FEATURE ENHANCEMENT WITH DEEP GENERATIVE MODELS IN DEEP BAYESIAN ACTIVE LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

PINAR EZGİ DUYMUŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2022

Approval of the thesis:

FEATURE ENHANCEMENT WITH DEEP GENERATIVE MODELS IN DEEP BAYESIAN ACTIVE LEARNING

submitted by **PINAR EZGİ DUYMUŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Assoc. Prof. Dr. Şeyda Ertekin
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ______________

Assoc. Prof. Dr. Şeyda Ertekin
Computer Engineering, METU ______________

Assoc. Prof. Dr. Tansel Dökeroğlu
Software Engineering, Çankaya University ______________

Date: 01.09.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Pınar Ezgi Duymuş

Signature :

ABSTRACT

FEATURE ENHANCEMENT WITH DEEP GENERATIVE MODELS IN DEEP BAYESIAN ACTIVE LEARNING

Duymuş, Pınar Ezgi
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Şeyda Ertekin

September 2022, 66 pages

Data-intensive models emerge as new advances in Deep Learning take place. However, access to annotated datasets with many data points is not constantly prevalent. This situation emphasizes the need for Active Learning to select the least possible amount of data without compromising the accuracy of the classifier models. Recent advancements occur in Deep Bayesian Active Learning (DBAL), which means incorporating uncertainty of model parameters into a Deep Network. In this work, we present an algorithm that improves the accuracy of a DBAL model in an image classification task. We utilize the representation power of Deep Generative Models by employing their feature extraction capabilities. We obtain improved feature space representation of input data referred to as a latent vector by training a generative model. Instead of using the entire image space in the active learning setting, we demonstrate that utilizing latent space provides better data point selection for the active learning problem, hence obtaining higher accuracy. Furthermore, this study compares different generative models in terms of the ability to capture better feature representation. The informativeness of the data points defines how well an active learning algorithm performs. Therefore, capturing the latent space representation of a data point by extracting the highest information value possible is a significant contribution. We provide comparisons and experiments on different kinds of Generative Models, namely Vanilla Variational Autoencoders (VAEs), Maximum Mean Discrepancy Variational Autoencoders (MMDVAE) and Bidirectional Generative Adversarial Networks (BiGANs). Additionally, Bayesian Active Learning suffers from the Mode-Collapse problem. In order to ease that, we propose a diversity-based query algorithm to enhance the diversity of active points and improve the accuracy of the algorithm.

Keywords: Bayesian Active Learning, Deep Generative Models, Feature Learning, Latent Space Representation, Mode-Collapse Problem

ÖZ

BAYES DERİN AKTİF ÖĞRENMEDE DERİN ÜRETİCİ MODELLER İLE ÖZNİTELİK İYİLEŞTİRME

Duymuş, Pınar Ezgi
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Şeyda Ertekin

Eylül 2022 , 66 sayfa

Derin Öğrenme alanında yeni gelişmeler gerçekleştikçe, veriye yoğun olarak ihtiyaç duyan modeller ortaya çıkmaktadır. Bununla birlikte, birçok veri noktasına sahip işaretlenmiş veri kümelerine ulaşmak sürekli olarak yaygın değildir. Bu durum, sınıflandırıcı modelin doğruluğundan ödün vermeyerek mümkün olan en az miktarda veri noktası seçmek için Aktif Öğrenmenin gerekliliğini vurgulamaktadır. Derin Ağ modeline, model değişkenlerindeki belirsizliğin dahil edilmesi anlamına gelen Derin Bayes Aktif Öğrenme alanında yeni gelişmeler ortaya çıkmaktadır. Bu çalışmada, resim sınıflandırma problemi özelinde, Derin Bayes Aktif Öğrenme modelinin doğruluğunu artıran bir algoritma sunulmuştur. Derin Üretken Modellerin öznitelik çıkarma yeteneklerini kullanarak onların temsil gücünden yararlanılmaktadır. Üretken bir model eğitilerek, saklı vektör olarak adlandırılan girdi veri noktalarının gelişmiş öznitelik uzayı temsili elde edilmektedir. Aktif öğrenme ortamında, tüm görüntü uzayını kullanmak yerine, gizli alan kullanmanın aktif öğrenme problemi için daha iyi veri noktası seçimine imkan sağladığı görülmektedir; dolayısıyla daha yüksek doğruluk elde edilmiş olur. Ayrıca, bu çalışma, daha iyi öznitelik temsilini yakalama yeteneği açısından farklı üretken modelleri karşılaştırır. Veri noktalarının bilgilendiriciliği, aktif bir öğrenme algoritmasının ne kadar iyi olduğunu belirlemektedir. Bu sebeple, bir veri noktasının gizli uzay temsilini, mümkün olan en yüksek bilgi değerini çıkararak yakalayabilmek önemli bir katkıdır. Sıradan Değişimsel Özkodlayıcılar, Maksimum Ortalama Çelişkili Değişimsel Özkodlayıcılar ve İkiyönlü Çekişmeli Üretici Ağlar gibi farklı Üretken Modeller üzerinde karşılaştırmalar ve deneyler sunulmuştur. Ek olarak, Bayes Aktif Öğrenme, Mod-Çökmesi sorunundan etkilenmektedir. Bu problemi azaltmak amacıyla, veri noktalarının çeşitliliğini ve algoritmanın doğruluğunu artırmak için çeşitlilik bazlı bir sorgu algoritması önerilmiştir.

Anahtar Kelimeler: Bayes Aktif Öğrenme, Derin Üretken Modeller, Öznitelik Öğrenme, Saklı Uzay Temsil Etme, Mod-Çökmesi Problemi

To my family.

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my thesis supervisor Assoc. Prof. Dr. Şeyda Ertekin for her priceless support throughout this journey. This thesis would not be possible without her invaluable contributions.

Also, I would like to thank my friends who always support me and my studies. They have always encouraged me whenever I needed them.

Last but not least, I would like to thank my dear parents, Hacı and Yazar, my lovely sister Pelin and my beloved husband Mustafa, who always love and support me unconditionally.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AL	Active Learning
DAL	Deep Active Learning
DBAL	Deep Bayesian Active Learning
BNN	Bayesian Neural Network
PBAL	Pool Based Active Learning
MQS	Membership Query Synthesis
SBS	Stream-Based Sampling
GAN	Generative Adversarial Networks
VAE	Variational Autoencoder
MMDVAE	Maximum Mean Discrepancy Variational Autoencoder
BiGAN	Bidirectional Generative Adversarial Networks
DGM	Deep Generative Model
BALD	Bayesian Active Learning by Disagreement
MCD	Monte Carlo Dropout

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Active Learning (AL) is one of the most influential topics in Computer Engineering research. As the volume of the datasets grows, the need for labelling a large amount of data emerges for Deep Neural Networks (DNNs) to be successful. However, annotating data points in large quantities with manual labelling is not feasible. Due to annotation costs, it is essential to adjust algorithms to achieve high performance with the least labelled data possible. Bayesian Active Learning (BAL) is a promising field to alleviate the labelling problem of unlabelled datasets. BAL provides an information-theoretic perspective to the AL problems. Typical AL algorithms only consider the uncertainty of data points. However, BAL considers the uncertainty of model parameters, leading to a query of more informative data samples. Therefore, fewer data points that represent the entire dataset can be selected within this approach.

BAL is an important research area that is still open to improvements. In recent years, one of the most critical advances in this area is the introduction of Deep Bayesian Networks [1]. With the help of the generalization power of deep networks, Deep Bayesian Active Learning (DBAL) has achieved the top baseline accuracy. However, it poses several difficulties to create a Bayesian approximation of the model parameters in Deep Networks; for example, training such networks is computationally intensive. Therefore, a new technique [2] called Monte Carlo Dropout (MCD) was developed to reduce the network's training time. MCD provides an efficient training performance in terms of time and enables the everyday use of DBAL in the literature. Another drawback of DBAL is that it suffers from the Mode-Collapse problem. In other words, it does not sample diverse, active points since only uncertainty of data points is considered; hence this situation degrades the accuracy.

The work produced during this thesis introduces two main contributions to the DBAL algorithm. The first contribution proposes a feature-based improvement to the DBAL model by utilizing a generative model, which we refer to as GEN-DBAL. The second contribution alleviates the Mode-Collapse problem of DBAL by proposing a diversity-enhancing query selection policy.

GEN-DBAL is a two-stage algorithm. In the first stage, a latent vector representation of the input space is learnt by a Deep Generative Model (DGM). This latent vector is used as the feature vector in the DBAL algorithm instead of utilizing the original input vector. The generative part of the algorithm can be replaced by any model containing an encoder as long as it can capture an informative latent vector. In the second stage, summarized feature space is used in the AL setting of the DBAL algorithm. The improved feature vector enhances the performance of the DBAL model, enabling the selection of more informative data points and leading to higher accuracy in the Active Learning policy. We validate the performance of our proposed model with state-of-the-art data acquisition functions. Furthermore, we provide experiments on three different generative models and compare their feature learning performance on the DBAL algorithm. Our proposed method can be generalized for different DGMs, which can learn informative feature space.

In the experiments, we compare the accuracy results of the baseline model against the GEN-DBAL algorithm using three different DGMs, since accuracy is the most evident metric when measuring the performance of an AL algorithm. Additionally, we provide convergence rate analysis of each algorithm on baseline datasets as the second performance metric. Convergence rate refers to the number of points required to be labelled in an AL algorithm to achieve a target accuracy. The target accuracy depends on the problem that we would want to solve.

Furthermore, we establish a connection between the informativeness of a latent space and the performance of the DBAL algorithm. For this purpose, we compare the generated examples from DGMs to reveal their information preservation capabilities. If a DGM can generate realistic samples, we can deduct that it preserves and extract the most helpful information possible; hence we obtain higher accuracy in the DBAL algorithm. We also analyze the feature spaces obtained from different DGMs and try to differentiate their feature learning abilities. Finally, we perform experiments on the effect of the latent vector size on the algorithm's accuracy.

As our second significant improvement, we propose a diversity-enhancing query algorithm for GEN-DBAL schema to ease the Mode-Collapse problem. In the proposed method, we aim to select more diverse data points while considering the uncertainty of data points. The proposed algorithm mitigates the Mode-Collapse problem and improves the accuracy of GEN-DBAL.

1.2 Contributions and Novelties

Our contributions are as follows:

- To the best of our knowledge, this thesis introduces the first method that combines a Deep Generative Model for feature learning in the DBAL problem setting. We refer to the proposed algorithm as GEN-DBAL and show that it outperforms the typical DBAL algorithm in terms of accuracy.
- We provide experiments using different generative models, namely Vanilla Variational Autoencoders (VAEs), Maximum Mean Discrepancy Variational Autoencoder (MMDVAEs) and Bidirectional Generative Adversarial Networks (BiGANs). Hence, we contribute to the literature by comparing the feature learning performance of different generative models.
- We establish a relationship between the informativeness of latent vectors learnt by generative models and their performance in the active learning setting. We show that more informative latent features improve performance in active query selection.
- We demonstrate the link between the latent vector size of a DGM and the AL algorithm's accuracy.
- We propose a diversity-enhancing query policy to alleviate the Mode-Collapse problem of DBAL, improving the accuracy metric.

The work presented in this thesis has been published in the following conference papers:

- P. E. Çöl and Ş. Ertekin, "Feature Dimensionality Reduction with Variational Autoencoders in Deep Bayesian Active Learning," 2021 29th Signal Processing and Communications Applications Conference (SIU), 2021, pp. 1-4, doi: 10.1109/SIU53274.2021.9477979. [3]

- P. E. Duymuş and Ş. Ertekin, "Alleviating Mode Collapse Problem in Deep Bayesian Active Learning," 2022 30th Signal Processing and Communications Applications Conference (SIU), 2022, pp. 1-4. [4]

1.3 The Outline of the Thesis

The outline of this thesis is as follows. In Chapter 1, we introduce the research problem and our contributions to the literature. In Chapter 2, we provide an extensive literature review about Active Learning, Bayesian Active Learning and applications of Deep Generative Models on Active Learning. In Chapter 3, we describe the related work that is adopted in this study. In particular, we define the Bayesian Active Learning method and provide definitions of Deep Generative Models used in this study. In Chapter 4, we explain the baseline methodology and our proposed approach as a contribution to the baseline model. In Chapter 5, we provide implementation details of baseline DBAL algorithm and proposed GEN-DBAL algorithm. Specifically, we explain the training details of the DBAL algorithm and DGMs. In Chapter 6, we provide experiments and discussions concerning the accuracy performance of baseline and proposed models. Also, we provide experiments with different hyperparameter settings and analyze the latent vector effect in detail. In Chapter 7, we propose an algorithm to alleviate the Mode-Collapse problem in DBAL and further improve the accuracy of GEN-DBAL. Finally, in Chapter 8, we conclude our findings and results along with possible future improvements.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we provide an extensive literature review of the AL methodologies focusing on the area of Bayesian Active Learning. Furthermore, we provide a review of applications of Deep Generative Models in AL problems.

2.1 Active Learning

One of the most critical challenges in Deep Learning architectures is to obtain labelled data. Acquiring labelled data is a costly process that requires manual annotations, which are hand-marked by human experts. Active learning frameworks [5], [6] emerged to address the labelling problem without compromising the accuracy of the machine learning algorithms. The primary goal of an active learning algorithm is to select the least possible amount of data points from an unlabelled pool set and annotate them by a human expert while enhancing or maintaining the machine learning model's performance.

2.1.1 Sampling Strategies

In Active Learning literature, there are three commonly used data sampling strategies, namely Membership Query Synthesis (MQS) [7], Stream-Based Sampling (SBS) [8] and Pool-Based Active Learning (PBAL) [9].

MQS means that the learner queries data points from a pool of artificially generated data points and asks for labels for them; preferably, these data points are generated based on the uncertainty region of data. In other words, the queried instances do not come from the distribution of the pool set. This method usually does not produce reliable queries. However, it might be effective when samples are synthesized from an indisputable world experiment [10].

In the SBS method, instances are queried and immediately decided whether to be annotated or not. This strategy covers the case when the pool set is not readily available because data points are streamed from a source, such as a mobile device which depends on a timetable. Compared to the MQS strategy, the queried instances come from an underlying natural distribution. However, the instance space is not directly known since there is no way of simultaneously accessing the whole pool set. There is a wide range of application areas of SBS; including document classification [11], classification of 3D point clouds [12], Twitter sentiment analysis of stock market [13] etc.

The most common sampling strategy in AL problems is the PBAL strategy. In this method, there is a large set of unlabelled points readily available to be processed and a small set of labelled data points, which is suitable for most real-world datasets [10]. PBAL has access to all the unlabelled data at the beginning compared to SBS. Therefore, the sampling strategy can be formulated by greedily evaluating all unlabelled data points and selecting the best ones according to a query strategy. PBAL has been widely studied in the literature with the various machine learning models, such as Expectation Maximization [14], Linear Regression [15], [16], Support Vector Machines [17], [18], [19]; to name a few. In this thesis, our method also follows the principles of PBAL as our sampling strategy, whose details are described in Chapter 4.

2.1.2 Query Strategies

In AL problems, querying data points to be labelled among the possible choices is crucial for an AL method to perform well. An AL algorithm's performance depends heavily on selecting the adequate query strategy for the task. Three main strategies are extensively studied in the literature, namely Uncertainty-Based Sampling [9], Diversity-Based Approach [20] and Expected Model Change [21]. In addition, there have been studies focusing on hybrid query strategies [22], [23].

The uncertainty-based sampling strategy intends to query samples so that the machine learning model is least certain about its label. More precisely, this strategy favours the most informative samples. Nowadays, Uncertainty-based approaches are heading towards Bayesian Active Learning (BAL) frameworks [1], [24], [25], [26]. In this thesis, the proposed algorithm also employs an Uncertainty-Based query strategy.

