BGDB: Bernoulli-Gaussian Decision Block with Improved Denoising Diffusion Probabilistic Models

Chengkun Sun[1], Jinqian Pan[1], Russell Stevens Terry[2], Jiang Bian[1], Jie Xu[1]

[1]Department of Health Outcomes and Biomedical Informatics, University of Florida, Gainesville, FL 32611, USA
[2]Department of Urology, University of Florida, Gainesville, FL 32611, USA
{sun.chengkun,jinqianpan,bianjiang,xujie}@ufl.edu, russell.terry@urology.ufl.edu

Abstract

Generative models can enhance discriminative classifiers by constructing complex feature spaces, thereby improving performance on intricate datasets. Conventional methods typically augment datasets with more detailed feature representations or increase dimensionality to make nonlinear data linearly separable. Utilizing a generative model solely for feature space processing falls short of unlocking its full potential within a classifier and typically lacks a solid theoretical foundation. We base our approach on a novel hypothesis: the probability information (logit) derived from a single model training can be used to generate the equivalent of multiple training sessions. Leveraging the Central Limit Theorem (CLT), this synthesized probability information is anticipated to converge toward the true probability more accurately. To achieve this goal, we propose the **Bernoulli-Gaussian Decision Block (BGDB)**, a novel module inspired by the CLT and the concept that the mean of multiple Bernoulli trials approximates the probability of success in a single trial. Specifically, we utilize Improved Denoising Diffusion Probabilistic Models (IDDPM) to model the probability of Bernoulli Trials. Our approach shifts the focus from reconstructing features to reconstructing logits, transforming the logit from a single iteration into logits analogous to those from multiple experiments. We provide the theoretical foundations of our approach through mathematical analysis and validate its effectiveness through experimental evaluation using various datasets for multiple imaging tasks, including both classification and segmentation.

Code — https://github.com/sunck1/BGDB

Introduction

Classifiers are fundamental tools in machine learning, responsible for discerning intricate relationships between predictors and responses to allocate new observations into predetermined classes (Rubinstein, Hastie et al. 1997). Among them, discriminative classifiers have gained prominence for their efficiency. Discriminative classifiers directly learn the conditional probability $P(y|x)$, selecting the label y with the highest likelihood given an input x (Ng and Jordan 2001). This direct approach bypasses the need to model the joint probability distribution $P(x, y)$, as generative classifiers do, leading to faster decision-making (Raina et al. 2003). Consequently, discriminative classifiers, particularly within convolutional neural networks (CNNs), have become the preferred choice for tasks such as image processing (Krizhevsky, Sutskever, and Hinton 2012; Miao et al. 2019, 2018).

Despite their widespread use and efficiency, discriminative classifiers face challenges in extracting features and defining metric relations between examples, especially with complex data types such as medical images (Jaakkola and Haussler 1998). This limitation stems from their focus on learning the decision boundary rather than understanding the underlying data distribution. In contrast, generative models offer a promising solution by constructing more intricate feature spaces and providing a sophisticated framework for understanding the data generation process (Perina et al. 2012). By creating structured hierarchies of latent variables linked through conditional distributions, generative models can establish nuanced correspondences between model components and observed features, enabling them to handle missing, unlabeled, and variable-length data effectively (Perina et al. 2012). Techniques such as Fisher's method exemplify this approach, where original data is mapped into a low-dimensional feature space and then projected into a higher-dimensional space by kernel techniques for linear classification (Jaakkola and Haussler 1998). Another strategy involves augmenting data with generative models to improve feature representations, as seen in methods like Dataset Diffusion, which enhances the accuracy of segmentation and classification tasks (Nguyen et al. 2024). However, the direct integration of generative models into feature construction in discriminative classifiers often lacks a robust theoretical foundation. In such cases, the generative model typically generates an unknown latent space from another unknown latent space, making the generation process inherently difficult to interpret.

In this paper, we propose a new hypothesis that the probability distribution obtained by a single training process can be used to generate the probability distribution for multiple training processes. Ideally, this generated distribution would represent the true classification probability distribution. Specifically, compared to other generative models such as GANs (Goodfellow et al. 2014), which produce data through the adversarial process between the generator and the discriminator, diffusion models (Jarzynski 1997) have

the advantage of generating one distribution from another and provide a mathematical foundation for this process. On the other hand, leveraging the distributions from a single training process, we can generate the probability distributions for multiple training iterations. According to the Central Limit Theorem, these generated distributions will more precisely approximate the true classification probabilities. This methodology thus enhances the model's classification performance through supervised learning. Building on this idea, we incorporated the diffusion model into the discriminative classifier, developing a Bernoulli-Gaussian Decision Block (BGDB) designed to enhance the deep learning model. Our contributions can be summarized as follows:

- We introduce the Bernoulli-Gaussian Decision Block, which enhances the stability and performance of discriminative classifiers by leveraging the mean of logits from multiple experiments to supervise a single learning process.
- We employ IDDPM to construct and refine the probability distributions of Bernoulli Trials, improving inference accuracy without adding computational complexity during inference.
- We provide a theoretical analysis and validate the effectiveness of our approach through extensive experiments on multiple datasets, including Cityscapes, ISIC, and Pascal VOC, demonstrating notable improvements in classification and segmentation tasks.

