# ENHANCING UNCERTAINTY ESTIMATION AND INTERPRETABILITY WITH BAYESIAN NON-NEGATIVE DECISION LAYER

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Although deep neural networks have demonstrated significant success due to their powerful expressiveness, most models struggle to meet practical requirements for uncertainty estimation. Concurrently, the entangled nature of deep neural networks leads to a multifaceted problem, where various localized explanation techniques reveal that multiple unrelated features influence the decisions, thereby undermining interpretability. To address these challenges, we develop a **B**ayesian **N**onnegative **D**ecision **L**ayer (**BNDL**), which reformulates deep neural networks as a conditional Bayesian non-negative factor analysis. By leveraging stochastic latent variables, the BNDL can model complex dependencies and provide robust uncertainty estimation. Moreover, the sparsity and non-negativity of the latent variables encourage the model to learn disentangled representations and decision layers, thereby improving interpretability. We also offer theoretical guarantees that BNDL can achieve effective disentangled learning. In addition, we developed a corresponding variational inference method utilizing a Weibull variational inference network to approximate the posterior distribution of the latent variables. Our experimental results demonstrate that with enhanced disentanglement capabilities, BNDL not only improves the model's accuracy but also provides reliable uncertainty estimation and improved interpretability.

## 1 INTRODUCTION

Over the last decade, deep neural networks(DNNs) have achieved significant success and have been widely applied across various research domains (LeCun et al., 2015). As these applications expand, quantifying uncertainty in their predictions has gained importance, particularly within the realm of AI safety (Amodei et al., 2016). A key goal of uncertainty quantification is to ensure that neural networks assign low confidence to test cases poorly represented by training data or prior knowledge (Gal & Ghahramani, 2016). One approach to this challenge is Bayesian neural networks, which treat model parameters as random variables. Although progress has been made in developing approximate inference methods for Bayesian neural networks (Li et al., 2016; Louizos & Welling, 2017; Shi et al., 2017), computational scalability continues to pose a significant obstacle. Alternatively, deterministic methods such as deep ensembles (Lakshminarayanan et al., 2017) and dropout (Gal & Ghahramani, 2016) have been proposed. However, these approaches necessitate running full DNNs multiple times during the testing phase, resulting in high computational costs and increased inference time(FAN et al., 2021).

In addition to uncertainty evaluation, there is a growing demand for interpretability in DNNs, aimed at helping users understand the rationale behind model's decisions through various tools. To address this need, numerous advanced techniques have been developed to deliver localized explanations, often by identifying key features or regions of the input that significantly influence the model's output (Olah et al., 2017; Yosinski et al., 2015). However, a key challenge in this field is that neurons within well-trained DNNs tend to be multifaceted (Nguyen et al., 2016), meaning they respond to multiple, often unrelated, features. This phenomenon may arise from the entangled nature of DNNs, wherein multiple features are utilized for various tasks. To address this challenge, significant efforts have been made in the literature, including the use of specialized regularizers aimed at promoting feature disentanglement (Nguyen et al., 2016), and the employment of sparse linear decision layers to select the most important features (Wong et al., 2021). While these approaches have led to notable advancements, they warrant further exploration from both practical and theoretical standpoints. Specifically, the method of employing a sparse linear decision layer may impair task performance due to the loss of essential dimensional information. Furthermore, while empirical results suggest that sparsity enhances model interpretability and mitigates the challenges posed by multifaceted neurons (Moakhar et al., 2023), these claims lack rigorous theoretical support.

While DNNs face significant challenges, non-negative factor analysis (NFA), such as Poisson factor analysis (Zhou et al., 2012), benefits from its sparse stochastic latent variables, exhibiting strong capabilities in discovering disentangled concepts that enhance model interpretability (Lee & Seung, 1999), as well as performing effective uncertainty evaluation. Thus, a compelling motivation exists to transfer the good properties of NFA to DNN. However, integrating the two models may be challenging due to their differing paradigms: NFA typically employs a shallow structure and linear transformations, whereas DNNs utilize a deep architecture with non-linear functions. Nevertheless, DNNs can be decomposed into a deep non-linear feature extractor and a linear decision layer, and recent research indicates that the linear decision layer is sufficient for capturing uncertainty (Kristiadi et al., 2020; Dhuliawala et al., 2023b; Joo et al., 2020; Parekh et al., 2022). Further, (Wong et al., 2021) demonstrates that leveraging a sparse decision layer can enhance a model's interpretability through innovative techniques. This line of inquiry motivates the use of NFA as a linear decision layer, integrated with DNNs as feature extractors, to enhance both uncertainty quantification and model's interpretability.