The Diversity-Based Approach focuses on selecting data points such that the diversity between selecting data points is maximized, hence acquiring a small representative core-set. Sener et al. [27] proposed a model that uses CNNs as the classification model and formulates the problem as a greedy k-center problem to obtain a core-set. This algorithm is the current state-of-the-art method in the Diversity-Based strategy.

The Expected Model Change strategy targets the instance that would significantly impact the accuracy of the machine learning algorithm when queried. The model parameters' gradient can measure the model's change; hence, this strategy is solely suitable for gradient-based optimization algorithms. In recent years, studies in EMC focused on regression problems [28], [29], [30].

2.2 Bayesian Active Learning

Uncertainty-Based AL methods focus on selecting data points that the machine learning model is least certain. Solving this problem is NP-hard; therefore, several heuristic algorithms that propose approximations exist in the literature. In addition to estimating the uncertainty of data points, it also becomes crucial for AL algorithms to incorporate the model uncertainty. Uncertainty of model parameters can be modelled using Bayesian approaches [31], [32], [33], [34]. Bayesian models are instances of information-theoretic models. Houlsby et al. [35] proposed the Bayesian Active Learning by Disagreement (BALD), which is an acquisition method, filling the gap between mutual information coming from queried instances and model parameters.

The dataset scales in terms of dimension and amount are increasing rapidly; however, regular Bayesian models are insufficient to model high-dimensional data. Hence, a need arose to combine Deep Networks with Active Learning methods, which is not a trivial task. The application of Deep Networks in an AL problem brings several difficulties. First of all, Deep Learning models usually require much data to achieve high accuracy [36]. However, AL algorithms tend to select and label a small amount of data. In addition, AL algorithms rely on model uncertainty in many cases; Deep Networks do not implicitly have that property. Therefore, researchers developed Deep Active Learning (DAL) algorithms that consider model uncertainty. In order to represent model uncertainty in Deep Networks, Deep Bayesian Networks models appeared.

Monte-Carlo Dropout (MCD) method [2] provides a state-of-the-art approach to training a Bayesian Network efficiently. In this method, the parameters of a Deep Network are approximated as a Bayesian Network by applying parameter dropouts on the model during test time. Several times, applying random dropouts on a Deep Network yields an approximation to a Bayesian model. Therefore, MCD integrates Bayesian Neural Networks (BNN) into the AL setting by providing efficient training. One study opposes the BNNs by suggesting that the Ensemble-Based Deep Active Learning algorithm [37] outperforms DBAL in terms of accuracy. On the contrary, Rakesh et al. [38] state that BNNs are superior to Ensemble methods because they are more efficiently trainable than Ensemble-based models. Also, BNNs achieve performances on par with Ensemble algorithms. There is a combination of both approaches, which uses a Bayesian approximation algorithm [39] by using deep probabilistic ensembles.

Even though using a BNN in an AL setting is advantageous, BNNs have some disadvantages. BNNs suffer from the problem of selecting low diversity of data points; hence they suffer from the Mode-Collapse problem. Pinsler et al. proposed a method that enhances the selected data points in BNNs in terms of diversity [40]. They put the problem as a sparse subset approximation of the posterior function of the dataset, hence covering more diverse points. Another method that covers the diversity problem proposes a new acquisition function, BatchBALD [41]. BatchBALD incorporates the joint information of data points in the pool set, resulting in the selection of more diverse instances. There have been hybrid approaches that combine uncertainty information, including the diversity of points. BADGE [25] is an algorithm that estimates the uncertainty of data points using gradient embeddings of the BNN. Afterwards, BADGE acquires a diverse set of points by applying a clustering algorithm similar to K-means on the computed gradients.

To the best of our knowledge, there are no studies in the literature on DBAL that focus on improving the dataset quality using generative models. In this thesis, we propose an algorithmic approach that increases the accuracy of DBAL with the help of Deep Generative Models. By learning latent space representation of the feature space, we improve the quality of data points, hence acquiring an improved accuracy performance.

2.3 Applications of Deep Generative Models in Active Learning

There have been efforts to integrate Deep Generative Models (DGM) in AL problems in Active Learning literature. There are three main methods; generating new samples to increase the number of data points in the unlabelled pool set, using agent-based generators and discriminators to formulate an active learning policy, and learning latent space representation from the dataset to enhance information gained from data instances.

One of the methods is to use the generative capabilities of DGMs to generate new data points and query instances to label from the generated pool set. Zhu et al. [42] reformulated Generative Adversarial Networks (GANs) to generate new samples closer to the decision boundary instead of sampling unlabelled points from an existing pool set. Tran et al. [43] introduce a method that utilizes a BN as a classifier, a Generative model VAE-ACGAN, to generate informative samples and a discriminator to distinguish between real or fake samples. Huijser et al. [44] generates new data points using a GAN model and asks for labels if the point lies on a hypothetical decision boundary. Each iteration adjusts the decision boundary according to the generated samples. Therefore, complex examples lying on a decision boundary are learnt by the active learner with the help of GANs.

Sinha-Ebrahimi et al. proposed an agent-based algorithm called Variational Adversarial Active Learning (VAAL) [45] where the selection of data points is accomplished based on a minimax game played by a VAE and a binary adversarial classifier (discriminator). On the one hand, VAE tries to deceive the discriminator as if all data points belong to the labelled set; on the other hand, the discriminator aims to differentiate between labelled and unlabelled data points.

In a study [46], a Variational Autoencoder (VAE) is trained to learn the latent space representation of data, and a clustering algorithm is applied to the generated latent features in order to obtain a core-set. They aim to increase the accuracy of the active learning algorithm using the ability of VAE's feature representation. Additionally, the training time of the active learning model is improved due to the reduction in the dimensions of feature space.

CHAPTER 3

RELATED WORK

3.1 Deep Bayesian Networks

In this thesis, we focus on the image classification problem in AL. Therefore, we use image datasets, and our methodology relies on Bayesian Networks, which are capable of modelling high-dimensional data. We utilize a Multi-Layer Perceptron (MLP) model as the active learning classifier in our framework. In order to perform Bayesian Approximation on the MLP model, we follow the formulation provided by the authors in [24]. We utilize an MLP classifier as a difference from the original work and use a CNN classifier. The difference in model selection is due to CNN being suitable for image data with spatial information; however, we utilize a latent vector which does not contain spatial features. For consistency in our experiments and comparisons, it is more convenient to use an MLP model instead of a CNN.

First of all, model parameters of MLP are members of a Gaussian prior distribution $\omega \sim p(\omega)$; $\omega = \{\Omega_1, ..., \Omega_m\}$ where m is the number of model parameters. Also, we define a likelihood for the classification model as in Equation (3.1).

$$p(y = c|x, \omega) = softmax(f^{\omega}(x)) \tag{3.1}$$

In order to perform approximate Bayesian inference from the model f, which corresponds to the output function of the MLP model in our case, we adopt a technique called Monte-Carlo Dropout (MC-Dropout) [2], [1]. Dropout regularization is applied to every dense layer on training and test time. Applying the MC-Dropout technique yields a distribution $q_{\theta}(\omega)$ such that Kullback-Leibler to true model posterior

$p(\omega|D_L)$ is minimized where L is the set of labelled data points. Finally, we deduct $p(y = c|x, D_L)$ as in Equation (3.2) where $\hat{\omega}_t$ corresponds to the estimation of $q^*_\theta(\omega)$.

$$\begin{aligned} p(y = c|x, D_L) &= \int p(y = c|x, \omega)p(\omega|D_L)d\omega \\ &\approx \int p(y = c|x, \omega)q^*_\theta(\omega)d\omega \\ &\approx \frac{1}{T}\sum_{t=1}^{T} p(y = c|x, \hat{\omega}_t) \end{aligned} \tag{3.2}$$

3.2 Deep Generative Models

This section provides background information about the state-of-the-art generative models in the literature focusing on feature learning. The selected models are suitable for unsupervised feature learning tasks since we do not have access to a large set of annotated data points in an active learning setting. We provide definitions of the following DGMs in this section; Vanilla Variational Autoencoders (Vanilla-VAE) with Elbo loss objective, Maximum Mean Discrepancy VAE (MMDVAE) with Info-VAE loss objective and Bidirectional GANs (BiGANs) which has adversarial feature learning capabilities.

3.2.1 Vanilla VAE

Variational Autoencoders are from the family of Latent Variable Generative Models, meaning they have two main usages: learning a latent representation of the feature space and generating new data points. VAEs are composed of an encoder and a decoder which are trained jointly. Encoder maps high dimensional input data $x \in X$ into a lower dimensional latent space $z \in Z$ while preserving as much information as possible. The Decoder reconstructs the input space from latent space variables with the least amount of loss possible. The Encoder generates a mean vector μ and a standard deviation vector ρ which are sampled into a latent variable z. The Decoder consumes z and generates new data points.

$$\mathcal{L}_{ELBO} = \mathbf{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)) \tag{3.3}$$

During the training of a VAE model, ELBO loss function is utilized [47] as described in Equation (3.3). $q(z|x)$ corresponds to Encoder and represents the conditional distribution of latent space variable $z \in Z$ conditioned on original feature space $x \in X$. $q(z|x)$ can be referred to as amortized inference distribution in different contexts. $p(x|z)$ corresponds to the conditional distribution representing the Decoder part of the VAE. $p(z)$ is a Gaussian prior distribution of feature space z. The left-hand side of the Equation (3.3) maximizes the expected value of the Decoder's log-likelihood, which can be referred to as the reconstruction term. Whereas the right-hand side of the Equation minimizes Kullback-Leibler divergence [48] between $p(z)$ and the amortized inference distribution. In a way, it acts like a regularization term.

3.2.2 MMDVAE

Maximum Mean Discrepancy VAE (MMDVAE) is a member of the Information Maximization VAE family. In particular, it offers an information-theoretical improvement. MMDVAE addresses the shortcomings of the VanillaVAE model. In particular, the optimization objective of MMDVAE is different than a typical VanillaVAE [49]. In other words, the loss function of MMDVAE is reformulated to include an information maximization term as the main difference against the ELBO loss function.

One of the main problems with VanillaVAE is that it might focus on fitting data by disregarding variational inference if two objectives conflict. In other words, the reconstruction objective surpasses the regularization term, which is essential for learning latent features. Another problem is that VAE fails to extract latent vector representation when learning $p(x|z)$ is more dominant over $q(z|x)$ [50].

In order to overcome these challenges, Zhao et al. proposed a new loss function for training a VAE, which is referred to as InfoVAE. First, they start with obtaining an equivalent form of ELBO loss function in Equation (3.4).

$$\mathcal{L}_{ELBO} = -\mathbf{E}_{p(z)}[D_{KL}(q(x|z)||p(x|z))] - D_{KL}(q(z)||p(z)) \tag{3.4}$$

Zhao et al. modify the ELBO loss function by introducing a scaling parameter λ to the Kullback-Leibler divergence between $q(z)$ and $p(z)$. Also, they integrate a mutual information term between x and z as observed in Equation 3.5.

$$\mathcal{L}_{InfoVAE} = -\mathbf{E}_{p(z)}[D_{KL}(q(x|z)||p(x|z))] - \lambda D_{KL}(q(z)||p(z)) + \alpha I(x; z) \tag{3.5}$$

3.2.3 BiGAN

Bidirectional Generative Adversarial Networks [51] (BiGANs) are a special type of GAN model. Their dissimilarity to a regular GAN model is that BiGAN also captures inverse data mapping. In other words, it can capture the dataset's latent space representation while preserving other properties of GANs. A typical GAN framework [52] is composed of a generator G and a discriminator D. The training objective of a GAN is formulated such that G aims to generate realistic data points from an arbitrary latent distribution to trick D, whereas D tries to distinguish between real samples and fake samples generated by G. Let $p_x(x)$ be the data distribution of the points $x \in X$ and $p_z(z)$ be the fixed latent space distribution of $z \in Z$. The minimax objective $\underset{G}{min}\ \underset{D}{max}\ \mathcal{V}_{GAN}(D, G)$ of GAN can be formulated as in Equation (3.6).

$$\mathcal{V}_{GAN}(D, G) = \mathbf{E}_{x \sim p_x}[\log D(x)] + \mathbf{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{3.6}$$

In addition to a generator and a discriminator, BiGANs include an encoder network as well in order to generate latent vector $z \in Z$ from input space $x \in X$. Thus, there are two main modifications to the GAN model in order to make it a BiGAN model:

- An encoder network $E : X \mapsto Z$ is included in the model to map input space into latent space.

- Discriminator model D is modified to take input from the encoder model. Therefore, D aims to predict $P_D(Y|x, z)$ where $Y = 1$ if x is sampled from data distribution p_x, and $Y = 0$ if x is sampled from $G(z)$.

As a result, the training objective of BiGAN can be formulated as $\underset{G,E}{min}\ \underset{D}{max}\ \mathcal{V}_{BiGAN}(E, D, G)$ where $\mathcal{V}_{BiGAN}(E, D, G)$ is defined in Equation (3.7).

$$\mathcal{V}_{BiGAN}(E, D, G) = \mathbf{E}_{x \sim p_x}[\log D(x, E(x))] + \mathbf{E}_{z \sim p_z}[\log(1 - D(G(z), z))] \quad (3.7)$$

CHAPTER 4

METHOD

4.1 Active Learning Loop

In this work, we employ the Pool Based Active Learning (PBAL) [9] as an active learning process. In this method, there exists an unlabelled data set U originally. Labelled data set L is initially acquired from U and labelled by an $Annotator$. A machine learning model, which is a Bayesian Multi-Layer Perceptron (M_{BMLP}) model in this study, is trained using L. Afterwards, an Acquisition Function $\alpha(x,\, M_{BMLP})$ selects data points from the unlabelled set U where $x \in U$. Selected data points are annotated and included into L, then M_{BMLP} is retrained. This process continues until an annotation budget of B is reached. The process is visualized in Figure 4.1.

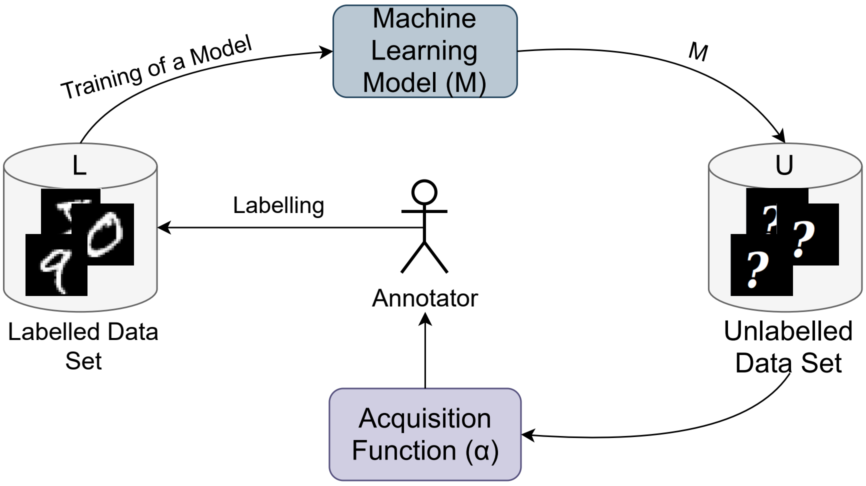


Figure 4.1: Pool Based Active Learning

In the regular PBAL algorithm, data points are queried from the pool set by selecting one point at a time. However, this query strategy is not efficient enough to use in experimental settings. Therefore, we adopt a more feasible approach, namely Batch Mode Active Learning [53], where data points are queried in the size of batch number b in each iteration until a query budget of B is reached.

4.2 Acquisition Functions

An acquisition function in an AL algorithm serves as a measure of uncertainty. Since there is not a single way of measuring the uncertainty, we utilize five different state-of-the-art acquisition functions in our experiments to estimate the uncertainty of data points. Their details are explained in this section.