Related Work

Central Limit Theorem in Neural Networks

Learning conditional and marginal probabilities from a dataset is fundamental to constructing machine learning methods, such as belief networks (Davidson and Aminian 2004). Leveraging the Central Limit Theorem (CLT) could enhance this process by providing a robust statistical foundation (Davidson and Aminian 2004). According to the CLT, the sum of a large number of random variables approximates a Gaussian distribution. This principle also applies to neural networks, where the pre-activations of each layer tend to be Gaussian (Huang et al. 2021). As the network width increases towards infinity, the output distribution of each neuron converges to a Gaussian distribution (Zhang, Wang, and Fan 2022). Thus, optimization in neural networks can be framed as optimizing a Gaussian process (Lee et al. 2017).

Many neural network optimization techniques are developed based on the CLT. For instance, from a width-depth symmetry perspective, shortcut networks demonstrate that increasing the depth of a neural network also results in a Gaussian process manifestation (Zhang, Wang, and Fan 2022). In the Empirical Risk Minimization (ERM) framework, the long-term deviation, scaled by the CLT, is governed by a Monte Carlo resampling error, providing width-asymptotic guarantees independent of data dimension (Chen et al. 2020). Self-Normalizing Neural Networks utilize the CLT to approximate network inputs with a Gaussian distribution, enabling robust learning and introducing novel regularization schemes (Klambauer et al. 2017). Despite these advancements, existing methods primarily rely on the CLT's mathematical properties for parameter estimation rather than directly modeling the CLT process within neural networks. This approach limits the potential of the CLT for optimizing neural networks to some extent.

Logit-Based Optimization

The logit function, introduced by Joseph Berkson in 1944, is derived from the term "logistic unit" and describes the logarithm of odds (Berkson 1951). It maps the probability range $(0, 1)$ to the entire real number line $(-\infty, +\infty)$, allowing the application of linear regression techniques to probabilities (Cramer 2003). This mapping facilitates the use of regression methods in domains where outputs are naturally bounded probabilities rather than unbounded real numbers. In modern machine learning, the flexibility to let data drive model structures has led to more adaptive and predictive capabilities (Zhao et al. 2020). This flexibility contrasts with traditional logit models, which often rely on specific data structures and inherent behavioral assumptions.

Various methods have been developed to optimize neural networks by focusing on the logit function. Wu et al. (Wu and Klabjan 2021) introduced a reliable uncertainty measure based on logit outputs, aiding classification models in identifying instances prone to errors. This uncertainty measure can trigger expert intervention during high uncertainty classifications (Wu and Klabjan 2021). Neural networks often exhibit overconfidence, producing high confidence scores for both in- and out-of-distribution inputs. Wei et al. (Wei et al. 2022) addressed this issue with Logit Normalization (LogitNorm), modifying the cross-entropy loss to enforce a constant vector norm on the logits during training. In medical image analysis, Hu et al. (Hu et al. 2021) proposed logit space data augmentation, adaptively perturbing logit vectors to enhance classifier generalizability and mitigate overfitting from limited training data. These methods demonstrate that optimizing based on logit can significantly enhance neural network performance on finite datasets.

Diffusion Probabilistic Models

Diffusion probabilistic models (DPMs) (or diffusion models [DMs]), inspired by non-equilibrium statistical physics (Jarzynski 1997), have recently gained traction in computer vision due to their remarkable generative capabilities. DMs generate highly detailed and diverse examples by iteratively reconfiguring data distribution through a diffusion process (Yang et al. 2023). Incorporating small amounts of Gaussian noise, DMs use conditional Gaussians for straightforward parameterization of neural networks. Leveraging variational inference via a parameterized Markov chain (Gagniuc 2017), DMs generate samples closely following the original data distribution within finite iterations.

Notable examples include latent diffusion models (LDMs) (Croitoru et al. 2023; Yang et al. 2023), which have set new standards in generative modeling. Stable Diffusion, a variant of LDMs, generates high-quality images based on text prompts, showcasing minimal artifacts and strong alignment with the prompts (Yang et al. 2023). DMs have

been extensively applied in image generation (Nichol and Dhariwal 2021), super-resolution (Rombach et al. 2022), and image-to-image translation (Choi et al. 2021). Additionally, the latent representations learned by DMs have proven effective in discriminative tasks like image segmentation (Baranchuk et al. 2021), and classification (Zimmermann et al. 2021). This versatility underscores the potential of diffusion models in a broad range of applications, connecting them to the field of representation learning, which includes designing novel neural architectures and developing advanced learning strategies (Croitoru et al. 2023; Yang et al. 2023).

Methods

In this paper, we propose the Bernoulli-Gaussian decision block, a novel module inspired by the CLT, which utilizes IDDPMs (Nichol and Dhariwal 2021) to model the probability of Bernoulli trials. We will first review the formulation of IDDPMs, followed by a detailed description of the proposed Bernoulli-Gaussian Decision Block built upon the IDDPMs.