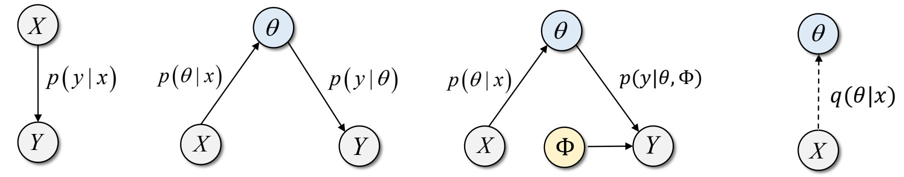
With the considerations above, we developed a **B**ayesian **N**onnegative **D**ecision **L**ayer (**BNDL**), designed to empower DNNs with enhanced interpretability and uncertainty estimation capabilities. Specifically, under the categorical likelihood, the label is factorized into a gamma-distributed factor score matrix (local latent variables) and a corresponding gamma-distributed factor loading matrix (global latent variables). The former represents the latent representation of the observation, while the latter captures the interaction between the latent variables and the label. Given the challenge of intractable posterior distributions for the latent variables, we introduce a deep Weibull variational neural network to effectively approximate the gamma-distributed latent variables (Zhang et al., 2018). All parameters are trained using stochastic gradient boosting (SGB) within a variational inference framework. Furthermore, we provide theoretical guarantees for the model's disentanglement capabilities, which enhances its interpretability. Additionally, our complexity analysis indicates that the increase in computational effort is minimal during both the training and uncertainty testing phases. To assess the efficacy of the proposed model, we conducted evaluations on a wide range of benchmark datasets using image classification tasks. The experimental results demonstrate that the proposed approach consistently outperforms standard classification models and offers superior uncertainty estimation. The main contributions of the paper can be summarized as follows:

- We develop a flexible **B**ayesian **N**onnegative **D**ecision **L**ayer (**BNDL**) for deep neural networks, empower its interpretability and uncertainty estimation capabilities.
- The complexity analysis shows that the computational overhead introduced by BNDL is minimal compared to DNNs. Further, we take theory analysis to verify its disentangled properties.
- We assessed the effectiveness of BNDL across multiple datasets, including CIFAR-10, CIFAR-100, and ImageNet-1k. BNDL not only preserves or even enhances baseline performance but also facilitates uncertainty estimation and improves the interpretability of neurons.

## 2 RELATED WORK

### 2.1 UNCERTAINTY EVALUATION

In supervised learning, existing research has focused on modeling conditional distributions beyond just the conditional mean, particularly in quantifying predictive uncertainty through various approaches. Ensemble methods (Liu et al., 2021; Lakshminarayanan et al., 2017) quantify uncertainty by combining multiple neural networks with stochastic outputs. Similarly, bayesian neural networks achieve this by placing distributions over network parameters, reflecting model plausibility given the data (Blundell et al., 2015; Hernández-Lobato & Adams, 2015; Kingma et al., 2015; Gal &



(a) DNNs

(b) Extend DNNs with $\theta$

(c) Generative Model of BNDL

(d) Inference Model of BNDL

Figure 1: Illustration of the graphical models. ***1(a):*** the predictive process of output $Y$ for the baseline Deep Neural Network; ***1(b):*** the generative model of DNNs with introducing stochastic latent variable $\boldsymbol{\theta}$ ; ***1(c):*** the generation model of the Bayesian non-negative decision layer; and ***1(d):*** corresponding approximate inference for latent variables $\boldsymbol{\theta}$.