4.2.1 Entropy

Data points that maximize the predictive entropy are selected. The points are selected such that the machine learning model is the least certain. This method is also known as Shannon Entropy [54] as can be observed in Equation (4.1). We can reformulate this Equation in order to retrieve the Entropy Acquisition function. In Equation (4.1) and (4.2), p corresponds to the posterior distribution of the model conditioned on a single data point $x \in U$ and L where L corresponds to the labelled dataset.

$$\mathbf{H}[y|x, L] = -\sum_{c} p(y = c|x, L) \log p(y = c|x, L) \tag{4.1}$$

$$\alpha_{entropy}(x, L) = -\sum_{c} p(y = c|x, L) \log p(y = c|x, L) \tag{4.2}$$

4.2.2 Variation Ratios

In this acquisition function [55], the aim is to maximize Variation Ratios which is defined in Equation (4.3). Similar to the Entropy acquisition function, it measures the lack of confidence of the model over the data points.

$$\alpha_{VarRatio}(x, L) = 1 - \underset{y}{max} p(y|x, L) \tag{4.3}$$

4.2.3 BALD (Bayesian Active Learning by Disagreement)

When choosing data points, BALD aims to maximize information gain between predictions and model posterior [35] as described in Equation (4.4). In other words, it represents how much a relation exists between the model parameters and the model predictions for a data point. We can reformulate Equation (4.4) by replacing the $\mathbf{H}$ by $\alpha_{entropy}$. In Equation (4.5), the left side corresponds to the regular entropy, which is expected to be high, whereas the right side is the expected value of the entropy of the model over the posterior of the model parameters. It is preferable to have a low value for the right side of the Equation.

$$\mathbf{I}[y, \omega|x, L] = \mathbf{H}[y|x, L] - \mathbf{E}_{p(\omega|L)}[\mathbf{H}[y|x, \omega] \tag{4.4}$$

$$\alpha_{BALD} = \alpha_{entropy}(x, L) - \mathbf{E}_{p(\omega|L)}[\alpha_{entropy}(x, \omega)] \tag{4.5}$$

4.2.4 Mean Standard Deviation

In the Mean Standard Deviation acquisition function, the standard deviation of the model posterior is calculated as in Equation (4.6). And then, it is averaged over all classes as in Equation (4.8). The aim is to select data points that maximize this function. It is a recent technique adopted in the literature [56], [57].

$$\sigma_c(x, \omega) = \sqrt{\mathbf{E}_{q(\omega)}[p(y = c|x, \omega)^2] - \mathbf{E}_{q(\omega)}[p(y = c|x, \omega)]^2} \tag{4.6}$$

$$\alpha_{MeanStd}(x, \omega) = \frac{1}{C} \sum_{c} \sigma_c(x, \omega) \tag{4.7}$$

4.2.5 Uniform

It samples data points from a uniform distribution, equivalent to randomly selecting a data point from an unlabelled pool set.

$$\alpha_{uniform}(x) = uniform(0, 1) \tag{4.8}$$

4.3 Baseline Model: Deep Bayesian Active Learning - DBAL

What makes an active learning algorithm a "Deep" AL is employing a Deep Network as the machine learning model. We utilize a Multi-Layer Perceptron (MLP) as the classifier model in this work. In order to approximate the MLP model into a Bayesian one, we adopt a technique used in the literature; MC Dropout [2]. From now on, we will refer to the Bayesian model as BMLP. During the estimation of the posterior function of the BMLP model, the dropout regularization is applied to the model parameters during test time. This process is repeated K times to obtain K different deep networks. As a result, the model posterior p is mapped to a Bayesian approximation as in the Equation (4.9). $\hat{\omega}_k$ corresponds to dropout distribution of model parameters [58]. Posterior functions of K different models with dropout are averaged over to estimate p. The baseline algorithm is referred to as DBAL in the rest of this paper as described in Algorithm 1.

$$p(y = c|x, L) = \frac{1}{K} \sum_{k=1}^{K} p(y = c|x, \hat{\omega}_k) \tag{4.9}$$

The baseline DBAL model is composed of four parts:

- U: A large unlabelled dataset, it contains data points of amount N_u.
  $U = \{x_j\}_{j=1}^{N_u}, x_j \in X.$

- L: Labelled dataset, it contains data points of amount N_l.
  $L = \{x_j, y_j\}_{j=1}^{N_l}, x_j \in X, y_j \in Y.$

- A: Annotator, it marks labels to data points queried from U.
  $A : X \to Y$.

- M_{BMLP}: A Bayesian Multi-Layer Perceptron model as a classifier. Also, its posterior is used to query data points from the set U.
  $M_{BMLP} : X \to Y$.

Algorithm 1 Active Learning Algorithm of DBAL

Input U - unlabelled dataset, A - annotator, B - active learning budget, n_l - initial number of labelled data points, α - acquisition-function, n_{batch} - batch size, K - number of forward passes in MC-Dropout

```
Query data points randomly such that $\{x^j\}_{j=1}^{n_l} \in U$
Annotator $A$ marks selected data points such that $\{y^j = A(x^j)\}_{j=1}^{n_l}$
$L \leftarrow \{x^j, y^j\}_{j=1}^{n_l}$
$b \leftarrow n_l$
while $b < B$ do
    $M_{BMLP} \leftarrow$ initialize the model
    Train $M_{BMLP}$ model on dataset $L$
    for $k \leftarrow 1$ to $K$ do
        $\hat{M}_{BMLP}^k \leftarrow$ apply MC-Dropout on $M_{BMLP}$
    end for
    for $x_j \in U$ do
        $\hat{M}_{BMLP} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \hat{M}_{BMLP}^k(x_j))$        ▷ uncertainty metric
    end for
    $S \leftarrow AcquisitionStep(\alpha, n_{batch}, M_{BMLP}, U)$
    for $x_s \in S$ do
        $y_s \leftarrow A(x_s)$
        Move $(x_s, y_s)$ data point from $U$ to $L$
    end for
    $b \leftarrow b + n_{batch}$
end while
```

```
Algorithm 2 Acquisition Step
    Input $\alpha$ - acquisition function, $n_{batch}$ - batch size, $M$ - Classifier Model, D - Dataset

  return $argmax_{d_i \in D | 1 \leq i \leq n_{batch}}(\alpha(D, M))$
```

4.4 Proposed Approach: GEN-DBAL

In our proposed approach, we formulate a two-stage algorithm which we refer to as GEN-DBAL. In the first stage of the algorithm, we utilize a Deep Generative Model (DGM) to extract the latent vector representation of the original input space. Extracted latent vectors enhance the informativeness of the dataset. Therefore, the active learner is trained using the latent space instead of the input space in the second stage of the algorithm. There are two main conditions regarding whether a DGM is suitable for GEN-DBAL or not:

- The model should support backwards inference; in other words, the latent vector of the model should be accessible.
- The latent space should be informative enough to contribute to the accuracy of GEN-DBAL.

The architecture of the proposed model can be viewed in Figure 4.2. The active learning policy of the GEN-DBAL model works as described in Algorithm 3.

The proposed GEN-DBAL model is composed of six main parts:

- U: A large unlabelled dataset, it contains data points of amount N_u.
  $U = \{x_j\}_{j=1}^{N_u}, x_j \in X.$
- L: Labelled dataset, it contains data points of amount N_l.
  $L = \{x_j, y_j\}_{j=1}^{N_l}, x_j \in X, y_j \in Y.$
- A: Annotator, it marks labels to data points queried from U.
  $A : X \to Y.$

- $M_{encoder}$: Encoder model, which is obtained from the training of a Generative model. It is used to reduce the dimensions of the input vector.
  $M_{encoder} : X \to Z$.

- $M_{decoder}$: It generates new images using the latent vector.
  $M_{decoder} : Z \to X$.

- M_{BMLP}: A Bayesian Multi-Layer Perceptron model as a classifier. Also, its posterior is used to query data points from the set U.
  $M_{BMLP} : Z \to Y$.

Algorithm 3 Active Learning Algorithm of GEN-DBAL

Input: U - labelled dataset, A - annotator, B - active learning budget, n_l - initial number of labelled data points, α - acquisition-function, n_{batch} - batch size, K - number of forward passes in MC-Dropout, n_{latent} - size of the latent space in Generative Model, n_{hidden} - size of hidden vector in Generative Model

```
                                                        ▷ Stage 1
$M_{GEN}$ ← create model using parameters of($n_{latent}, n_{hidden}$)
$M_{encoder}, M_{decoder} \leftarrow M_{GEN}$
Train $M_{GEN}$ model on $U$
$Z \in \emptyset$
for $x_j \in U$ do
    $z_j \leftarrow M_{encoder}(x_j)$
    Add $z_j$ to set $Z$
end for
                                                        ▷ Stage 2
DBAL($Z, A, B, n_l, \alpha, n_{batch}, K$)    ▷ Call DBAL algorithm with unlabelled set $Z$
```

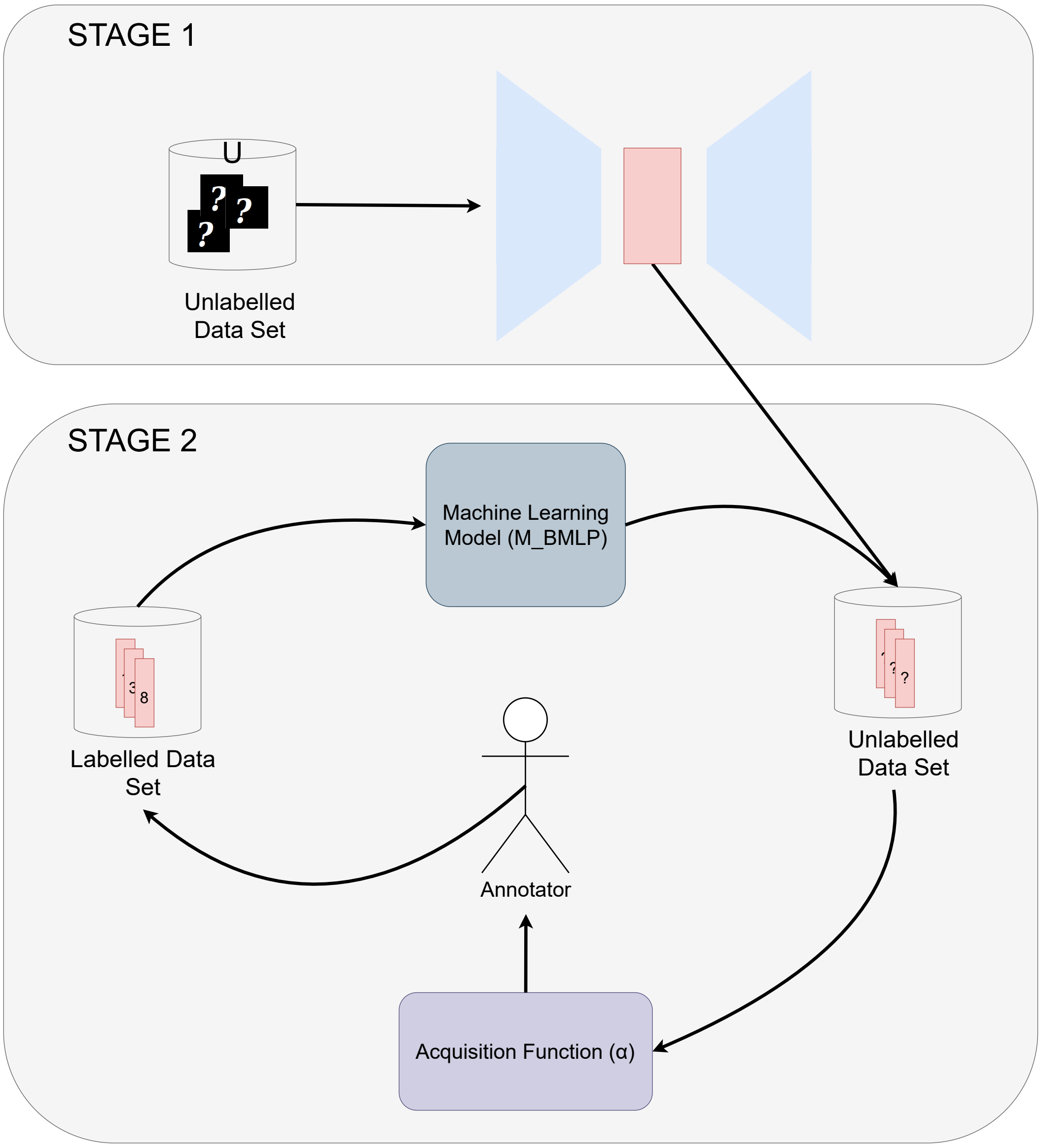


Figure 4.2: GEN-DBAL Architecture

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Multi Layer Perceptron

The network architecture of the BMLP model is illustrated in Figure 5.1. The same structure is used in all experiments for consistency. During the training of the BMLP model, the ReLU activation function is applied after every Dense Layer except the last dense layer. Adam optimizer and Cross-Entropy loss function are utilized. In the first two dense layers, a dropout regularization is applied on 25% of model weights, whereas on the third layer, 50% of model parameters are dropped.

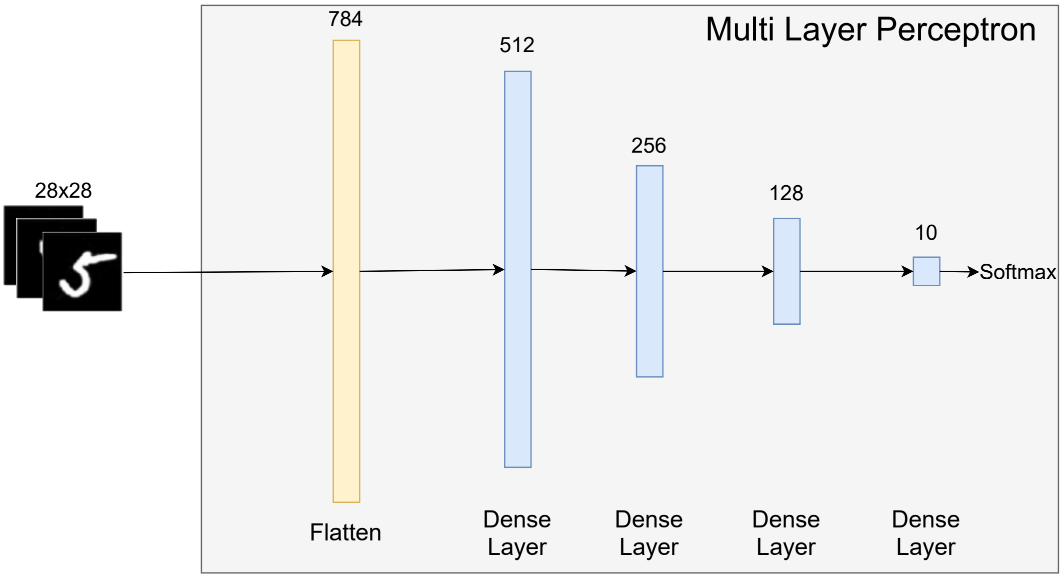


Figure 5.1: The Architecture of Multi Layer Perceptron (MLP) Model

5.2 Active Learning Scheme

For the active learning scheme of baseline model DBAL, we utilize the BMLP model, whose structure is visualized in Figure 5.1. In order to approximate this model to a Bayesian one, MC-Dropout [1], [2] methodology is adopted. In particular, parameter dropouts are applied on both training and test time. During test time, K different Dropout Regularizations are applied to accomplish Bayesian approximation.

GEN-DBAL model is composed of two stages. In the first stage, we utilize a generative model to learn the latent space representation of the dataset. In the second stage, we operate the baseline DBAL scheme on the informative latent space in place of the original input space. For the first stage of GEN-DBAL, we experiment on three different Deep Generative Models, namely Vanilla VAE, MMDVAE and BiGAN, whose details are explained in Chapter 3. All of these models include an Encoder and a Decoder part; in addition to those, there is a Discriminator model in BiGAN as well. For consistency, we keep Encoder and Decoder parts the same for all DGMs.