Improved Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPMs) (Ho, Jain, and Abbeel 2020) have demonstrated superior sample generation quality, often surpassing other generative models like GANs (Goodfellow et al. 2014) and VQ-VAE (Van Den Oord, Vinyals et al. 2017). Improved DDPMs (IDDPMs) (Nichol and Dhariwal 2021) build on DDPMs by incorporating learned variances, allowing sampling in fewer steps with minimal quality loss. In DDPMs, given data distribution $x_0 \sim q(x_0)$, a forward noising process q generates latent variables x_1 through x_T by adding Gaussian noise at each time t with variance $\beta_t \in (0, 1)$, as follows (Nichol and Dhariwal 2021):

$$\begin{aligned} &q(x_1, ..., x_T|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ &\text{where } \; q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}). \end{aligned} \tag{1}$$

With a sufficiently large T and a carefully designed schedule for β_t, the latent variable x_T approximates an almost isotropic Gaussian distribution (Nichol and Dhariwal 2021). Consequently, if the exact reverse distribution $q(x_{t-1}|x_t)$ were known, we could sample $x_T \sim \mathcal{N}(0, \boldsymbol{I})$ and reverse the process to obtain a sample from $q(x_0)$. However, since $q(x_{t-1}|x_t)$ relies on the entire data distribution, it is approximated using a neural network (Nichol and Dhariwal 2021):

$$\begin{aligned} &p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \\ &\text{where } \; \Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}. \end{aligned} \tag{2}$$

Through Maximum Likelihood Estimation (MLE), the distribution of x_0 can be derived. The combined use of q and p forms a variational auto-encoder, and the Variational Lower Bound (VLB) can be written as follows (Nichol and Dhariwal 2021):

$$\begin{aligned} L_{\text{vlb}} = &-\overbrace{\log p_\theta(x_0|x_1)}^{L_0} + \overbrace{D_{KL}(q(x_T|x_0)||p(x_T))}^{L_T} \\ &+ \sum_{t>1} \overbrace{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}^{L_{t-1}}. \end{aligned} \tag{3}$$

With $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_t^{s=0} \alpha_s$, the marginal can be written as follow (Nichol and Dhariwal 2021; Ho, Jain, and Abbeel 2020):

$$\begin{aligned} &q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \\ &\qquad \text{where } \; x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \; \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \end{aligned} \tag{4}$$

By applying Bayes' theorem, the posterior $q(x_{t-1}|x_t, x_0)$ can be determined with $\tilde{\beta}_t$ and $\tilde{\mu}_t(x_t, x_0)$, defined as follows (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021):

$$\begin{aligned} &\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, \\ &\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha}_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \\ &q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \boldsymbol{I}). \end{aligned} \tag{5}$$

According to (Ho, Jain, and Abbeel 2020), the L_{t-1} can be calculated as:

$$L_{t-1} = \mathbb{E}_{q(x_{1:T})}\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2\right] + C. \tag{6}$$

There are several ways to parameterize $\mu_\theta(x_t, t)$. One approach is to predict the noise ϵ with a neural network, and use Eqs. (4) and (5) to derive (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021):

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)). \tag{7}$$

Predicting ϵ with a reweighted loss function has proven effective (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021):

$$L_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon}[||\epsilon - \epsilon_\theta(x_t, t)||^2]. \tag{8}$$

In particular, as (Nichol and Dhariwal 2021) mentioned, IDDPM could generate a vector v containing one component pre dimension, and this vector v composes the new variances, $\Sigma_\theta(x_t, t)$ in Eq. 2:

$$\Sigma_\theta(x_t, t) = \exp(v \log \beta_t + (1-v) \log \tilde{\beta}_t). \tag{9}$$

Since L_{simple} doesn't reply on $\Sigma_\theta(x_t, t)$ (Nichol and Dhariwal 2021), the two loss functions L_{vlb} and L_{simple} can be simply combined into a new hybrid objective by introducing a hyperparameter λ_1 to scale one of them, i.e.,

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda_1 L_{\text{vlb}}. \tag{10}$$

This reparameterization technique allows the diffusion model to reconstruct Gaussian distributions, enabling the transformation of the logit from a single iteration into logits analogous to those from multiple experiments.

Bernoulli Approximation

In traditional settings, a single iteration of forward propagation yields one probability estimate. However, we can view each iteration as an independent and replicable trial, treating it as a Bernoulli Trial (BT). By conducting multiple independent trials within a single forward propagation, we can obtain more precise results. When the number of BTs is large enough, the distribution of the BT results approximates a Gaussian distribution, as described by the De Moivre–Laplace theorem (Walker 2006). This allows us to incorporate the CLT to estimate the mean of the Gaussian distribution, representing the results of BTs. This mean can be predicted, enabling us to simulate this Bernoulli process in a single iteration instead of multiple training runs.

A Bernoulli trial has exactly two possible outcomes: "success" (i.e., the positive case) and "failure" (i.e., the negative case). Let p be the probability of the positive case. In a typical CNN, logits are generated and then converted into probabilities (for classification), confidence scores, and other expected outputs through functions like softmax and sigmoid. In an ideal scenario, the probability of the positive case $p = 1$. Therefore, each training iteration can be viewed as a BT, with the logit representing the expected value of a random variable following the Bernoulli distribution. We define this random variable as the Bernoulli logit y_{Blogit}, which can take two fixed values: positive Bernoulli logit y_{Blogit_+} and negative Bernoulli logit y_{Blogit_-}. The logit y_{logit} can be calculated using the following equation:

$$y_{\text{logit}} = \mathbb{E}(y_{\text{Blogit}}) = y_{\text{Blogit}_+} p + y_{\text{Blogit}_-}(1-p). \tag{11}$$

If $p = 1$, the logit is equal to the true value of the positive Bernoulli logits, i.e., $y_{\text{logit}} = y_{\text{Blogit}_+}$ as $n \to \infty$, according to the CLT. We refer to this process as the Bernoulli approximation.