Ghahramani, 2016; Tomczak et al., 2021). Among these methods, increasing attention toward on enhancing the efficiency of Bayesian approaches and reducing their computational costs within neural networks, giving rise to two closely related fields: Bayesian Last Layer (BLL) and evidential deep learning (EDL). BLL has demonstrated effectiveness in uncertainty estimation for limited data and out-of-distribution scenarios, offering efficient and deterministic training and predictions (Wilson et al., 2016; Harrison et al., 2020b;a; 2024). Similarly, EDL quantifies uncertainty by constructing evidence distributions in the final layer, but introduces priors in distinct ways, such as modeling ”evidence” vectors with task-specific priors (Sensoy et al., 2018; Malinin et al., 2020), or directly learning task-specific parameter distributions (Amini et al., 2020). Building on these principles, BNDL models the decision layer as a Bayesian generative model to enable efficient uncertainty estimation. However, it distinguishes itself by leveraging a Gamma prior, which enhances the network's sparsity and disentanglement, thereby improving its interpretability.

### 2.2 INTERPRETABILITY TOOLS FOR DEEP NEURAL NETWORK

The goal of neural network interpretability is to identify the mechanisms underlying DNN's decision-making processes. The related research ranges from approaches which link abstract concepts to structural network components, such as specific neurons, for example via visualization (Yosinski et al., 2015; Nguyen et al., 2016), to approaches which aim to trace individual model outputs on a per-sample basis such as local surrogates (Ribeiro et al., 2016) and salience maps (Simonyan et al., 2013). However, as noted by various recent studies, these local attributions can be easy to fool or may otherwise fail to capture global aspects of model behavior (Adebayo et al., 2018; Leavitt & Morcos, 2020; Wong et al., 2021). A major confounder for interpretability is that neurons in a well-trained DNN are often multifaceted (Nguyen et al., 2016; Moakhar et al., 2023), responding to various, often unrelated, features. In contrast, our approach ensures that the identified high-level concepts—*i.e.*, the deep features utilized by the sparse decision layer—fully determine the model's behavior.

## 3 BAYESIAN NON-NEGATIVE DECISION LAYER

This section first reformulate the traditional DNNs as a latent variable model (Sec.3.1) and then provides a detailed description of the proposed Bayesian non-negative decision layer, which consists of the generative model (Sec.3.2) and the variational inference network (Sec.3.3). Followed by the description of the variational inference (Sec.3.4).

### 3.1 PRELIMINARIES ON DEEP NEURAL NETWORKS

We first adopt a latent variable model to re-examine the DNNs, which are commonly tackled by training domain-specific neural networks with a sigmoid or softmax output layer. Dataset samples $x$ are mapped deterministically by a neural network $f$ to a real vector $\boldsymbol{z} = f(x)$, which is transformed in the softmax layer to a point on the simplex $\Delta^{|\boldsymbol{y}|}$, a discrete distribution over class labels $\boldsymbol{y} \in \mathcal{Y}$:

$$p(\boldsymbol{y} \mid x) = \frac{\exp\{\boldsymbol{z} w_{\boldsymbol{y}}^T + b_{\boldsymbol{y}}\}}{\sum_{\boldsymbol{y}' \in \mathcal{Y}} \exp\{\boldsymbol{z} w_{\boldsymbol{y}'}^T + b_{\boldsymbol{y}'}\}} \tag{1}$$

Where the $w_y$ and $b_y$ represent the weights and biases of the final fully connected layer. Thus, the classification process can be viewed as a generative process for label $y$, as shown in Fig.1(a). Previous works (Dhuliawala et al., 2023a; Joo et al., 2020) have shown that a softmax classifier is a special case of Equation 2:

$$p(\boldsymbol{y} \mid x) = \int_{\boldsymbol{z}} p(\boldsymbol{y} \mid \boldsymbol{z}) p(\boldsymbol{z} \mid x) \tag{2}$$

The neural network $f$, up to the softmax layer, parameterizes $p(\boldsymbol{z} \mid x)$ as a delta distribution $\delta_{\boldsymbol{z}-f(x)}$; the softmax layer input is considered as a sample from $p(\boldsymbol{z} \mid x)$; and $p(\boldsymbol{y} \mid z)$ is defined by the softmax layer.

