$128 \times 16 \times 16 \times 16$	-	-	-	-
Convolution	$128 \times 16 \times 16 \times 16$	$256 \times 8 \times 8 \times 8$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$256 \times 8 \times 8 \times 8$	$256 \times 8 \times 8 \times 8$	-	-	-	-
Convolution	$256 \times 8 \times 8 \times 8$	$512 \times 4 \times 4 \times 4$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$512 \times 4 \times 4 \times 4$	$512 \times 4 \times 4 \times 4$	-	-	-	-
Convolution	$512 \times 4 \times 4 \times 4$	$1000 \times 1 \times 1 \times 1$	$4 \times 4 \times 4$	2	1	-

G Broad impacts and safeguards

One potential positive impact of this paper is the advancement of artificial intelligence (AI) techniques in healthcare. The proposed model offers a solution to the challenge of accessing high-quality, diverse medical datasets for clinical decision-making. This can lead to improved accuracy and effectiveness in diagnosing and treating diseases, ultimately benefiting patients and healthcare providers. Additionally, the incorporation of a BNP procedure helps uncover underlying structures in the data and reduce overfitting, enhancing the reliability of the generated samples.

However, the use of AI in healthcare also raises concerns about data privacy and security. It is important to prioritize patient privacy and data protection by implementing strict regulations and protocols to ensure that patient data is anonymized and protected. Additionally, there is a risk of bias in the generated samples, which could lead to disparities in healthcare outcomes if the models are not properly validated and tested on diverse populations. Therefore, transparency and accountability in the development and deployment of these AI models are necessary. Rigorous testing and validation on diverse datasets should be conducted to mitigate bias and ensure fairness. Collaboration between healthcare professionals, AI researchers, and policymakers is essential to establish guidelines and regulations that prioritize patient privacy, data protection, and equitable healthcare outcomes.

Table 5: Code-Generator: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BNPWMMD-GAN and α-WGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Convolution	100 (Sub-latent dimension)	$16 \times 2 \times 5 \times 10$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm2	$16 \times 2 \times 5 \times 10$	$16 \times 2 \times 5 \times 10$	-	-	-	-
Max-pooling	$16 \times 2 \times 5 \times 10$	$16 \times 2 \times 5 \times 5$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	-	-
Convolution	$16 \times 2 \times 5 \times 5$	$32 \times 2 \times 5 \times 5$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm2	$32 \times 2 \times 5 \times 5$	$32 \times 2 \times 5 \times 5$	-	-	-	-
Max-pooling	$32 \times 2 \times 5 \times 5$	$32 \times 2 \times 5 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	-	-
Convolution	$32 \times 2 \times 5 \times 2$	$64 \times 2 \times 5 \times 2$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm3	$64 \times 2 \times 5 \times 2$	$64 \times 2 \times 5 \times 2$	-	-	-	-
Max-pooling	$64 \times 2 \times 5 \times 2$	$64 \times 2 \times 5 \times 1$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	-	-
Flatten	$64 \times 2 \times 5 \times 1$	640	-	-	-	-
Fully-connected	640	1000	-	-	-	-

Table 6: Generator: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BNPWMMD-GAN and α-WGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Transposed convolution	1000 (Latent dimension)	$512 \times 4 \times 4 \times 4$	$4 \times 4 \times 4$	1	0	ReLU
BatchNorm	$512 \times 4 \times 4 \times 4$	$512 \times 4 \times 4 \times 4$	-	-	-	-
Upscale	$512 \times 4 \times 4 \times 4$	$512 \times 4 \times 4 \times 4$	-	-	-	-
Convolution	$512 \times 4 \times 4 \times 4$	$256 \times 8 \times 8 \times 8$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm	$256 \times 8 \times 8 \times 8$	$256 \times 8 \times 8 \times 8$	-	-	-	-
Upscale	$256 \times 8 \times 8 \times 8$	$256 \times 8 \times 8 \times 8$	-	-	-	-
Convolution	$256 \times 8 \times 8 \times 8$	$128 \times 16 \times 16 \times 16$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm	$128 \times 16 \times 16 \times 16$	$128 \times 16 \times 16 \times 16$	-	-	-	-
Upscale	$128 \times 16 \times 16 \times 16$	$128 \times 16 \times 16 \times 16$	-	-	-	-
Convolution	$128 \times 16 \times 16 \times 16$	$64 \times 32 \times 32 \times 32$	$3 \times 3 \times 3$	1	1	ReLU
BatchNorm	$64 \times 32 \times 32 \times 32$	$64 \times 32 \times 32 \times 32$	-	-	-	-
Upscale	$64 \times 32 \times 32 \times 32$	$64 \times 32 \times 32 \times 32$	-	-	-	-
Convolution	$64 \times 32 \times 32 \times 32$	$1 \times 64 \times 64 \times 64$	$3 \times 3 \times 3$	1	1	Tanh