Repeating the BT independently n times, the possible values of the total number of positive outcomes range from 0 to n. Let $\hat{p}$ denote the estimated probability of a positive outcome in n trials, we have

$$\mathbb{E}(\hat{p}) = p, \qquad Var(\hat{p}) = \frac{p(1-p)}{n}, \tag{12}$$

where $\mathbb{E}(\hat{p})$ denotes the expected value of $\hat{p}$, $Var(\hat{p})$ denotes the variance of $\hat{p}$. We incorporate a CNN to construct a Gaussian distribution by learning its mean and variance. According to the De Moivre–Laplace theorem (Walker 2006), as n increases, the distribution of $\hat{p}$ increasingly resembles a Gaussian distribution:

$$\hat{p} \sim \mathcal{N}(p, \sqrt{\frac{p(1-p)}{n}}). \tag{13}$$

According to Eqs. (13) and (12), the mean of the Gaussian distribution is equal to the true value of the success probability of BT as $n \to \infty$, according to the CLT. Under optimal conditions, y_{logit} can be calculated through multiple BTs. However, since the Bernoulli logit follows a Gaussian distribution, y_{logit} can be calculated as follows:

$$\begin{aligned} y_{\text{logit}} &= \mathbb{E}(y_{\text{Blogit}}) \\ &= y_{\text{Blogit}_+}\hat{p} + y_{\text{Blogit}_-}(1-\hat{p}). \end{aligned} \tag{14}$$

In an ideal scenario, the probability $\hat{p}$ is 1, meaning each BT would succeed, otherwise is 0. Thus, y_{logit} is equal to the true value of y_{Blogit_+} as $n \to \infty$, according to the CLT. Following Eq. (13), after applying the softmax or sigmoid function, the mean of the Gaussian distribution can be used to categorize outputs as 0 or 1, thereby supervising the CNN model. Additionally, the variance of the Gaussian distribution would be zero in this ideal case, allowing us to simulate multiple BTs with their mean and variance in only one iteration. Through this entire process, logits are transformed into a Gaussian distribution.

Bernoulli-Gaussian Decision Block

Building on the concepts of Bernoulli approximation and IDDPMs, we introduce the Bernoulli-Gaussian decision block into the deep model training process, shown in Figure 1. This Bernoulli-Gaussian Decision Block (BGDB) aims to enhance the stability and performance of discriminative classifiers by leveraging the mean of logits from multiple experiments to supervise a single learning process.

Meanwhile, we employ IDDPM to construct and refine the probability distributions of BTs. The entire construction process can be supervised by the L_{hybrid}. Compared to DDPM, IDDPM can generate both mean and variance, this approach perfectly aligns with Bernoulli Approximation. Simultaneously, through the inverse diffusion process, we sample the mean μ_{output} and variance σ_{output} at time t_0, where $p(x_0) \sim (\mu_{\text{output}}, \sigma_{\text{output}})$, from the logit produced by the backbone. After applying the softmax or sigmoid function, μ_{output} of the Gaussian distribution is required to categorize outputs as 0 or 1 to supervise the CNN model. Ideally, σ_{output} should be 0, allowing us to construct a multiple BTs with μ_{output} and σ_{output} in a single iteration. Let L_μ and L_σ be the loss targeting at mean μ_{output} and variance σ_{output} (for Bernoulli approximation). Let L_{BCE} denote the Balanced Cross-Entropy (BCE) loss, L_{MSE} denote the Mean Squared Error (MSE) loss, F represents the softmax or sigmoid function. Given that the mean is represented as a probability while the variance is numerically zero, the mean loss is calculated using BCE, whereas the variance loss is obtained using MSE. Especially, L_y, task-specific loss such as Dice loss in segmentation tasks, can be calculated from the logit of a single learning process.

Thus, the entire loss function $\mathcal{L}$ for the model with BGDB module is calculated as follows:

$$\begin{aligned} \mathcal{L} &= L_{\text{y}} + \lambda_2 L_{\text{hybrid}} \\ &+\lambda_3 \left(\overbrace{(L_{\text{BCE}} F(\mu_{\text{output}}),\ \text{label})}^{L_\mu} + \overbrace{L_{\text{MSE}}(\sigma_{\text{output}}, 0)}^{L_\sigma} \right). \end{aligned} \tag{15}$$

Since $F(\mu_{\text{output}})$ is a probability, it can also be used in other loss functions, such as Dice loss (Milletari, Navab, and Ahmadi 2016). This module is added after the logits and before the softmax to compute the loss function during training. After training, this structure is removed, and predictions are made using the original network, without any burden in inference. The label may encompass options such as the category of a single object or pixel.