While the output softmax layer can be viewed as a categorical distribution, the limited randomness originating solely from the model's output layer is often inadequate for capturing complex dependencies (Chung et al., 2015). Moreover, DNNs with a softmax layer frequently encounter the issue of overconfidence (Kristiadi et al., 2020; Liu et al., 2020; Guo et al., 2017), which complicates the provision of high-quality uncertainty evaluations. Additionally, the data feature $\boldsymbol{z}$ and parameter $w_{\boldsymbol{y}}$ are often entangled and represented as dense vectors, with multiple features contributing to the prediction, leading to a multifaceted phenomenon in localized explanations (Nguyen et al., 2016). These challenges motivate us to develop a Bayesian non-negative decision layer in the following sections.

### 3.2 BAYESIAN NON-NEGATIVE DECOMPOSITION LAYER

Building on the latent variable model of the softmax classification task, we can further reformulate DNNs as a Non-negative Factor Analysis, referred to as the **Bayesian Non-negative Decision Layer (BNDL)**. Firstly, to better capture complex dependence and aleatoric uncertainty, referring to the inherent randomness in the data that cannot be explained away, we can intuitively extend the generative model of original DNNs with stochastic latent variables $\boldsymbol{\theta}$ by modeling latent representation $z$ with a distribution. Thus, as illustrated in Fig.1(b), the Eq. 2 is improved to

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{x}) \tag{3}$$

To further account for epistemic uncertainty, which refers to the uncertainty inherent in the model itself, we treat the weights of the final fully connected layer as stochastic latent variables. The generative model is then defined as follows, and its graphical model is shown in Fig.1(c):

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \int_{\boldsymbol{\theta}, \boldsymbol{\Phi}} p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{x}) p(\boldsymbol{\Phi}) \tag{4}$$

This formulation bears similarities to factor analysis (Yu et al., 2008), where both $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ in classical DNNs are commonly sampled from a Gaussian distribution, making the generative model akin to Gaussian factor analysis. However, while Gaussian factor analysis can effectively evaluate uncertainty, it struggles to achieve disentangled representation learning with dense latent variables (Moran et al., 2021), which is crucial for model interpretability(Nguyen et al., 2016).

Considering the Gamma distribution possesses both non-linearity and non-negativity, we use the Gamma distribution as the prior distribution for $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$. This choice allows for a more comprehensive capture of complex relationships within the model (Zhou et al., 2015; Duan et al., 2021; 2024), while also accommodating the characteristics of sparse non-negative variables, thereby enhancing the disentanglement and interpretability of the learned representations $\boldsymbol{\theta}$ and corresponding decision layers $\boldsymbol{\Phi}$ (Lee & Seung, 1999). Specifically, given data samples $\boldsymbol{x}_j$ and their corresponding one-hot label $\boldsymbol{y}_j \in \mathbb{R}_+^C$, where $C$ is the number of classes, we can factorize $\boldsymbol{y}_j$ under the category likelihood as follows:

$$\boldsymbol{y}_j \mid \boldsymbol{\theta}_j \sim \text{Category}\left(\boldsymbol{\theta}_j \boldsymbol{\Phi}\right), \quad \boldsymbol{\theta}_j \mid \boldsymbol{x}_j \sim \text{Gamma}\left(f_\theta(x_j), 1\right), \quad \boldsymbol{\Phi} \sim \text{Gamma}\left(1, 1\right). \tag{5}$$

where $\boldsymbol{\theta}_j \in \mathbb{R}_+^K$ is the factor score matrix, each column of which encodes the relative importance of each atom in a sample $\boldsymbol{x}_j$; and $\boldsymbol{\Phi} \in \mathbb{R}_+^{K \times C}$ is the factor loading matrix, each column of which is a factor encoding the relative importance of each term. Intuitively, in a classification problem, the $k$-th column of $\boldsymbol{\Phi} \in \mathbb{R}_+^{K \times C}$, denoted as $\boldsymbol{\phi}_k \in \mathbb{R}_+^C$, represents the $k$-th distribution across all classes. Although in our formulation the prior of the latent variable $\boldsymbol{\theta}_j$ is modulated by the input

$\boldsymbol{x}_j$, the constraint can be easily relaxed to allow the latent variable to be statistically independent of the input variable (Sohn et al., 2015; Kingma et al., 2014). Therefore, we can simply define the data-independent prior of the latent variable $\boldsymbol{\theta}$ as $f_\theta(x_j) = 1$.