Hyperparameters used in both baseline DBAL algorithm and proposed GEN-DBAL algorithm are provided in Table 5.1. The parameters belonging to the DBAL part of both models are kept the same for consistency. We use an active learning budget of B for 1000 data points. We query a randomly selected but balanced set of 20 data points as the initial pool set. The active learning framework queries another 10 data points in each iteration using the selected acquisition function. Those selected points are labelled by Annotator and included in the pool set. For MC-Dropout, we utilize K as 100 for the number of different dropout regularizations applied during test time. The latent vector's size is 64, and hidden layers in the generative model are selected as [512, 256].

5.3 The Training Details of Deep Generative Models in GEN-DBAL Algorithm

We utilize three different DGMs to perform feature learning on the dataset. The selected models are Vanilla VAE, MMDVAE and BiGAN. For both Vanilla VAE and MMDVAE models, we utilize the same architecture as depicted in Figure 5.2. For

Table 5.1: MODEL HYPER PARAMETERS

Parameters	Model	
	DBAL	GEN-DBAL
B	1000	1000
n_l	20	20
n_{batch}	10	10
K	100	100
n_{latent}	-	64
n_{hidden}	-	512, 256

the architecture of the BiGAN model, we utilize the model as visualized in Figure 5.3. We analyze the feature learning capability of those models and compare their performances in the Deep Bayesian Active Learning setting.

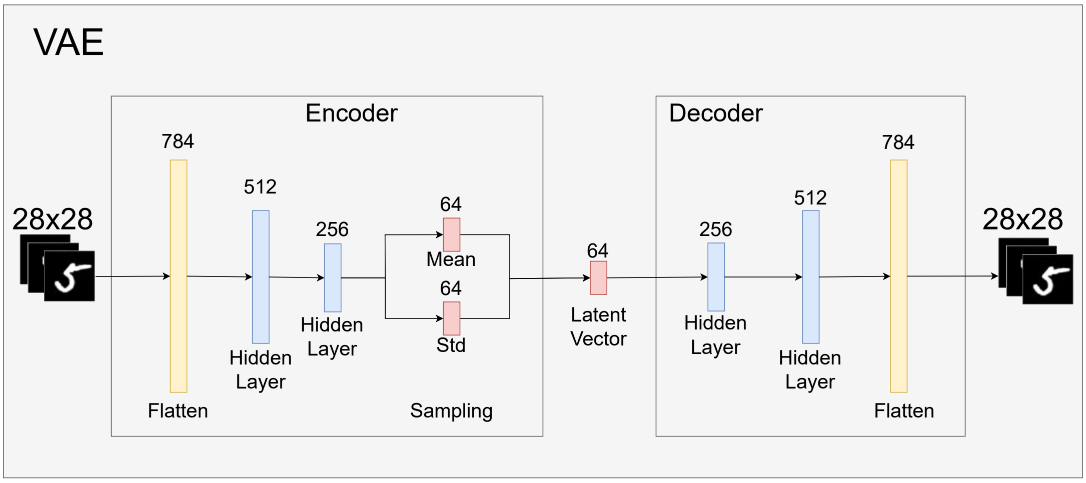


Figure 5.2: The Network Architecture of Variational Autoencoder Models

We provide implementation details of these models and mention their architectural differences. The hyper-parameters for generative models are provided in Table 5.2. The same architecture is kept for the Encoder and Decoder parts in all three models. Since the BiGAN model includes a Discriminator model, its architecture is provided in Figure 5.3. VanillaVAE and MMDVAE are trained for 50 epochs, whereas BiGAN

Table 5.2: HYPER PARAMETERS OF GENERATIVE MODELS

Hyper Parameters	Model		
	Vanilla VAE	MMDVAE	BiGAN
epochs	50	50	100
batch size	256	256	128
activation of Decoder	sigmoid	sigmoid	sigmoid
activation of Discriminator	-	-	sigmoid
activation of dense layers	ReLU	ReLU	Leaky ReLU (slope=0.2)

is trained for 100 epochs due to implicit learning of the latent vector. The Sigmoid activation function is applied in the decoder part of all models. Similarly, in the Discriminator part of BiGAN, the Sigmoid activation function is utilized. For the activation of dense layers, Vanilla VAE and MMDVAE use ReLU; however, we used Leaky ReLU with a slope of 0.2 for BiGAN to be consistent with the original BiGAN paper [51]. Additionally, in the BiGAN model, we use Batch Normalization with a momentum of 0.8 on dense layers.

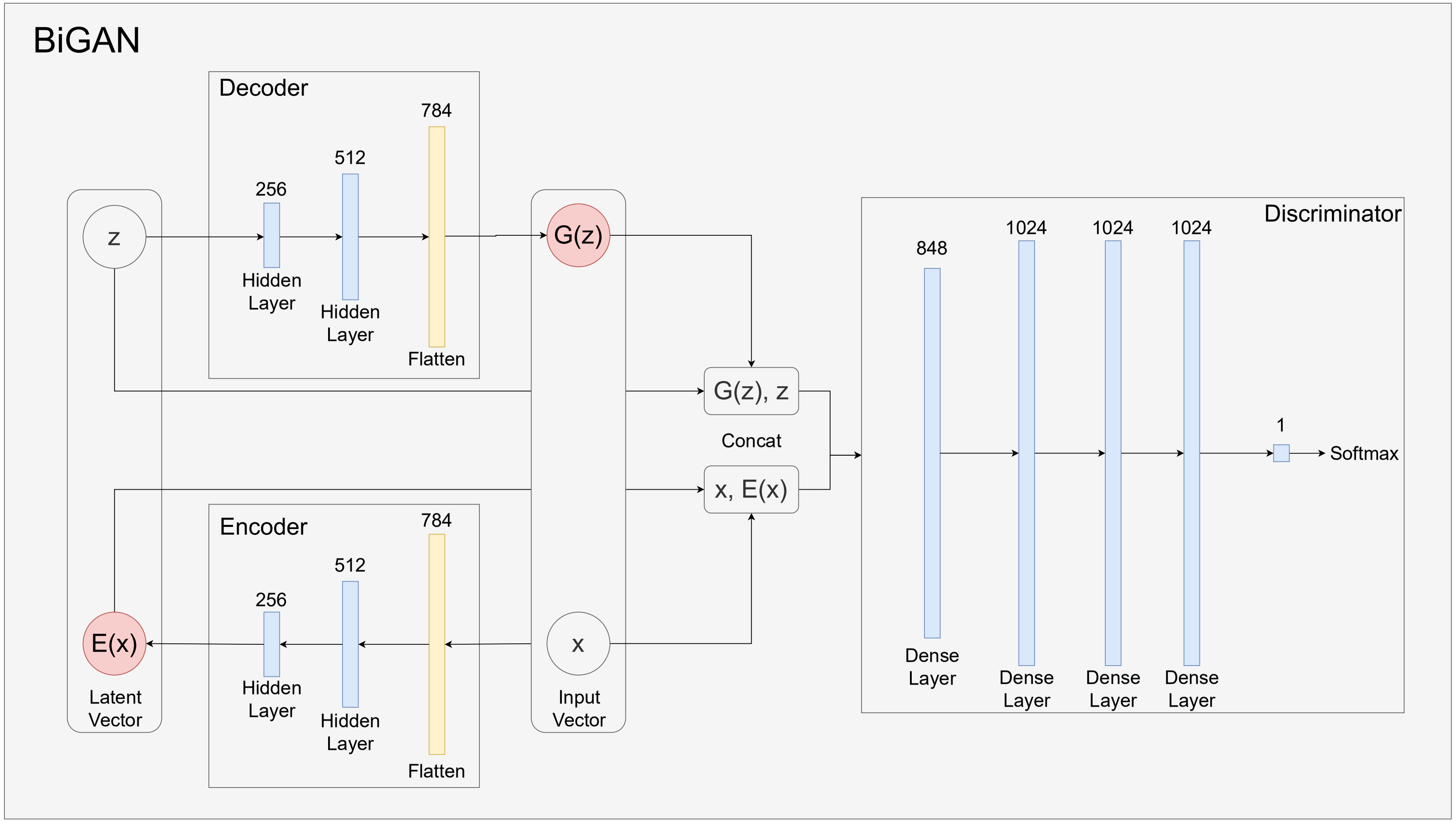


Figure 5.3: The Network Architecture of Bidirectional Generative Adversarial Networks (BiGANs)

CHAPTER 6

EXPERIMENTS AND EVALUATION

6.1 Datasets

6.1.1 MNIST

MNIST [59] contains hand-written digits with 10 different class labels. This dataset contains black and white images with dimensions of 28 × 28 pixels. It comprises a training set containing 60,000 samples and a test set containing 10,000 samples. Sample images from the MNIST training set are displayed in Figure 6.1. It is a relatively simple dataset compared to other datasets we use in our experiments.

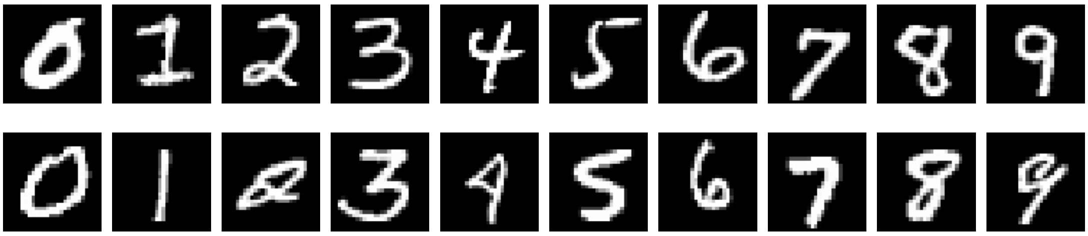


Figure 6.1: Sample Images from MNIST Training Set

6.1.2 Fashion MNIST

Fashion MNIST [60] contains fashion clothing items with 10 different class labels. The images in the dataset are black and white with dimensions of 28 × 28 pixels. It comprises a training set containing 60,000 samples and a test set containing 10,000 samples. Sample images from the Fashion MNIST training set are displayed in Figure

6.2. It is a more challenging dataset compared to the MNIST digit dataset. However, fashion items in the images are easily recognizable by the human eye.

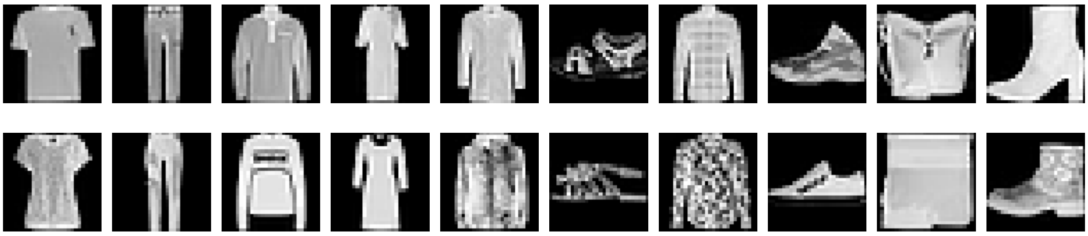


Figure 6.2: Sample Images from Fashion MNIST Training Set

6.1.3 Organ AMNIST

Organ AMNIST [61] contains organ images with 11 classes of dimensions (28 × 28). It is from a family of Medical datasets, Med MNIST. It comprises a training set containing 34,581 samples, a validation set containing 6,491 images and a test set containing 17,778 samples. Sample images from the Organ AMNIST training set are displayed in Figure 6.3. It is a complex dataset compared to MNIST and Fashion MNIST because images contain many details and very complex backgrounds. Also, this dataset is highly imbalanced compared to others.

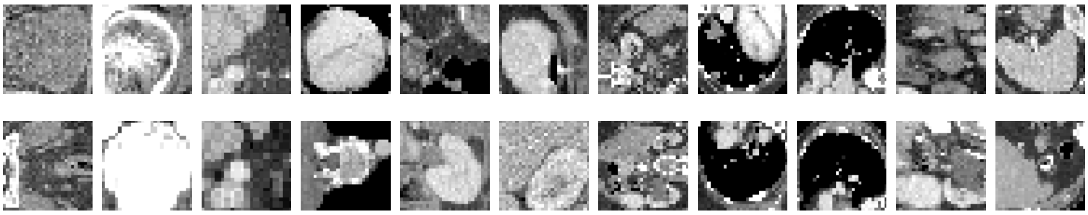


Figure 6.3: Sample Images from Organ AMNIST Training Set

6.2 Comparison of Deep Bayesian Active Learning Against Full Dataset Training

In this section, we compare the baseline DBAL algorithm against the Multi-Layer Perceptron (MLP) model trained with the full dataset. We used MNIST dataset in the experiments. During the training of the DBAL algorithm, we utilize Bayesian MLP as the classifier model, and for the full training, we utilize a regular MLP classifier. The same model architecture is used in all experiments, as explained in Section 5.1. The accuracy results are displayed in Figure 6.4. As expected, training with a full dataset produces a higher accuracy value of 98%. In comparison, the DBAL model results in an accuracy of approximately 94% for all acquisition functions except the Uniform acquisition function. The performance difference is due to the number of data points used in training. DBAL model is trained with 1000 data points, and full training is performed with 60000 data points.

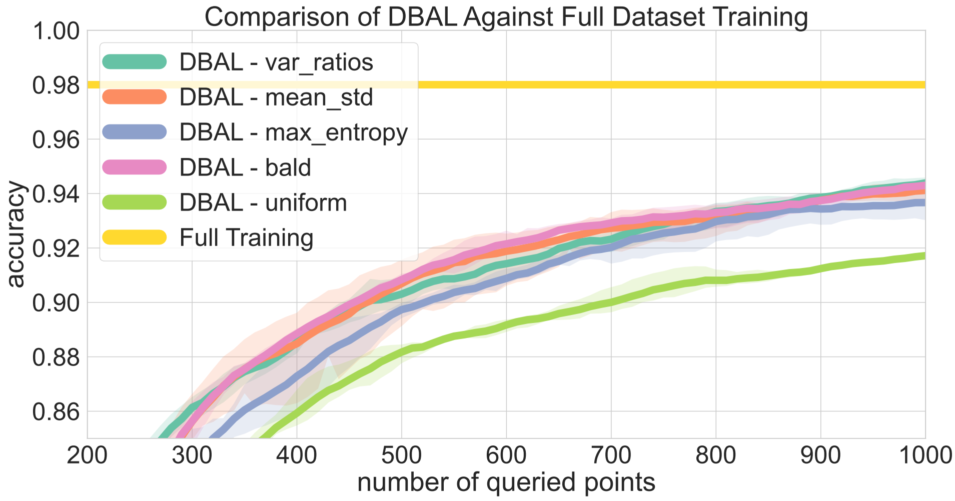


Figure 6.4: Comparison of DBAL Algortihm Against Full Training for MNIST dataset

6.3 Accuracy Results

This section provides test accuracy comparisons between baseline model DBAL and our proposed approach GEN-DBAL. We perform experiments with three different DGMs and compare the accuracy results against the baseline DBAL model and with each other. DGM-based proposed models are VanillaVAE-DBAL, MMDVAE-DBAL and BiGAN-DBAL. Experiments are performed on three different datasets: MNIST, Fashion MNIST and Organ AMNIST. We validate our results on five acquisition functions; Entropy, BALD, Mean Standard Deviation, Variation Ratios and Uniform.

In Table 6.1, we provide accuracy result of all model comparisons. For the MNIST dataset, three configurations perform equally well; VanillaVAE-DBAL with Variation Ratios, MMDVAE-DBAL with BALD and MMDVAE-DBAL with Variation Ratios. Overall, MMDVAE-DBAL with Variation Ratios consistently outperformed other configurations for all datasets in terms of accuracy.

MNIST

In Figure 6.5, we provide accuracy results for MNIST dataset. VanillaVAE-DBAL and MMDVAE-DBAL models outperformed the baseline model DBAL and performed very similarly in terms of accuracy. However, the BiGAN-DBAL model is the least performing model, which could not exceed the performance of the baseline model DBAL. It appears that accuracy loss in the BiGAN-DBAL setting is due to the implicit learning property of the BiGAN model. Implicit learning is a phenomenon which occurs during the training of the BiGAN model. Encoder and Generator parts of the BiGAN network do not have direct access to outputs of each other; therefore, the parameters of both parts are learnt with the help of the discriminator model. This situation leads to learning a poor representation of latent space in BiGAN models.