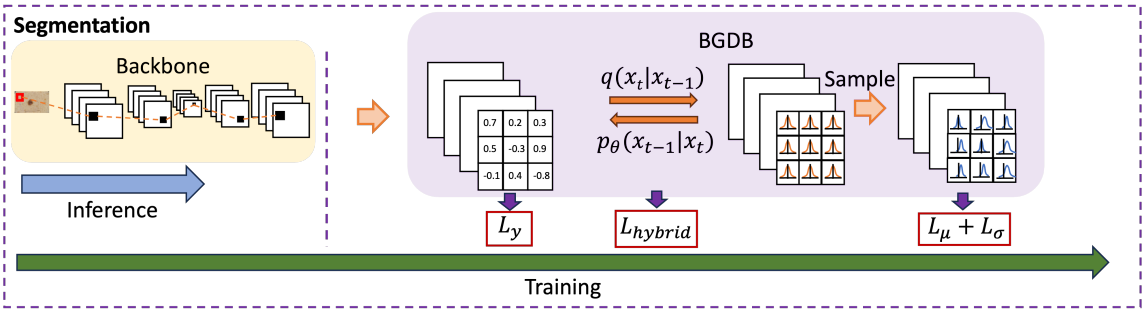

Figure 1: Workflow for performing segmentation tasks. The total loss in the training pipeline includes L_y (task-specific loss), L_{hybrid} (for IDDPM), and $L_\mu + L_\sigma$ (for Bernoulli approximation). During training, the input image is first processed by the backbone model to obtain the logits for a single experiment, supervised by L_y. These logits are then used to train the IDDPM model, resulting in a latent space composed entirely of Gaussian noise, which is supervised by L_{hybrid}. By sampling from this latent space, a Gaussian distribution for CLT's results is formed. This process is supervised by $L_\mu + L_\sigma$. After training, only the backbones are retained for inference.

This construction process begins by minimizing a loss function to generate a new distribution from an existing one. Because the output derived from the loss function adheres to the same distribution as the input, supervised learning is primarily needed for the mean and variance of the noise. By controlling these parameters, the entire diffusion process can transform one distribution into another desired distribution, the probability of multiple successful BT experiments. In generative tasks, the input distribution for diffusion models is initially fixed. However, in classification problems, the input logits are obtained through supervised learning, which can introduce instability. By leveraging the learning process of diffusion models, we use the distribution of logits from multiple experiments to supervise the logits obtained from a single training session. This approach aims to stabilize and enhance training by supervising the process with multiple experimental logits derived from a single training instance.

Experiment

We evaluate the proposed method across various imaging tasks, including both classification and segmentation.

Urban and General Scene Segmentation

Datasets We utilized **Cityscapes** (Cordts et al. 2016) and PASCAL Visual Object Classes (VOC) Challenge (**Pascal VOC**) (Everingham et al. 2010) datasets for this task. The Cityscapes dataset addresses the need for detailed semantic understanding by providing annotated stereo video sequences from 50 cities. It includes 5,000 images with high-quality pixel-level annotations, making it well-suited for evaluating segmentation methods that leverage extensive, high-quality labeled data. The Pascal VOC dataset offers publicly accessible images and annotations along with standardized evaluation software. For segmentation tasks, each test image requires predicting the object class for each pixel, with "background" designated for pixels that do not belong to any of the twenty specified classes.

Compared Methods For our experiments on the Cityscapes and Pascal VOC datasets, we utilized the DeepLabV3 framework (Chen et al. 2017, 2018), following the experimental protocols outlined in the original papers. We evaluated the performance using three distinct backbones: MobileNet (Howard et al. 2017), ResNet101 (He et al. 2016), and HRNet (Wang et al. 2020). This approach allowed us to systematically assess the model's adaptability and efficacy across varied scenarios.

Experimental Settings Our training regimen consisted of 30,000 iterations, with each batch comprising 16 samples. All input images were uniformly cropped to dimensions of 256×256. We employed the cross-entropy loss function, coupled with a learning rate of 0.01 and a weight decay of 1e-4. Stochastic Gradient Descent (SGD) (Robbins and Monro 1951) was used as the optimizer throughout the training process to ensure optimal convergence and model refinement. For testing, the images from the Cityscapes dataset retained their original size, while the Pascal VOC images were resized to 256×256. Model performance was assessed using the Mean Intersection over Union (mIoU) metric.

In this study, all models were trained on an NVIDIA A100 GPU with 80 GB of memory. The hyperparameters were set as follows: λ_1 to 1×10^{-3}, and both λ_2 and λ_3 to 1. These settings were used for all subsequent experiments.

Experimental Results As illustrated in Table 1, on both Cityscapes and Pascal VOC, all models experienced moderate improvements. Specifically, the models showed an increase in performance ranging from 0.08% to 1.48% on the Cityscapes dataset and from 0.21% to 0.41% on the Pascal VOC dataset. These results demonstrate the effectiveness of the proposed Bernoulli-Gaussian decision block in enhancing the performance.

Skin Lesion Segmentation

Datasets We used the International Skin Imaging Collaboration (ISIC) dataset (Tschandl, Rosendahl, and Kittler 2018; Codella et al. 2019) for skin lesion segmentation. The ISIC dataset is the world's largest collection of dermoscopic skin images. The ISIC 2018 challenge, held at the MICCAI

Model	Cityscapes	Pascal VOC
	mIoU (%)	mIoU (%)
DLP_MobileNet	63.61 ± 0.72	61.78 ± 0.57
+ours	65.09 ± 0.38 **+1.48**	62.17 ± 0.65 **+0.39**
DLP_ResNet101	72.00 ± 0.36	69.74 ± 0.49
+ours	72.08 ± 0.10 **+0.08**	69.95 ± 0.52 **+0.21**
DLP_HRNet	72.09 ± 0.47	69.87 ± 0.42
+ours	72.92 ± 0.37 **+0.82**	70.28 ± 0.57 **+0.41**

Table 1: The results of mIoU (Mean $\pm$ Std) for urban and general scene segmentation on Cityscapes and Pascal VOC datasets. "DLP" denotes "DeepLabv3+".

conference, included three tasks and featured over 12,500 images. The challenge attracted 900 registered users, with 115 submissions for lesion segmentation, 25 for lesion attribute detection, and 159 for disease classification.