### 3.3 VARIATIONAL INFERENCE NEURAL NETWORK

We have constructed the generative process of $\boldsymbol{y}$, which includes the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$, in Section 3.2. Due to the intractable posterior in BNDL, we build a Weibull variational inference to approximate the posterior of $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$.

**Weibull Approximate Posterior:** While the gamma distribution appears suitable for the posterior distribution due to its encouragement of sparsity and adherence to the nonnegative condition, it is difficult to reparameterize a gamma distributed random variable (Kingma & Welling, 2014; Knowles, 2015; Ruiz et al., 2016). Hence, we use the reparameterizable Weibull distribution (Zhang et al., 2018) to approximate the posterior for the gamma latent variables, mainly due to the following considerations: $\boldsymbol{i}$), the Weibull distribution has a simple reparameterization so that it is easier to optimize; $\boldsymbol{ii}$) the Weibull distribution is similar to a gamma distribution, capable of modeling sparse, skewed and positive distributions. Specifically, the latent variable $x \sim \text{Weibull}(k, \lambda)$ can be easily reparameterized as:

$$x = \lambda(-\ln(1-\varepsilon))^{1/k}, \qquad \varepsilon \sim \text{Uniform}(0,1). \tag{6}$$

Where $\lambda$ and $k$ are the scale and shape parameter of Weilbull distribution respectively; $\boldsymbol{iii}$), The KL divergence from the gamma to Weibull distributions has an analytic expression as:

$$\begin{aligned}\text{KL}\left(\text{Weibull}(k,\lambda)\|\text{Gamma}(\alpha,\beta)\right) = &\frac{\gamma\alpha}{k} - \alpha\log\lambda + \log k + \beta\lambda\Gamma\left(1+\frac{1}{k}\right) \\ &-\gamma - 1 - \alpha\log\beta + \log\Gamma(\alpha)\end{aligned} \tag{7}$$

where $\gamma$ is the Euler-Mascheroni constant.

**Local latent variables Inference Network:** As shown in Fig. 1(d), the variational inference network construct the variational posterior:

$$q\left(\boldsymbol{\theta}_j \mid \boldsymbol{x}_j\right) = \text{Weibull}\left(\boldsymbol{k}_j, \boldsymbol{\lambda}_j\right) \tag{8}$$

where the inference network can be defined as :

$$\boldsymbol{k}_j = \text{Softplus}\left(\boldsymbol{f}_k(\boldsymbol{h}_j)\right), \quad \boldsymbol{\lambda}_j = \text{ReLu}\left(\boldsymbol{f}_\lambda(\boldsymbol{h}_j)\right)/\exp\left(1 + 1/\boldsymbol{k}_j\right), \quad \boldsymbol{h}_j = \boldsymbol{f}_{NN}(\boldsymbol{x}_j). \tag{9}$$

where $\boldsymbol{h}_j$ is an extracted feature with deep neural networks, such as ResNet, which can be seen as a deep feature extractor; Let $\boldsymbol{f}.(\cdot)$ denotes the neural network, where $\boldsymbol{f}_{NN}$ is the feature extractor, encompassing all layers from the input layer to the penultimate layer, and $\boldsymbol{f}_\lambda$ and $\boldsymbol{f}_k$ are the network to infer the scale and shape parameters of Weibull distribution, respectively. The Softplus function, defined as $\log(1+\exp(\cdot))$, is applied element-wise non-linearity to ensure positive Weibull shape parameters. The Weibull distribution is used to approximate the gamma-distributed conditional posterior, and its parameters $\boldsymbol{k}_j^{(l)} \in \mathrm{R}_+^{K_l}$ and $\boldsymbol{\lambda}_j^{(l)} \in \mathrm{R}_+^{K_l}$ are inferred by the bottom-up data information using the neural networks.