Fashion MNIST

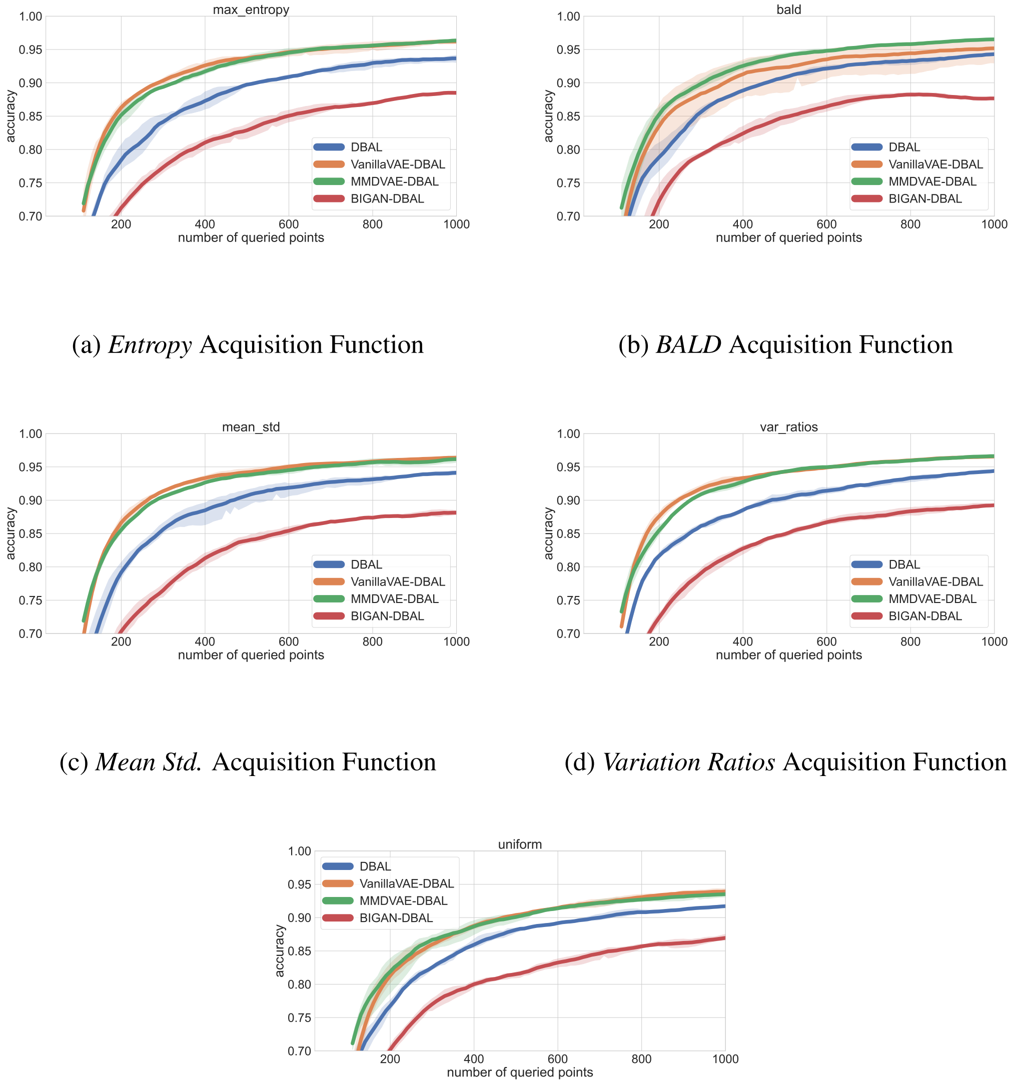
In Figure 6.6, we provide accuracy comparison between proposed models and the baseline model for Fashion MNIST dataset. In this case, the MMDVAE-DBAL model outperformed all other models. However, the VanillaVAE-DBAL model could not perform more satisfactorily than the baseline DBAL model. The reason it failed in this task is the deficiency of ELBO loss function, which is minimized for the training of the Vanilla VAE model. Reconstruction term has become dominant over regularization term, and therefore Vanilla VAE could not learn a proper representation of latent space; hence it generates a deficiency in accuracy performance. BiGAN-DBAL model either performs very similarly to baseline or falls back in terms of accuracy metric for the same reason as explained for MNIST dataset.

Organ AMNIST

In Figure 6.7, we compare baseline DBAL model against proposed models for Organ AMNIST dataset. In this case, all proposed models outperformed the baseline in all configurations due to the complex background of the Organ AMNIST dataset. The original dataset contains a lot of background noise for a classification task.

Table 6.1: TEST ACCURACY OF ALL MODEL CONFIGURATIONS

Dataset	Acq. Funcs.	Model			
		DBAL	VanillaVAE-DBAL	MMDVAE-DBAL	BiGAN-DBAL
MNIST	Entropy	0.94	0.96	0.96	0.89
	BALD	0.94	0.95	**0.97**	0.88
	Mean Std.	0.94	0.96	0.96	0.88
	Variation Ratios	0.94	**0.97**	**0.97**	0.89
	Uniform	0.92	0.94	0.94	0.87
Fashion MNIST	Entropy	0.77	0.77	0.77	0.75
	BALD	0.80	0.79	0.82	0.80
	Mean Std.	0.80	0.80	0.82	0.79
	Variation Ratios	0.81	0.80	**0.83**	0.80
	Uniform	0.81	0.79	0.82	0.78
Organ AMNIST	Entropy	0.52	0.69	0.70	0.68
	BALD	0.65	0.69	0.68	0.68
	Mean Std.	0.63	0.70	0.70	0.68
	Variation Ratios	0.50	0.71	**0.72**	0.70
	Uniform	0.58	0.68	0.70	0.68



(a) *Entropy* Acquisition Function

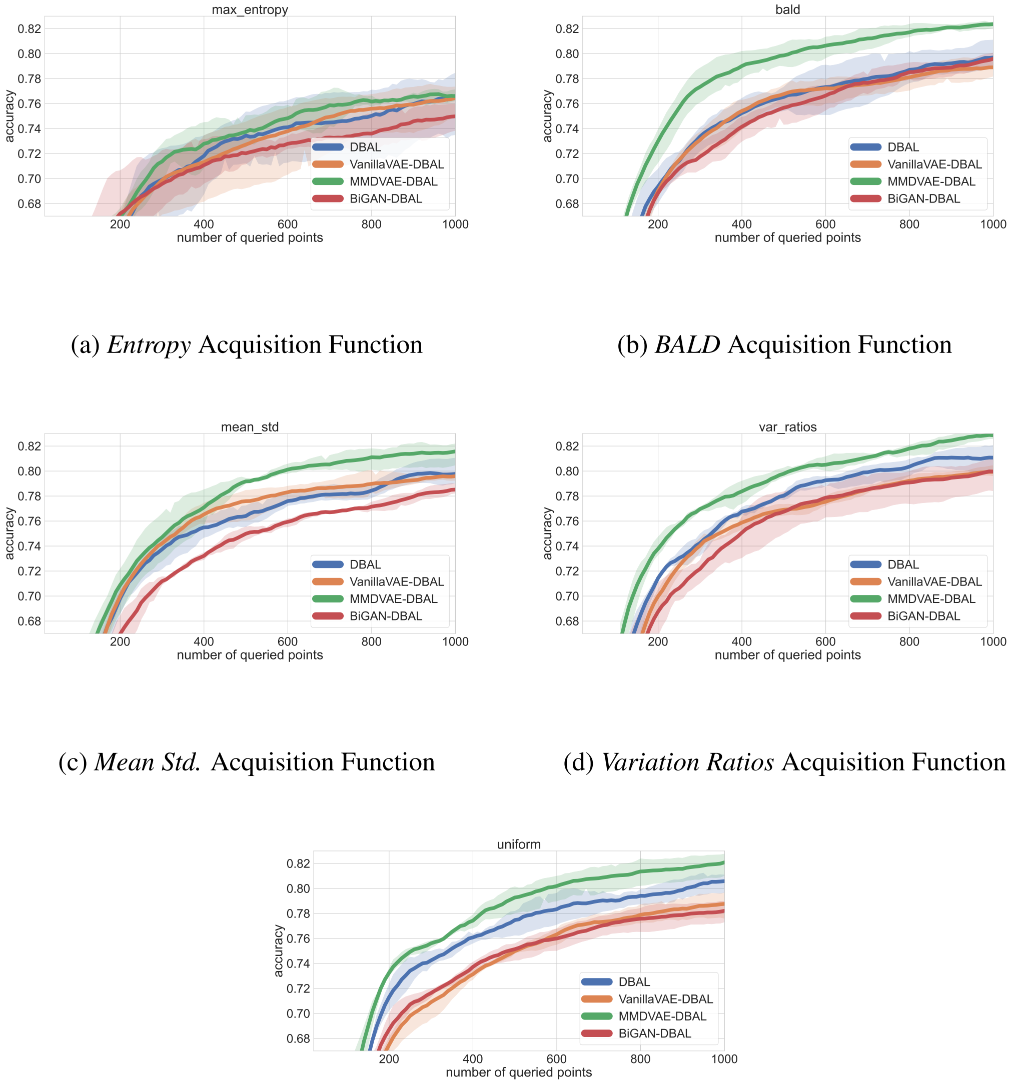
(b) *BALD* Acquisition Function

(c) *Mean Std.* Acquisition Function

(d) *Variation Ratios* Acquisition Function

(e) *Uniform* Acquisition Function

Figure 6.5: *MNIST* Test Accuracy Comparison Between DBAL and GEN-DBAL algorithms



(a) *Entropy* Acquisition Function

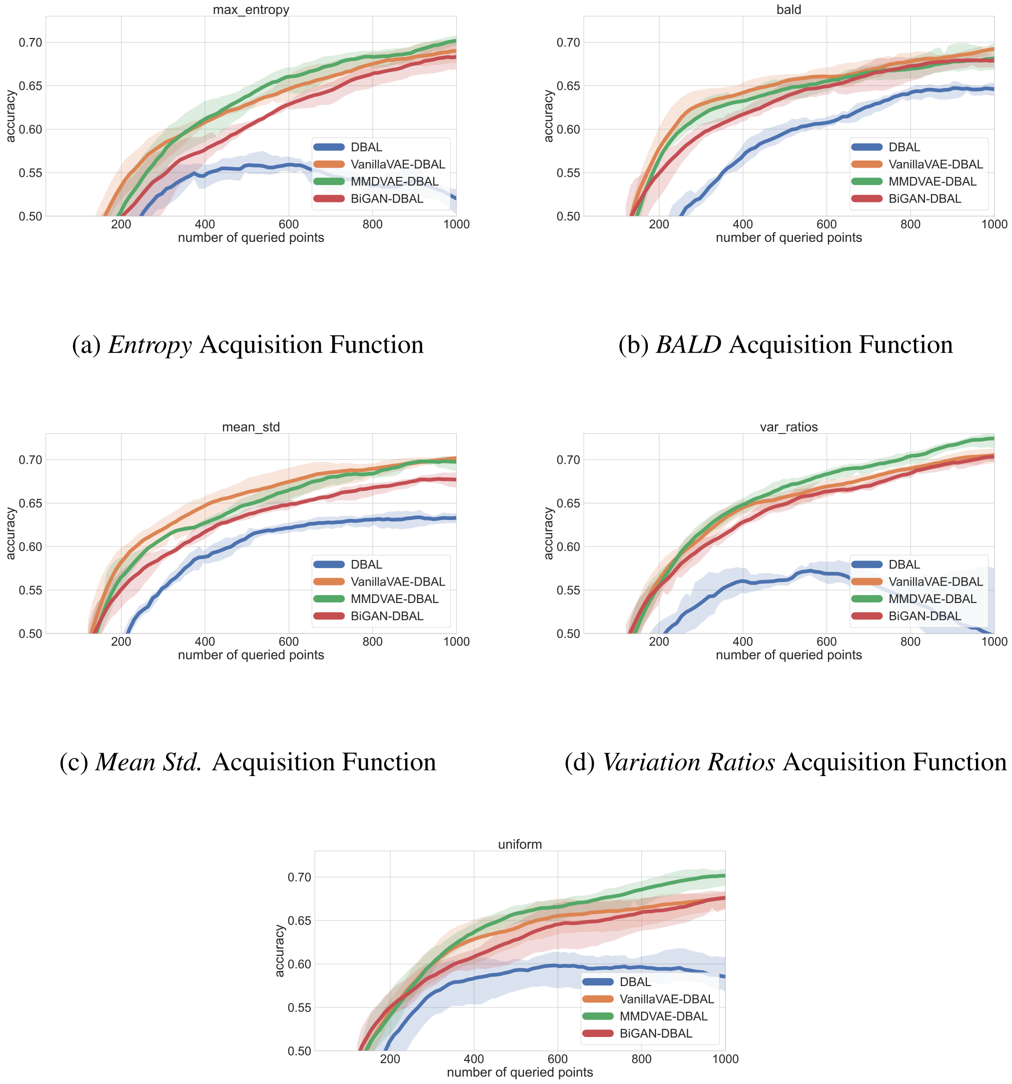
(b) *BALD* Acquisition Function

(c) *Mean Std.* Acquisition Function

(d) *Variation Ratios* Acquisition Function

(e) *Uniform* Acquisition Function

Figure 6.6: *Fashion MNIST* Test Accuracy Comparison Between DBAL and GEN-DBAL algorithms



(a) *Entropy* Acquisition Function

(b) *BALD* Acquisition Function

(c) *Mean Std.* Acquisition Function

(d) *Variation Ratios* Acquisition Function

(e) *Uniform* Acquisition Function

Figure 6.7: *Organ AMNIST* Test Accuracy Comparison Between DBAL and GEN-DBAL algorithms

6.4 Convergence Rate Analysis

In this section, we compare the convergence rate of the algorithms for a certain accuracy percentage. The convergence rate is calculated based on the number of points required to be labelled to reach a target accuracy percentage. It is preferable to have a low convergence rate.

MNIST

We performed this experiment using the MNIST dataset with Variation Ratios acquisition function as shown in Table 6.2. In order to achieve 80% accuracy, VanillaVAE-DBAL and MMDVAE-DBAL displayed the highest convergence rate, which requires 90 data points to be labelled. In order to achieve 90% accuracy, VanillaVAE-DBAL demonstrated the highest convergence rate, requiring 210 data points to be labelled.

Table 6.2: CONVERGENCE RATES OF ALGORITHMS FOR *MNIST* DATASET

Convergence Rates	Algorithms			
Target Accuracy	DBAL	VanilaVAE-DBAL	MMDVAE-DBAL	BiGAN-DBAL
80%	120	**90**	**90**	260
90%	450	**210**	230	-

Fashion MNIST

We performed this experiment using the Fashion MNIST dataset with Variation Ratios acquisition function as shown in Table 6.3.

Table 6.3: CONVERGENCE RATES OF ALGORITHMS FOR *Fashion MNIST* DATASET

Convergence Rates	Algorithms			
Target Accuracy	DBAL	VanilaVAE-DBAL	MMDVAE-DBAL	BiGAN-DBAL
70%	130	160	**90**	180
80%	640	950	**460**	950

In order to achieve 70% accuracy, MMDVAE-DBAL displayed the highest convergence rate, which requires 90 data points to be labelled. In order to achieve 80% accuracy, MMDVAE-DBAL demonstrated the highest convergence rate, which requires 460 data points to be labelled.

Organ AMNIST

We performed this experiment using Organ AMNIST dataset with the Variation Ratios acquisition function as shown in Table 6.4. In order to achieve 60% accuracy, both VanillaVAE-DBAL and MMDVAE-DBAL displayed the highest convergence rate, requiring 200 data points to be labelled. In order to achieve 70% accuracy, MMDVAE-DBAL demonstrated the highest convergence rate, requiring 630 data points to be labelled. The baseline DBAL model could not reach the target accuracy, therefore we could not provide a convergence rate for DBAL.