Compared Methods We evaluated the Bernoulli-Gaussian decision block across several classical and state-of-the-art 2D medical segmentation models using the ISIC dataset. These models include U-Net (Ronneberger, Fischer, and Brox 2015), Attention U-Net (Oktay et al. 2018), U-Net++ (Zhou et al. 2019), FCN (Liu et al. 2018), ResUNet (Diakogiannis et al. 2020), and UNETR (Hatamizadeh et al. 2022), all implemented using the MONAI framework (Cardoso et al. 2022). The baseline models were trained using the Dice loss (Milletari, Navab, and Ahmadi 2016), while "+ours" models were trained with our proposed loss function in addition to the Dice loss.

Experimental Settings We utilized the training, validation, and test datasets provided by the ISIC 2018 challenge. These datasets were combined and then randomly split into training and testing sets in a 5:2 ratio (2,600 images for training and 1,094 for testing). We performed 5-fold cross-validation, selecting the optimal model from each fold's validation set. The selected models were then evaluated on the testing set, and we recorded the mean and variance of performance metrics across the 5 folds. For data augmentation, we normalized pixel values to a range between 0 and 255 and resized the images to 256×256 to meet the input requirements of the proposed block.

The models were trained using the AdamW (Loshchilov and Hutter 2017) optimizer with a weight decay of $1e{-}5$ and a learning rate of $1e{-}4$. Each model underwent 10,000 iterations of training, with the goal of achieving the highest Dice scores. This approach enabled a thorough comparative analysis between the baseline and enhanced models by the proposed decision block. Model performance was assessed using Dice score (Milletari, Navab, and Ahmadi 2016), which measures the overlap between the predicted segmentation and the ground truth.

Experimental Results Table 2 shows the Dice scores for the models on the ISIC dataset. Upon integrating the proposed block, we observed performance improvements across most models, with the exception of U-Net++, which experienced a marginal decline of -0.29%. The performance improvements for the other models ranged from 0.6% to 4.74%.

Beyond Segmentation: Skin Lesion Classification

Datasets and Compare Methods Similar to the skin lesion segmentation task, we used the ISIC 2018 challenge dataset for skin lesion classification (Tschandl, Rosendahl, and Kittler 2018; Codella et al. 2019). We conducted empirical analyses across a spectrum of prominent models to assess the efficacy of the Bernoulli-Gaussian decision block. The models included DenseNet (Huang et al. 2017), ResNet (He et al. 2016), Vision Transformer (ViT) (Dosovitskiy et al. 2020), EfficientNet (Tan and Le 2019), and SENet (Hu, Shen, and Sun 2018).

Experimental Settings We utilized the entirety of the ISIC 2018 dataset, amalgamating all available images before randomly partitioning them into training and testing sets in a 5:1 ratio while maintaining the original distribution. We conducted rigorous 5-fold cross-validation within the test set. From each fold, we selected the model with the highest accuracy on the validation set for final testing. We documented the mean and variance of accuracy and the Area Under the ROC Curve (AUC) across the 5-fold models.

To rigorously evaluate the model's performance, we employed basic data augmentation strategies, including random rotations up to 15 degrees, flipping, and zooming in or out by a scale of 0.1 with a 50% probability. We used the Adam optimizer (Kingma and Ba 2014) with a learning rate of 1e-5. Each model was trained for 50 epochs to achieve the highest levels of accuracy. This approach allowed for an exhaustive comparative analysis between models with and without the proposed block, enhancing our understanding of their respective performances.

Experimental Results Table 3 shows the accuracy and AUC scores for the models on the ISIC dataset. The experimental findings indicate that, aside from slight declines in the ViT (0.04% accuracy), all other models exhibited performance enhancements. The improvements ranged from 0.54% to 1.6% in accuracy and from 0.24% to 0.74% in AUC.

Ablation Experiments

To thoroughly understand the impact of different loss functions on the performance of our model, we conducted ablation experiments using the U-Net model with various combinations of loss functions. We set all hyperparameters to 1. The loss functions evaluated included the task-specific loss (i.e., Dice loss L_{Dice}), the diffusion loss L_{hybrid} for IDDPM, the BT loss for Bernoulli approximation (i.e., $L_\mu + L_\sigma$). The specific combinations tested were: U-Net (i.e., Only L_{Dice}), $L_{\text{Dice}} + L_\mu + L_\sigma$ (i.e., no diffusion loss), $L_{\text{Dice}} + L_{\text{hybrid}}$ (i.e., no BT loss), all losses combined (i.e., $L_{\text{Dice}} + L_\mu + L_\sigma + L_{\text{hybrid}}$). For each combination of loss functions, we trained the U-Net model on the ISIC dataset using the same experimental settings as described previously.