**Global latent variables inference Network:** For the same reason, we also use Weibull distributions to approximate the posteriors of global latent variables $\boldsymbol{\Phi} \in \mathbb{R}_+^{K \times C}$, formulated as

$$q(\boldsymbol{\Phi} \mid -) = \text{Weibull}\left(\boldsymbol{k}_\Phi, \boldsymbol{\lambda}_\Phi\right) \tag{10}$$

where the inference network can be expressed as

$$\boldsymbol{k}_\Phi^{(l)} = \text{Softplus}(\boldsymbol{W}_1), \quad \boldsymbol{\lambda}_\Phi^{(l)} = \text{Relu}(\boldsymbol{W}_2)/\exp\left(1 + 1/\boldsymbol{k}_\Phi\right). \tag{11}$$

Note that $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are two random initialized matrices, with dimensions consistent with those of $\boldsymbol{\Phi}$.

**Connection with Non-Negative Matrix Factorization:** Equations 8 and 11 ensure that $\mathbb{E}[\boldsymbol{\theta}_j] = \boldsymbol{\lambda}_j$ and $\mathbb{E}[\boldsymbol{\Phi}] = \boldsymbol{\lambda}_\Phi$. Therefore, instead of sampling $\boldsymbol{\theta}_j$ and $\boldsymbol{\Phi}$ from their respective distributions, substituting their expectations makes the mapping equivalent to that of standard non-negative matrix factorization, resulting in $y_j = \boldsymbol{\lambda}_j \boldsymbol{\lambda}_\Phi$. In other words, if we let the shape parameter $k$ of the Weibull distribution approach infinity—implying that the variance of the latent variables approaches zero, and the distribution collapses into a point mass concentrated at the expectation, then the proposed stochastic decision layer reduces to non-negative matrix factorization.

### 3.4 VARIATIONAL INFERENCE

For BNDL, given the model parameters referred to as $\boldsymbol{\Omega}$, which consist of the parameters in the generative model and inference network, the marginal likelihood of the dataset $(X, Y)$ is defined as:

$$\boldsymbol{p}\,(Y \mid X) = \int\int \prod_{j=1}^{J} \boldsymbol{p}\,(\boldsymbol{y}_j \mid \boldsymbol{\theta}_j, \boldsymbol{\Phi}) \boldsymbol{p}\,(\boldsymbol{\theta}_j) \boldsymbol{d\theta}_{j=1}^{J} \boldsymbol{d\Phi} \tag{12}$$

The inference task is to learn the parameters of the generative model and the inference network. Similar to VAEs, the optimization objective of the BNDL can be achieved by maximizing the evidence lower bound (ELBO) of the log-likelihood as:

$$\begin{aligned}\mathcal{L}(Y) = &\sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{\theta}_j \mid \boldsymbol{x}_j)} \left[\ln p\,(\boldsymbol{y}_j \mid\ \boldsymbol{\theta}_j, \boldsymbol{\Phi})\right] \\ &- \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{\theta}_j \mid \boldsymbol{x}_j)} \left[\ln \frac{q\,(\boldsymbol{\theta_j} \mid \boldsymbol{x}_j)}{p\,(\boldsymbol{\theta}_j\,)}\right] - \mathbb{E}_{q(\boldsymbol{\Phi} \mid -)} \left[\ln \frac{q\,(\boldsymbol{\Phi} \mid -)}{p\,(\boldsymbol{\Phi}\,)}\right]\end{aligned} \tag{13}$$

where the first term is the expected log-likelihood of the generative model, which ensures reconstruction performance, and the last two term is the Kullback–Leibler (KL) divergence that constrains the variational distribution $q(-)$ to be close to its prior $p(-)$. The parameters in the Generalized GBN can be directly optimized by advanced gradient algorithms, like SGD (Kingma, 2014).