Table 6.4: CONVERGENCE RATES OF ALGORITHMS FOR *Organ AMNIST* DATASET

Convergence Rates	Algorithms			
Target Accuracy	DBAL	VanilaVAE-DBAL	MMDVAE-DBAL	BiGAN-DBAL
60%	-	**200**	**200**	230
70%	-	760	**630**	800

Discussion

Overall, the algorithms' convergence rates vary between datasets and depend heavily on the predefined target accuracy. A higher accuracy performance does not always imply that the algorithm converges faster to a target accuracy. In other words, the algorithm might lead to a high accuracy when the labelling budget is not strictly restricted. The convergence rate is a valid metric only if there is a training time and labelling budget restriction for an algorithm in order to reach a target accuracy.

6.5 Information Preservation with Deep Generative Models

We visualized the latent vectors generated by all three DGMs using the t-SNE representation [62] for the MNIST dataset. t-SNE is a projection method for data visualization. Using t-SNE, we map the high dimensional latent vector of size 64 into a smaller 2-D representation. In Figures 6.8 and 6.9, we can observe well-separated clusters of different class label for VanillaVAE and MMDVAE models respectively. However, BiGAN does not produce a proper separation for a few class labels, as seen in Figure 6.10. We attribute the low accuracy in the BiGAN-DBAL model to the latent space learnt by the BiGAN, which is not informative enough for the MNIST dataset. There are two main reasons why BiGAN could not outperform the baseline model. First, the Generator and Encoder models of BiGAN are trained without having access to outputs of each other; therefore, the network parameters of the Generator and Encoder are learned implicitly. Secondly, low-resolution images are used for the Generator model in the original paper [51]. However, we kept the resolution the same for both Generator and Encoder parts for consistency between all experiments.

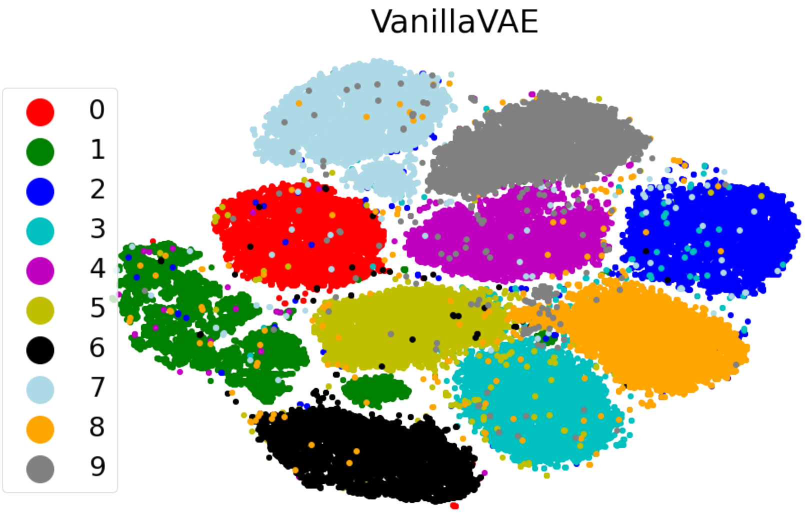


Figure 6.8: t-SNE Representations of Latent Space Learnt by VanillaVAE

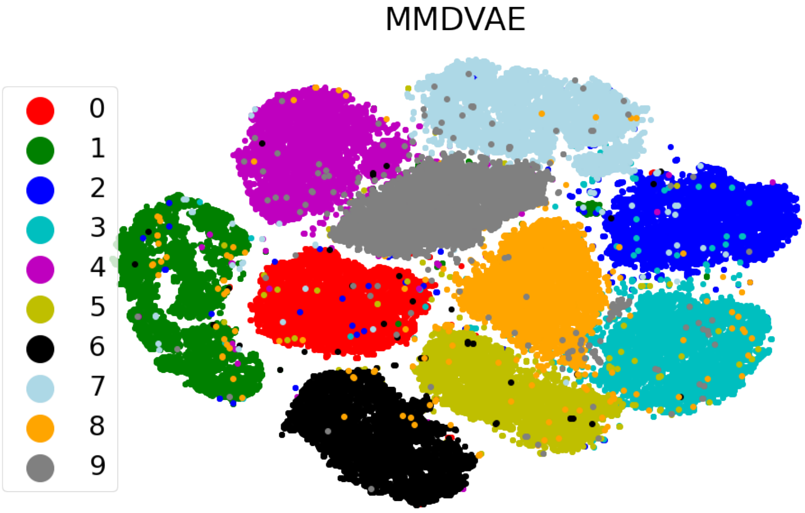


Figure 6.9: t-SNE Representations of Latent Space Learnt by MMDVAE

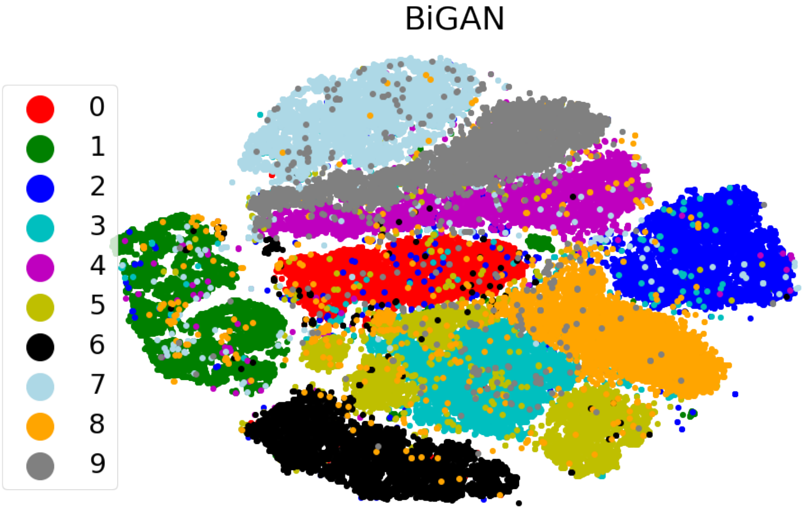


Figure 6.10: t-SNE Representations of Latent Space Learnt by BiGAN

6.6 Analysis of Feature Space

Let us refer to the Decoder models of VanillaVae and MMDVAE and the Generator model of BiGAN as G. $E(x)$ corresponds to the latent vector generated by Encoder models. Therefore, $G(E(x))$ corresponds to a generated image. In other words, it represents the output of a generative model. As shown in Figure 6.11, VanillaVAE and MMDVAE generate samples by preserving as much information as possible compared to the original image x. However, BiGAN lost much information during reconstruction.

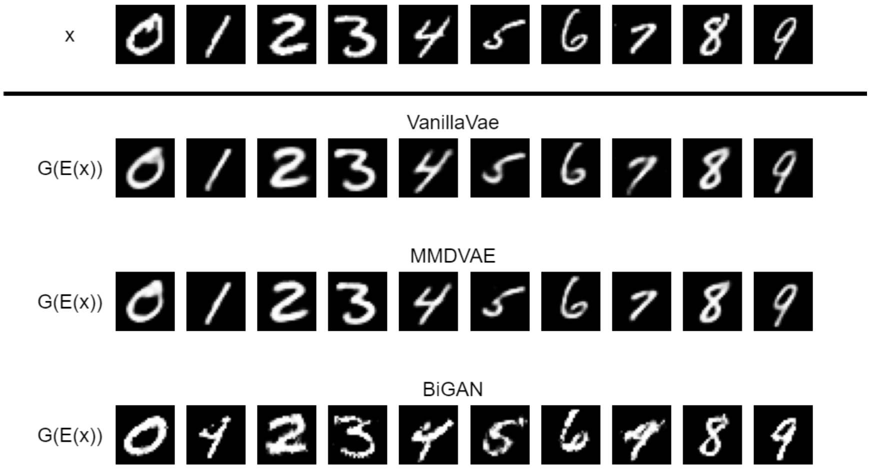


Figure 6.11: Reconstructed Images using Deep Generative Models

6.7 Does the size of the latent vector affect active learning accuracy?

In this section, we try to analyze the connection between the size of the latent vector and the information extracted from data points. In particular, we analyze the effectiveness of latent space dimension d on Active Learning accuracy. We trained a VanillaVAE on the MNIST train dataset in this experiment. The architecture and hyperparameters of VanillaVAE are the same as described in Chapter 4. In Figure 6.12, we can deduce that choosing a small size for d cannot preserve much information. When $d = 2$, the confidence interval oscillates in a large interval since the size of the latent vector is too small to extract any useful information. As d increases, the information preserved in the latent vector is improved. However, after some point, it converges into a place where choosing a larger d does not provide any significant improvement if not degrades the accuracy. Since having a larger vector means preserving unnecessary background information in the data, which harms the algorithm's accuracy. In the case of our experiment, we concluded that it is optimal to choose $d = 64$ for the MNIST dataset. However, this value might vary between datasets; for example, a dataset with larger images might require using a larger latent vector for feature extraction. Latent vectors are derived from VanillaVAE in this experiment. The Entropy acquisition function is utilized. Experiments are repeated three times, and confidence intervals are visualized in the figure.

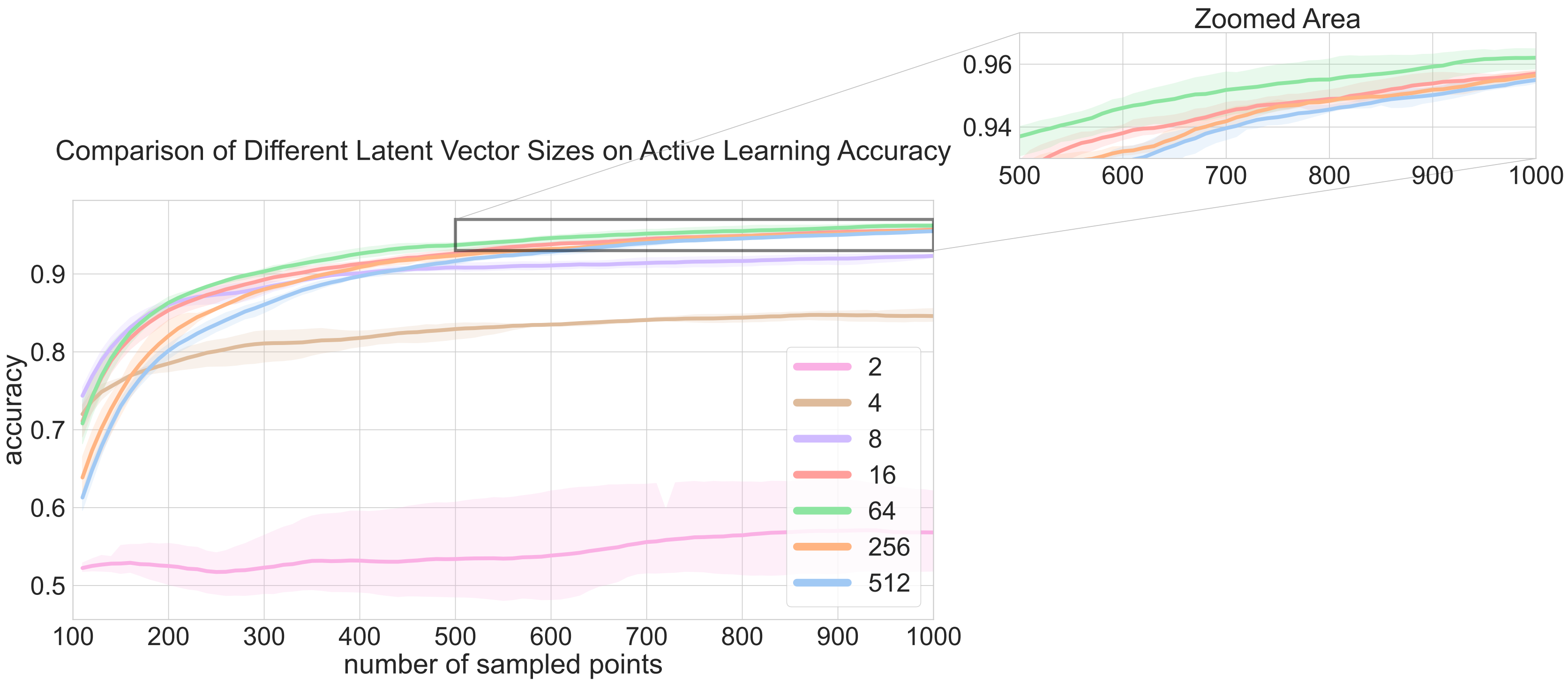


Figure 6.12: MNIST Accuracy Comparison of Different Latent Vector Sizes on $VanillaVAE - DBAL$ model.

CHAPTER 7

ALLEVIATING MODE-COLLAPSE PROBLEM IN DBAL

Uncertainty-based methods are essential in AL since they provide the most crucial information; hence, their use in AL has become prevalent. Although BNNs depict a strong foundation in learning high-dimensional data and estimating the uncertainty of data points, they are not robust against the Mode-Collapse problem. In particular, queried and labelled data points are class-imbalanced at the end, degrading accuracy. The main reason behind this deficiency is that Bayesian methods consider the uncertainty of data points by disregarding the diversity.

Different approaches in the literature ease the Mode-Collapse problem in BAL algorithms. BatchBALD [41] is an acquisition function which calculates the joint distribution of uncertainty of data points instead of calculating them independently. The joint uncertainty function enables the selection of more diverse data points, hence alleviating the Mode-Collapse problem. However, this methodology requires a consistent dropout on model parameters which contradicts the foundations of Bayesian approximation. In another study, [63], an ensemble of Bayesian Networks is utilized in the AL setting to solve the Mode-Collapse problem, whereas ensemble network training is not efficient.

This chapter proposes an algorithm incorporating a Diversity-Enhancing Acquisition Step into the GEN-DBAL algorithm, obtaining more diverse data points and improving the accuracy of AL. The proposed method is a hybrid method which combines the uncertainty metric with a diversity-based approach. To the best of our knowledge, this study proposes the first method, which alleviates the Mode-Collapse by combining the K-Means and Principal Component Analysis (PCA) with the DBAL algorithm.

7.1 Evidence of Mode-Collapse Problem in Deep Bayesian Active Learning

We obtain an unbalanced set of queried data points by the baseline DBAL algorithm even though we have a balanced unlabelled pool set U as is the case for MNIST and Fashion MNIST. However, Mode-Collapse is more evident in highly unbalanced training sets such as Organ AMNIST. This phenomena can be observed in Figure 7.1.