Table 4 shows the Dice scores for the different combinations of loss functions. Our analysis revealed that incorporating all loss functions led to the best performance, with

UNETR	FCN	U-Net	ResUNet	A*U-Net	U-Net++
77.62 ± 2.7	73.52 ± 2.7	69.17 ± 1.9	75.82 ± 1.24	72.47 ± 1.86	80.78 ± 0.83
+ours	+ours	+ours	+ours	+ours	+ours
80.30 ± 2.45	75.04 ± 2.7	73.91 ± 1.19	76.44 ± 0.84	73.49 ± 0.98	80.49 ± 1.22
+2.68	**+1.52**	**+4.74**	**+0.62**	**+1.02**	**-0.29**

Table 2: Dice (Mean (%) ± Std) for skin lession segmentation on ISIC dataset. "A*U-Net" denotes "Attention U-Net".

Model	**ISIC**	
	Accuracy (%)	AUC (%)
DenseNet169	68.34 ± 0.59	89.36 ± 0.50
+ours	69.83 ± 1.05 **+1.49**	90.10 ± 0.49 **+0.74**
ViT	69.28 ± 0.59	89.06 ± 0.24
+ours	69.24 ± 0.65 **-0.04**	89.30 ± 0.18 **+0.24**
ResNet50	66.66 ± 0.90	88.44 ± 0.29
+ours	68.26 ± 1.01 **+1.60**	88.90 ± 0.85 **+0.46**
SENet154	69.64 ± 1.04	89.79 ± 0.38
+ours	70.18 ± 1.50 **+0.54**	90.23 ± 0.56 **+0.44**
EfficientNet	65.78 ± 0.68	88.44 ± 0.49
+ours	67.14 ± 0.60 **+1.36**	89.10 ± 0.21 **+0.66**

Table 3: Accuracy and AUC (Mean (%) ± Std) for skin lession classification on ISIC dataset.

a Dice score improved by 4.74%. This underscores the profound impact of combining multiple loss functions on model performance.

U-Net	No diffusion loss	No BT loss	All losses combined
69.17±1.90	72.23± 1.14	73.17±0.86	**73.91±1.19**

Table 4: Dice scores of different loss combinations for skin lession segmentation on the ISIC dataset.

We also explored the impact of hyperparameter settings of λ_2 and λ_3 in Eq. 15. This ablation study was conducted on the first fold of the U-Net experiment's dataset. The hyperparameter preceding the initial model's Dice loss was fixed at 1. Initially, with λ_3 set to 1, the hyperparameter before the λ_2 was varied at 0.01, 0.5, 1, and 2 to observe changes in model performance. Similarly, with the Dice loss and λ_3 fixed at 1, the λ_2 was adjusted to 0.01, 0.5, 1, and 2, allowing us to evaluate its effect on the model's performance. Figure 2 shows that the model performs most stably when the hyperparameters for all three losses are consistent, which aligns with our default setting. Additionally, fine-tuning the hyperparameters within the range of 0.5 to 1 can be beneficial for achieving optimal performance.

Discussion

While our model achieved moderate advancements in segmentation tasks, it encounters several limitations. Statistically, the proposed Bernoulli-Gaussian decision block relies on a sufficiently large number of trials, n, to satisfy the formula under optimal conditions. The block can determine n to ensure the validity of the mean of the Gaussian distribution.

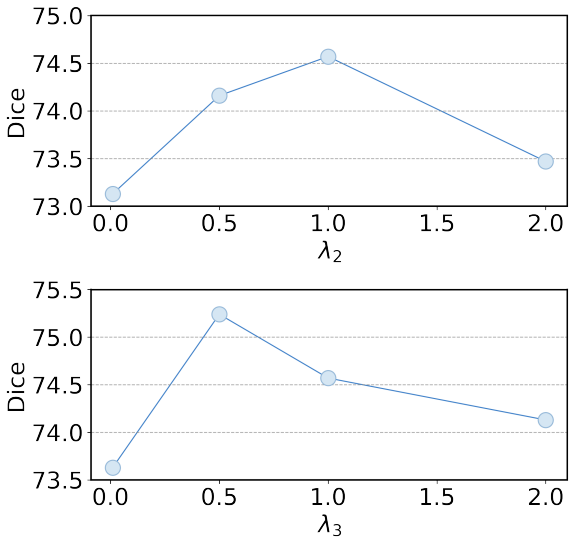

Figure 2: Impact of hyperparameters λ_2 and λ_3 on segmentation model performance, with the other hyperparameters fixed at 1.

The IDDPM, while faster than DDPM in training and inference, faces slowdown due to simultaneous training and inference within the proposed block. This limits the diffusion model's time steps to at least 25, complicating training. Our experiments focused on 2D segmentation (256 × 256 images), and the IDDPM's large parameter count makes it impractical for 3D images, requiring excessively small image sizes unsuitable for 3D segmentation.

The U-Net architecture's encoder-decoder structure limits the predicted value dimensions to powers of 2, complicating classification tasks. On the ISIC dataset, we explored resizing logits from 1×1 to 64×64 and averaging the reconstructed results from IDDPM. Although the model shows potential for classification, the averaging operation appears redundant. Given IDDPM's complexity, our BGDB block employs default hyperparameters tailored for the generative model. Unlocking its full potential requires meticulous tuning, which is more feasible with a simplified BGDB block.