**Complexity analysis** The modification of the last layer in the base deep neural network models results in a minimal increase in the number of parameters. Consequently, the added space complexity is insignificant. In details, the time complexity of ResNet is primarily determined by its convolution layers. Assuming the input image has a height $H$, width $W$, convolution kernel size $K \times K$, number of input channels $C_{\text{in}}$, and number of output channels $C_{\text{out}}$, the time complexity of ResNet can be simplified as: $O\left(\sum_{l=1}^{L} C_{in}^{(l)} \times C_{out}^{(l)} \times H^{(l)} \times W^{(l)} \times K^{(l)} \times K^{(l)}\right)$. For the proposed BNDL, the added time complexity introduced by the KL divergence computation between the prior and posterior of both local and global latent variables is $O(C_{\text{out}}^{(L)})$ and $O(C_{\text{out}}^{(L)} \times C)$, respectively. These terms are nearly negligible compared to the overall time complexity of ResNet. Moreover, at the testing stage, traditional ensemble methods and dropout approaches require running the entire neural network multiple times to evaluate uncertainty. In contrast, the BNDL only requires a single forward pass to infer the variational posterior, followed by multiple samples for uncertainty evaluation, significantly reducing computational demands.

## 4 THEORETICAL GUARANTEES FOR BNDL

We provide theoretical guarantees for BNDL from the perspective of identifiable features. As described in Sec.3.3, BNDL can be viewed as a Non-negative Matrix Factorization (NMF) problem. From this perspective, its objective function $\boldsymbol{p}\,(Y \mid \{\boldsymbol{\Phi}, \boldsymbol{\theta}, X\})$ can be further reformulated as:

$$\min_{\boldsymbol{\theta} \geq 0, \boldsymbol{\Phi} \geq 0} \|Y - \boldsymbol{\theta\Phi}\|_F^2 \tag{14}$$

In the realm of non-negative matrix factorization, many studies have aimed to establish the identifiability and uniqueness of a decomposition $\boldsymbol{\theta\Phi}$, up to permutation and scaling. Recently, (Gillis & Rajkó, 2023) demonstrated that a subset of the columns of $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ can be identified and made unique under more relaxed conditions. We will show how BNDL adheres to the criteria set forth in (Gillis & Rajkó, 2023), thereby enabling the learning of partially identifiable features.

**Proposition 1.** *(Gillis & Rajkó, 2023) The $k$-th column of $\boldsymbol{\theta}$ is identifiable under the two assumptions:*

- ***Selective Window**: There exists a row of $\boldsymbol{\Phi}$, say the $j$-th, such that $\boldsymbol{\Phi}(j,:) = \alpha e_{(k)}^T$ for $\alpha > 0$, where $e_{(k)}^T$ represents the $k$-th standard row vector in vector space.*
- ***Sparsity Constrain**: The $k$-th column of $\boldsymbol{\Phi}$ contains at least $r - 1$ entries equal to zero, where $r$ is the rank of $Y$.*

The selective window assumption means that the column in $\boldsymbol{\theta}$ corresponding to the $\boldsymbol{\Phi}(j,:)$ appears uniquely in the dataset, which is reasonable in many applications (Gillis, 2020). *e.g.*, $\boldsymbol{\theta}$ can be viewed as the latent class in classification task, (Wang et al., 2024) achieve full identifiability by positing that each latent class has a unique sample. In the context of selective window assumption, we only need to assume the presence of a single latent class with a unique sample to satisfy the selective window assumption, which makes it more feasible and easier to achieve. For the sparsity constraint, the use of a Gamma prior and a ReLU activation function in $\boldsymbol{\Phi}$ within the BNDL framework, as outlined in 5 and 11, enforces sparsity during the training process. Moreover, instead of relying on the parameter norm (Wong et al., 2021), which often leads to performance degradation, we propose a more scalable and effective adaptive activation function to achieve sparsity. Specifically, we employ $f(x) = \text{ReLU}(x-\alpha)$, where $\alpha$ is a predefined constant, set as a hyperparameter. This approach offers a more flexible mechanism for inducing sparsity without sacrificing model performance. Similarly, we can demonstrate the partial identifiability of $\boldsymbol{\Phi}$, as it is often considered to be the transpose of $\boldsymbol{\theta}$(Fu et al., 2017; HaoChen et al., 2021).