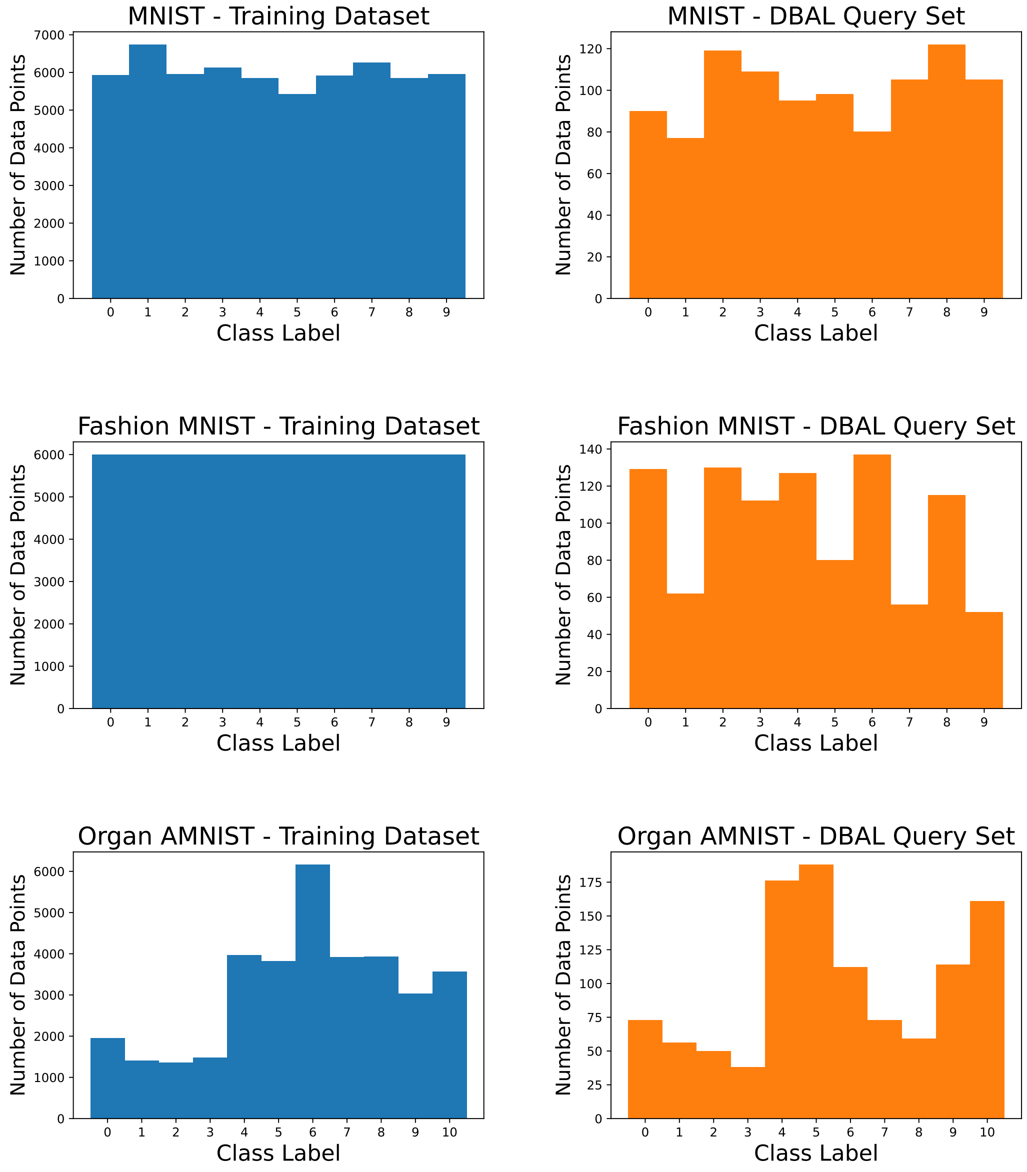


Figure 7.1: Histograms of the class labels. **Left:** Training Dataset Labels. **Right:** Queried Dataset Labels from DBAL Algorithm

7.2 Method

In this section, we propose a method on top of the GEN-DBAL algorithm, which is discussed in Chapter 4 in detail. It is a two-stage algorithm such that a DGM is trained using the whole dataset at first. In the second stage, we utilize the GEN-DBAL model with a proposed acquisition step as visualized in Figure 7.2. The details of the proposed acquisition step are explained in Algorithm 4.

Algorithm 4 Diversity Enhancing Acquisition Step

```
    Input $\alpha$ - acquisition function, $n_{batch}$ - batch size, $M$ - Classifier Model, D - Dataset

  $D_s \leftarrow argmax_{d_i \in D | 1 \leq i \leq \beta * n_{batch}}(\alpha(D, M))$
  $P_s \leftarrow PCA(D_s, d)$
  $C \leftarrow KMeans(P_s, k)$                                  ▷ $C$: Cluster Centers
  $S \leftarrow \emptyset$
  while $n_{batch}$ data points are queried do
      for $c \in C$ do
          $S \leftarrow S \cup argmin_{d_i \in D_s} distance(p_i, c)$
          return $S$
```

The proposed acquisition step is developed on top of a method in the literature; Diverse-Mini Batch Active Learning [64]. In this method, the most unconfident $\beta * n_{batch}$ data points are selected and clustered using K-Means [65] algorithm with k being the number of clusters. n_{batch} data points are sampled, which are closest to the cluster centers based on the Euclidean distance metric. The queried data points are then labelled and joined into the AL loop. As an improvement to this technique, we incorporate PCA in order to reduce the dimensions of the dataset before clustering since K-Means performs better in low-dimensional data [66]. PCA projects feature space Z into a lower-dimensional P. Although P is used in clustering and querying, M_{BMLP} is trained on Z since the accuracy of the AL algorithm depends heavily on the informativeness of data points. The purpose of the Diversity Enhancing Acquisition Step is to query the most diverse data points among the most informative $\beta * n_{batch}$ points.

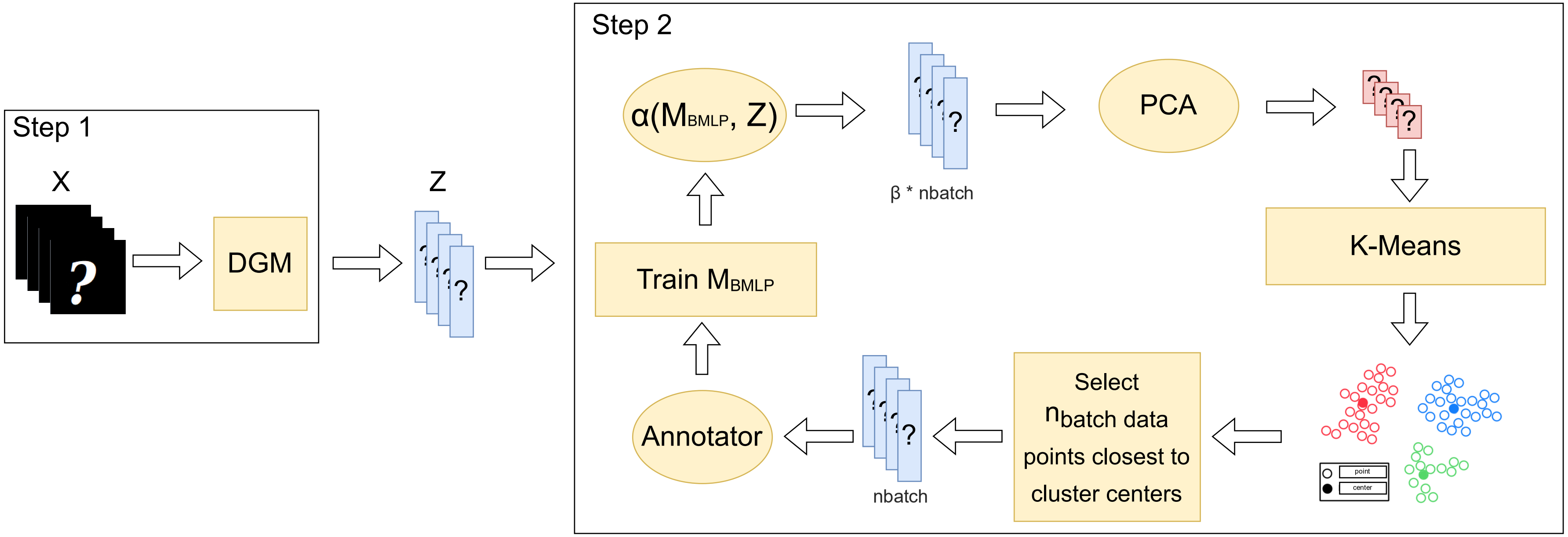


Figure 7.2: The Proposed Active Learning Loop

7.3 Implementation Details

In the proposed method, we utilize regular Vanilla VAE for feature learning and M_{BMLP} as the classifier model with the architectures and parameters as explained in Chapter 5. Active Learning budget B is selected as 1000, and the number of initially labelled data points n_l are selected as $2 * c$ where c is the number of classes for each dataset. In the experiments, we utilize 3 different datasets; MNIST [59], FashionMNIST [60] and OrganAMNIST [61]. In MNIST and FashionMnist, the number of classes c is 10, whereas, for the OrganAMNIST, it is 11. We utilize the BALD acquisition function in the experiments. Other parameters are kept the same as the GEN-DBAL method. In addition, the β coefficient is set to 10, and the number of principal components in PCA d is set to 2. k value refers to the number of clusters in the K-Means Clustering algorithm, which differs in value for each dataset.

7.4 Accuracy Results

We compare the accuracy of the proposed acquisition step with the baseline acquisition step. For both settings, we employ the VanillaVAE-DBAL algorithm with the BALD acquisition function for consistency. When the number of clusters k is selected as equal to the number of classes c, there is not any significant improvement in the accuracy metric. However, we observe that different k values result in different accuracy performances for each dataset. In this experimental setting, n_{batch} is always equal to k value.

MNIST

As visualized in Figure 7.3, the use of $k = 20$ has resulted in the best performance for the proposed acquisition step for the MNIST dataset. The proposed model outperforms the baseline model for k values of [10, 20, 50, 100]. However, when $k = 250$, then the accuracy is worse than the baseline method. The main reason is due to the over-clustering of data points, which results in the querying and labelling of 250 data points in each step which is not ideal for an AL setting.

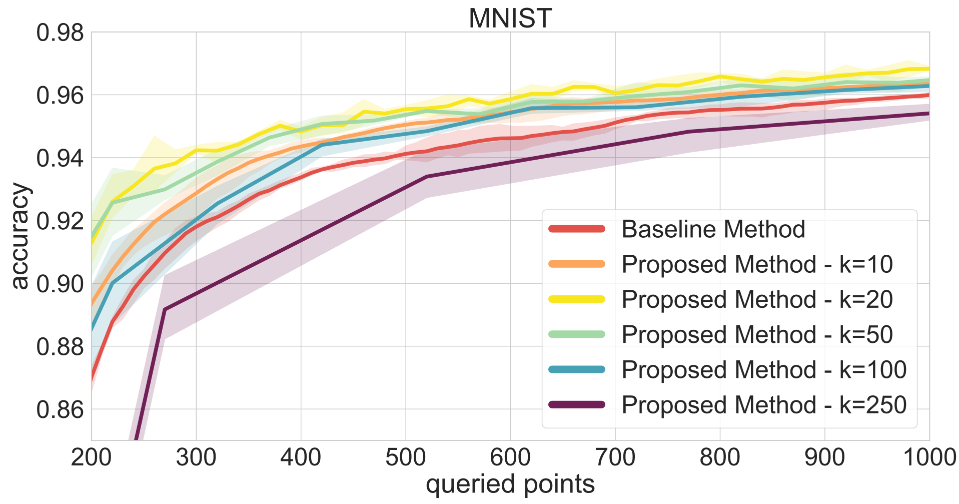


Figure 7.3: The Accuracy Metric for MNIST

Fashion MNIST

As shown in Figure 7.4, the proposed method outperforms the baseline accuracy for all k values for the Fashion MNIST dataset. The best performance is obtained when $k = 20$.

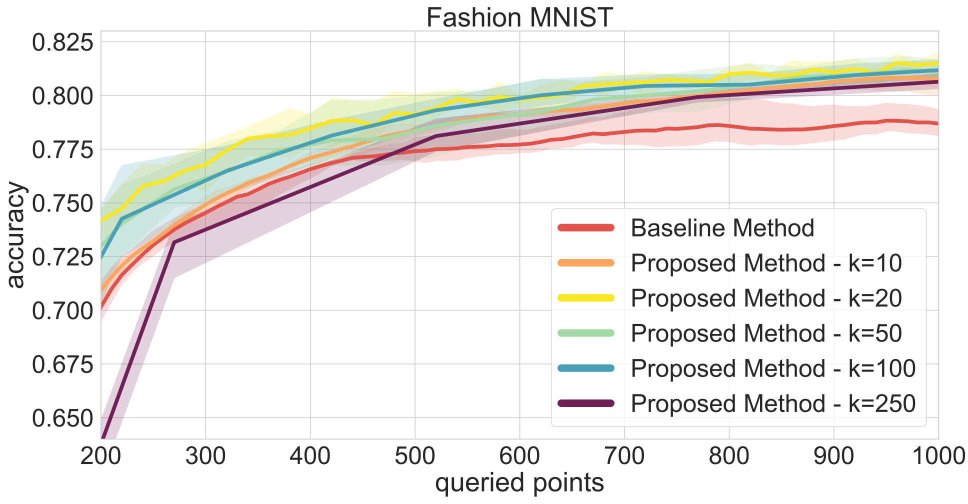


Figure 7.4: The Accuracy Metric for Fashion MNIST

Organ AMNIST

As shown in Figure 7.5, the proposed method outperforms the baseline accuracy for all k values for the Organ AMNIST dataset. The best performance is obtained when $k = 55$.

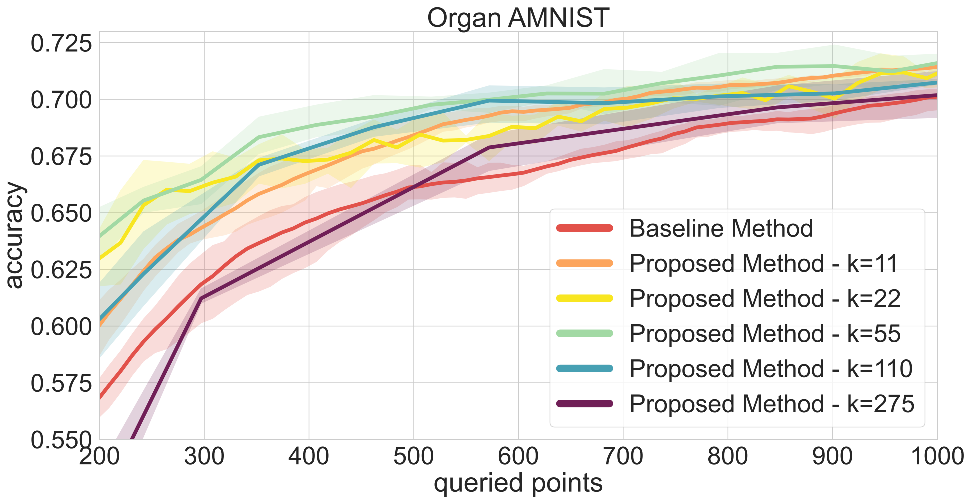


Figure 7.5: The Accuracy Metric for Organ AMNIST

7.5 Effectiveness of PCA on the Accuracy Metric

In this section, we provide experiments on the proposed Diversity-Enhancing Acquisition Step with and without the PCA algorithm and compare the results against the baseline acquisition. Since the PCA algorithm is our main improvement to the Diverse-Mini Batch AL algorithm, we demonstrate that the accuracy is even worse than the regular uncertainty-based approach for the proposed acquisition step without PCA. In Figure 7.6, it can be observed that the use of PCA significantly improves the accuracy compared to the proposed algorithm without PCA.

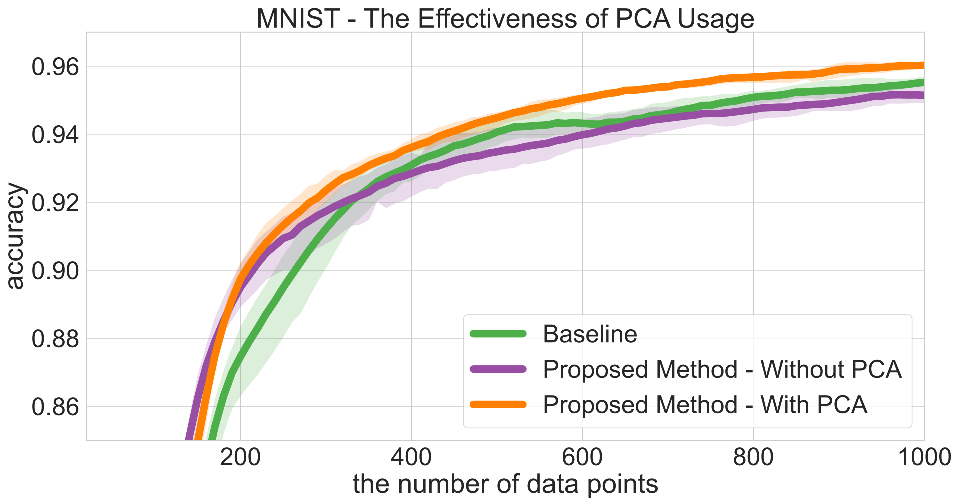


Figure 7.6: The Effectiveness of PCA Algorithm on Accuracy

7.6 Measuring Dataset Diversity

In this section, we measure the effect of the Mode-Collapse phenomenon for each dataset with different k values. In order to measure the diversity of data points in a selected algorithm, we calculate the Entropy of class labels. If the Entropy value of class labels is large, then we obtain a more diverse and more balanced dataset, hence easing the Mode-Collapse problem. We visualize the results in Figure 7.7. For each dataset, the proposed methods result in a more diverse selection of data points for k values of [c, 2c, 5c, 10c, 25c], where c is the number of class labels. There is not a significant increase in the dataset diversity when $k = 10$. Therefore, over-clustering is required to alleviate the Mode-Collapse problem. However, there does not exist a linear relationship between the k value and the diversity of the queried dataset.