Table 7: Discriminator: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BNPWMMD-GAN and α-WGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Convolution	$1 \times 64 \times 64 \times 64$ (Data dimension)	$64 \times 32 \times 32 \times 32$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
Convolution	$64 \times 32 \times 32 \times 32$	$128 \times 16 \times 16 \times 16$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$128 \times 16 \times 16 \times 16$	$128 \times 16 \times 16 \times 16$	-	-	-	-
Convolution	$128 \times 16 \times 16 \times 16$	$256 \times 8 \times 8 \times 8$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$256 \times 8 \times 8 \times 8$	$256 \times 8 \times 8 \times 8$	-	-	-	-
Convolution	$256 \times 8 \times 8 \times 8$	$512 \times 4 \times 4 \times 4$	$4 \times 4 \times 4$	2	1	Leaky ReLU (negative slope=0.2)
BatchNorm	$512 \times 4 \times 4 \times 4$	$512 \times 4 \times 4 \times 4$	-	-	-	-
Convolution	$512 \times 4 \times 4 \times 4$	$1 \times 1 \times 1 \times 1$	$4 \times 4 \times 4$	2	1	Sigmoid

Table 8: Generator: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BiGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Fully-connected	1000 (latent dimension)	$4 \times 4 \times 4 \times 512$	-	-	-	-
Transposed convolution	$512 \times 4 \times 4 \times 4$	$256 \times 8 \times 8 \times 8$	$4 \times 4 \times 4$	2	1	ReLU
Transposed convolution	$256 \times 8 \times 8 \times 8$	$128 \times 16 \times 16 \times 16$	$4 \times 4 \times 4$	2	1	ReLU
Transposed convolution	$128 \times 16 \times 16 \times 16$	$64 \times 32 \times 32 \times 32$	$4 \times 4 \times 4$	2	1	ReLU
Transposed convolution	$64 \times 32 \times 32 \times 32$	$1 \times 64 \times 64 \times 64$	$4 \times 4 \times 4$	2	1	Tanh

Table 9: Discriminator: A 3D Convolutional Network Architecture for the COVID-19 Dataset used in BiGAN.

Layer	**Input dimension**	**Output dimension**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**
Convolution	$1 \times 64 \times 64 \times 64$ (Data dimension)	$64 \times 32 \times 32 \times 32$	$5 \times 5 \times 5$	2	2	LeakyReLU
Convolution	$64 \times 32 \times 32 \times 32$	$128 \times 16 \times 16 \times 16$	$5 \times 5 \times 5$	2	2	LeakyReLU
Convolution	$128 \times 16 \times 16 \times 16$	$256 \times 8 \times 8 \times 8$	$5 \times 5 \times 5$	2	2	LeakyReLU
Flatten	$256 \times 8 \times 8 \times 8$	131072	-	-	-	-
Concatenate latent-variable	$1000 + 131072$	132072	-	-	-	-
Fully-connected	132072	1024	-	-	-	-
Fully-connected	1024	1	-	-	-	Sigmoid

Table 10: T_γ-network Architecture used in DPMINE and MINE computation.

Layer	**Input Dimension**	**Output Dimension**	**Activation Function**
concatenation	$1000 + 64^3$ (Latent dimension + Data dimension)	263144	-
Linear	263144	400	ReLU
Linear	400	400	ReLU
Linear	400	400	ReLU
Linear	400	1	ReLU