Conclusion

We proposed a novel Bernoulli-Gaussian Decision Block, which constructs experimental probability distributions using the diffusion model, achieving modest segmentation improvements and showing promise for classification tasks.

References

Baranchuk, D.; Rubachev, I.; Voynov, A.; Khrulkov, V.; and Babenko, A. 2021. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*.

Berkson, J. 1951. Why I prefer logits to probits. *Biometrics*, 7(4): 327–339.

Cardoso, M. J.; Li, W.; Brown, R.; Ma, N.; Kerfoot, E.; Wang, Y.; Murrey, B.; Myronenko, A.; Zhao, C.; Yang, D.; et al. 2022. Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, Z.; Rotskoff, G.; Bruna, J.; and Vanden-Eijnden, E. 2020. A dynamical central limit theorem for shallow neural networks. *Advances in Neural Information Processing Systems*, 33: 22217–22230.

Choi, J.; Kim, S.; Jeong, Y.; Gwon, Y.; and Yoon, S. 2021. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*.

Codella, N.; Rotemberg, V.; Tschandl, P.; Celebi, M. E.; Dusza, S.; Gutman, D.; Helba, B.; Kalloo, A.; Liopyris, K.; Marchetti, M.; et al. 2019. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Cramer, J. S. 2003. The origins and development of the logit model. *Logit models from economics and other fields*, 2003: 1–19.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Davidson, I.; and Aminian, M. 2004. Using the Central Limit Theorem for Belief Network Learning. In *AI&M*.

Diakogiannis, F. I.; Waldner, F.; Caccetta, P.; and Wu, C. 2020. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162: 94–114.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.

Gagniuc, P. A. 2017. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H. R.; and Xu, D. 2022. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 574–584.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Hu, Y.; Zhong, Z.; Wang, R.; Liu, H.; Tan, Z.; and Zheng, W.-S. 2021. Data augmentation in logit space for medical image classification with limited training data. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24*, 469–479. Springer.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, Z.; Shao, W.; Wang, X.; Lin, L.; and Luo, P. 2021. Rethinking the pruning criteria for convolutional neural network. *Advances in Neural Information Processing Systems*, 34: 16305–16318.

Jaakkola, T.; and Haussler, D. 1998. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, 11.

Jarzynski, C. 1997. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56(5): 5018.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G.; Unterthiner, T.; Mayr, A.; and Hochreiter, S. 2017. Self-normalizing neural networks. *Advances in neural information processing systems*, 30.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lee, J.; Bahri, Y.; Novak, R.; Schoenholz, S. S.; Pennington, J.; and Sohl-Dickstein, J. 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*.

Liu, S.; Xu, D.; Zhou, S. K.; Pauly, O.; Grbic, S.; Mertelmeier, T.; Wicklein, J.; Jerebko, A.; Cai, W.; and Comaniciu, D. 2018. 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, 851–858. Springer.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Miao, X.; Yuan, X.; Pu, Y.; and Athitsos, V. 2019. l-net: Reconstruct hyperspectral images from a snapshot measurement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4059–4069.

Miao, X.; Zhen, X.; Liu, X.; Deng, C.; Athitsos, V.; and Huang, H. 2018. Direct shape regression networks for end-to-end face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5040–5049.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. Ieee.

Ng, A.; and Jordan, M. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14.

Nguyen, Q.; Vu, T.; Tran, A.; and Nguyen, K. 2024. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *Advances in Neural Information Processing Systems*, 36.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Oktay, O.; Schlemper, J.; Folgoc, L. L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N. Y.; Kainz, B.; et al. 2018. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Perina, A.; Cristani, M.; Castellani, U.; Murino, V.; and Jojic, N. 2012. Free energy score spaces: Using generative information in discriminative classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7): 1249–1262.

Raina, R.; Shen, Y.; Mccallum, A.; and Ng, A. 2003. Classification with hybrid generative/discriminative models. *Advances in neural information processing systems*, 16.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Rubinstein, Y. D.; Hastie, T.; et al. 1997. Discriminative vs Informative Learning. In *KDD*, volume 5, 49–53.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Tschandl, P.; Rosendahl, C.; and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1): 1–9.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Walker, H. M. 2006. DE MOIVRE ON THE LAW OF NORMAL PROBABILITY.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3349–3364.

Wei, H.; Xie, R.; Cheng, H.; Feng, L.; An, B.; and Li, Y. 2022. Mitigating neural network overconfidence with logit normalization. In *International conference on machine learning*, 23631–23644. PMLR.

Wu, H.; and Klabjan, D. 2021. Logit-based uncertainty measure in classification. In *2021 IEEE International Conference on Big Data (Big Data)*, 948–956. IEEE.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.

Zhang, S.-Q.; Wang, F.; and Fan, F.-L. 2022. Neural network gaussian processes by increasing depth. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhao, X.; Yan, X.; Yu, A.; and Van Hentenryck, P. 2020. Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. *Travel behaviour and society*, 20: 22–35.

Zhou, Z.; Siddiquee, M. M. R.; Tajbakhsh, N.; and Liang, J. 2019. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, 39(6): 1856–1867.

Zimmermann, R. S.; Schott, L.; Song, Y.; Dunn, B. A.; and Klindt, D. A. 2021. Score-based generative classifiers. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.