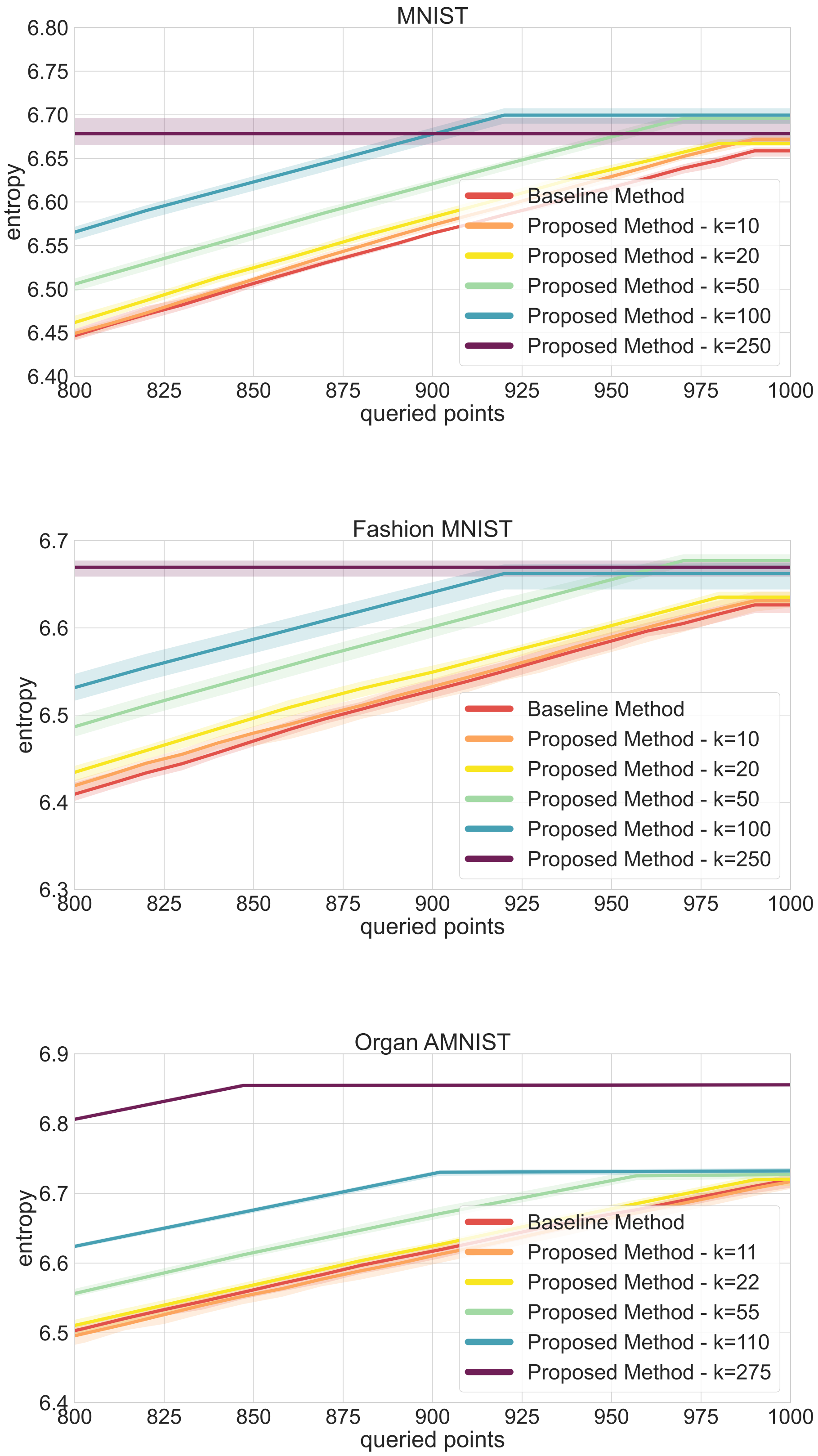


Figure 7.7: Entropy of Class Labels. **Top:** MNIST, **Middle:** Fashion MNIST, **Bottom:** Organ AMNIST

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this thesis, we introduce the details of an algorithm called GEN-DBAL, which improves the accuracy of the Deep Bayesian Active Learning algorithm. We demonstrate that feature learning from a Deep Generative Model outperforms the performance of the active learning model if the latent vector is informative enough. For this purpose, we performed experiments on three different generative models, namely Vanilla Variational Autoencoders (Vanilla VAEs), Maximum Mean Discrepancy Variational Autoencoders (MMDVAEs) and Bidirectional Generative Adversarial Networks (BiGANs). Our experiments lead to a result that if the latent space representations of a generative model are informative enough, it can improve the performance of DBAL accuracy.

Our experiments on generative models result in that MMDVAE can learn meaningful representations of feature space and consistently improve the accuracy of the active learning problem for each dataset since it utilizes the InfoVAE loss function, which is an information-centric loss. InfoVAE loss function is an improvement to the ELBO loss function, which is used in regular Vanilla VAE. InfoVAE does not fail in the task of learning latent vectors due to better regularization.

The proposed algorithm outperforms the baseline model for each DGM used when a dataset with complex background is utilized, such as Organ AMNIST. In particular, using the baseline DBAL algorithm should be avoided with datasets with a complex background. Even the BiGAN-DBAL algorithm outperforms the baseline in this case which is not effective for more simple datasets.

The VanillaVAE-DBAL algorithm might fall short in terms of feature learning due to conflicting objectives in the ELBO loss function; therefore, we might observe a decrease in the accuracy compared to the baseline. If the reconstruction term is more dominant over the regularization term, the Vanilla VAE model fails in the task of latent space learning, which is the case for the Fashion MNIST dataset in our experiments.

Furthermore, we establish a relationship between the informativeness of a latent space and the accuracy of the DBAL algorithm. We measure the amount of useful information captured by a DGM in two ways: analyzing generated images and analyzing the clusters by projecting latent space into lower dimensions. Having accurately reconstructed images shows that we would obtain better accuracy in DBAL. Also, having distinctive clusters from a latent space demonstrates that we obtain informative feature space, hence improved accuracy.

Furthermore, we performed experiments to reveal the effect of latent space size on active learning accuracy. It turns out that using very small latent sizes cannot capture enough information to represent data sufficiently. Using a very large value also degrades the performance since we also preserve unnecessary information, such as background noise, when we use a large vector. In other words, using an enormous latent vector size does not add a noticeable improvement to active learning accuracy; if not degrades the performance. The optimal size of a latent vector might vary between datasets.

In addition, we propose a diversity-enhancing acquisition step in order to alleviate the Mode-Collapse problem in GEN-DBAL. We incorporate the PCA algorithm into a diversity-based methodology, therefore obtaining a more balanced queried set. The proposed scheme increases the diversity of the queried data points, hence improving the accuracy.

8.2 Future Work

We proposed an algorithm that is composed of two-stage training. In the first stage, a DGM is trained using the whole dataset; then, the DBAL algorithm is trained on latent vector representation learnt by the DGM in the second stage. As a future improvement, a scheme can be developed to enable joint training of the proposed model, GEN-DBAL. The joint training of the model would provide an efficient training time due to speed up. Furthermore, in the diversity-enhancing acquisition step, different clustering methods and different distance metrics can be utilized in future work. Also, a clustering-based DGM can be fused into the model to speed up the overall algorithm since we would not need to perform extra clustering.

REFERENCES

[1] Y. Gal and Z. Ghahramani, “Bayesian convolutional neural networks with bernoulli approximate variational inference,” 2016.

[2] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” 2016.

[3] P. E. Çöl and Ş. Ertekin, “Feature dimensionality reduction with variational autoencoders in deep bayesian active learning,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2021.

[4] P. E. Duymuş and Ş. Ertekin, “Alleviating mode collapse problem in deep bayesian active learning,” in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2022.

[5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *J. Artif. Int. Res.*, vol. 4, p. 129–145, Mar. 1996.

[6] B. Settles, “Active learning literature survey,” 2009.

[7] D. Angluin, “Queries and concept learning,” *Machine Learning*, vol. 2, pp. 319–342, 1988.

[8] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Mach. Learn.*, vol. 15, p. 201–221, may 1994.

[9] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *SIGIR’94*, pp. 3–12, Springer, 1994.

[10] B. Settles, “Active learning literature survey,” 2009.

[11] M.-R. Bouguelia, Y. Belaïd, and A. Belaïd, “A stream-based semi-supervised active learning approach for document classification,” in *2013 12th International Conference on Document Analysis and Recognition*, pp. 611–615, IEEE, 2013.

[12] A. Narr, R. Triebel, and D. Cremers, “Stream-based active learning for efficient and adaptive classification of 3d objects,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 227–233, IEEE, 2016.

[13] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, “Stream-based active learning for sentiment analysis in the financial domain,” *Information sciences*, vol. 285, pp. 181–203, 2014.

[14] A. K. McCallumzy and K. Nigamy, “Employing em and pool-based active learning for text classification,” in *Proc. International Conference on Machine Learning (ICML)*, pp. 359–367, Citeseer, 1998.

[15] M. Sugiyama and S. Nakajima, “Pool-based active learning in approximate linear regression,” *Machine Learning*, vol. 75, no. 3, pp. 249–274, 2009.

[16] D. Wu, “Pool-based sequential active learning for regression,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1348–1359, 2018.

[17] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[18] J. Kremer, K. Steenstrup Pedersen, and C. Igel, “Active learning with support vector machines,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 4, pp. 313–326, 2014.

[19] Z. Wang and A. Brenning, “Active-learning approaches for landslide mapping using support vector machines,” *Remote Sensing*, vol. 13, no. 13, p. 2588, 2021.

[20] K. Brinker, “Incorporating diversity in active learning with support vector machines,” in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML’03, p. 59–66, AAAI Press, 2003.

[21] B. Settles, M. Craven, and S. Ray, “Multiple-instance active learning,” in *Advances in Neural Information Processing Systems* (J. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), vol. 20, Curran Associates, Inc., 2008.

[22] D. Agarwal, P. Srivastava, S. Martin-del Campo, B. Natarajan, and B. Srinivasan, “Addressing practical challenges in active learning via a hybrid query strategy,” *arXiv preprint arXiv:2110.03785*, 2021.

[23] E. Lughofer, “Hybrid active learning for reducing the annotation effort of operators in classification systems,” *Pattern Recognition*, vol. 45, no. 2, pp. 884–896, 2012.

[24] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” *ArXiv*, vol. abs/1703.02910, 2017.

[25] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, “Deep batch active learning by diverse, uncertain gradient lower bounds,” *ArXiv*, vol. abs/1906.03671, 2020.

[26] R. Pinsler, J. Gordon, E. T. Nalisnick, and J. M. Hernández-Lobato, “Bayesian batch active learning as sparse subset approximation,” in *NeurIPS*, 2019.

[27] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” 2018.

[28] W. Cai, Y. Zhang, and J. Zhou, “Maximizing expected model change for active learning in regression,” in *2013 IEEE 13th international conference on data mining*, pp. 51–60, IEEE, 2013.

[29] W. Cai, M. Zhang, and Y. Zhang, “Batch mode active learning for regression with expected model change,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 7, pp. 1668–1681, 2016.

[30] S. H. Park and S. B. Kim, “Robust expected model change for active learning in regression,” *Applied Intelligence*, vol. 50, no. 2, pp. 296–313, 2020.

[31] S. Tong and D. Koller, “Active learning for structure in bayesian networks,” in *International joint conference on artificial intelligence*, vol. 17, pp. 863–869, Citeseer, 2001.

[32] S. Tong and D. Koller, “Active learning for parameter estimation in bayesian networks,” in *NIPS*, vol. 13, pp. 647–653, 2000.

[33] S. Bodenstedt, D. Rivoir, A. Jenke, M. Wagner, M. Breucha, B. Müller-Stich, S. T. Mees, J. Weitz, and S. Speidel, “Active learning using deep bayesian networks for surgical workflow analysis,” *International journal of computer assisted radiology and surgery*, vol. 14, no. 6, pp. 1079–1087, 2019.

[34] A. Jonsson and A. Barto, “Active learning of dynamic bayesian networks in markov decision processes,” in *International Symposium on Abstraction, Reformulation, and Approximation*, pp. 273–284, Springer, 2007.

[35] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, “Bayesian active learning for classification and preference learning,” *arXiv preprint arXiv:1112.5745*, 2011.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[37] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, “The power of ensembles for active learning in image classification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9368–9377, 2018.

[38] V. Rakesh and S. Jain, “Efficacy of bayesian neural networks in active learning,” 2021.

[39] K. Chitta, J. M. Alvarez, and A. Lesnikowski, “Large-scale visual active learning with deep probabilistic ensembles,” 2019.

[40] R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato, “Bayesian batch active learning as sparse subset approximation,” 2021.

[41] A. Kirsch, J. van Amersfoort, and Y. Gal, “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning,” 2019.

[42] J.-J. Zhu and J. Bento, “Generative adversarial active learning,” 2017.

[43] T. Tran, T.-T. Do, I. Reid, and G. Carneiro, “Bayesian generative active deep learning,” 2019.

[44] M. Huijser and J. C. van Gemert, “Active decision boundary annotation with deep generative models,” in *Proceedings of the IEEE international conference on computer vision*, pp. 5286–5295, 2017.

[45] S. Sinha, S. Ebrahimi, and T. Darrell, “Variational adversarial active learning,” 2019.

[46] F. Pourkamali-Anaraki and M. B. Wakin, “The effectiveness of variational autoencoders for active learning,” 2019.

[47] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2014.

[48] J. M. Joyce, *Kullback-Leibler Divergence*, pp. 720–722. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[49] S. Zhao, J. Song, and S. Ermon, “Infovae: Balancing learning and inference in variational autoencoders,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5885–5892, Jul. 2019.

[50] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, “Variational lossy autoencoder,” 2017.

[51] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” 2017.

[52] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.

[53] Y. Guo and D. Schuurmans, “Discriminative batch mode active learning,” in *Advances in Neural Information Processing Systems* (J. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), vol. 20, Curran Associates, Inc., 2008.

[54] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[55] L. G. Freeman, “Elementary applied statistics for students in behavioral science,” 1965.

[56] A. Kendall, V. Badrinarayanan, and R. Cipolla, “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” 2015.

[57] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 680–688, 2016.

[58] Y. Gal, "Uncertainty in deep learning," 2016.

[59] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[60] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[61] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification," 2021.

[62] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[63] R. Pop and P. Fulop, "Deep ensemble bayesian active learning : Addressing the mode collapse issue in monte carlo dropout via ensembles," 2018.

[64] F. Zhdanov, "Diverse mini-batch active learning," 2019.

[65] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[66] N. Dhavamany and P. S, "A new method for dimensionality reduction using k-means clustering algorithm for high dimensional data set," *International Journal of Computer Applications*, vol. 13, 01 2010.