In conclusion, BNDL follows the aforementioned assumptions, and its optimization objective promotes the partial identifiability of the learned features and decision layer, thereby enhancing their disentanglement capability. We validated our theoretical guarantees through experiments presented in Section 5.2. More details for the theoretical guarantees can be found in Appendix A.2.

## 5 EXPERIMENTS

**Experiment Setup** Following (Wong et al., 2021), we analyze the following models: (a) ResNet classifiers—ResNet-50 trained on ImageNet-1k (Deng et al., 2009; Russakovsky et al., 2015) and Places-10 (a subset of Places365 (Zhou et al., 2017)), and ResNet-18 trained on CIFAR-10/100 (Krizhevsky, 2009); (b) a ViT-based model (Dosovitskiy et al., 2021) with pretrained weights from Hugging Face[1]. Baselines are detailed in Section 5.1. We evaluate accuracy and uncertainty for quantitative analysis and use LIME for qualitative insights into BNDL predictions. Setup details, including hyperparameters and datasets, are provided in Appendix A.1.

**Uncertainty evaluation metric.** We estimate uncertainty using a hypothesis testing approach (FAN et al., 2021). This method provides interpretable $p$-values, enabling practical deployment for binary uncertainty decisions. A prediction's certainty is determined by comparing its $p$-value against a threshold. To evaluate uncertainty estimates, we use the Patch Accuracy vs. Patch Uncertainty (PAvPU) metric (Mukhoti & Gal, 2018), which defined as PAvPU = $(n_{ac} + n_{iu})/(n_{ac} + n_{au} + n_{ic} + n_{iu})$, where $n_{ac}$, $n_{au}$, $n_{ic}$, and $n_{iu}$ represent the counts of accurate-certain, accurate-uncertain, inaccurate-certain, and inaccurate-uncertain samples, respectively. Higher PAvPU values indicate that the model reliably produces accurate predictions with high certainty and inaccurate ones with high uncertainty.

**Sparsity Measurement** We measure the sparsity of the final decision layer weights. Since the weights in our method are non-negative, we follow the approach of (Wong et al., 2021) and (Wang et al., 2024), considering weights greater than $1 \times 10^{-5}$ as non-sparse (denoted as $l_{non}$) and the remaining values as sparse (denoted as $l_{sparse}$). The sparsity is calculated as $nnz = l_{non}/(l_{non} + l_{sparse})$, where a smaller value indicates a higher level of sparsity.

---

[1] https://huggingface.co/google/vit-base-patch16-224

Table 1: Overall model accuracy across different datasets, with BNDL being our method. We use ResNet-50 as the baseline for ImageNet-1k, and ResNet-18 as the baseline for CIFAR-10 and CIFAR-100. Vit refers to vit-base-patch16-224.

| **Model** | **CIFAR-10** | | **CIFAR-100** | | **ImageNet-1k** | |
|---|---|---|---|---|---|---|
| | ACC | PAvPU | ACC | PAvPU | ACC | PAvPU |
| ResNet | 94.98 ± 0.12 | - | 74.62 ± 0.23 | - | 75.33 ± 0.14 | - |
| MC Dropout | 94.54 ± 0.03 | 78.83 ± 0.12 | 78.12 ± 0.06 | 64.41 ± 0.22 | 75.98 ± 0.08 | 76.50 ± 0.02 |
| BM | 94.07 ± 0.07 | 93.98 ± 0.3 | 75.81 ± 0.34 | 77.13 ± 0.67 | - | - |
| CARD | 90.93 ± 0.02 | 91.11 ± 0.04 | 71.42 ± 0.01 | 71.48 ± 0.03 | 76.20 ± 0.00 | 76.29 ± 0.01 |
| ResNet-BNDL | **95.54 ± 0.08** | **95.58 ± 0.20** | **79.82 ± 0.13** | **81.1 ± 0.21** | **77.01 ± 0.14** | **77.66 ± 0.03** |
| ViT-Base | 95.51 ± 0.03 | - | 84.15 ± 0.03 | - | 80.33 | - |
| ViT-BNDL | **96.34 ± 0.04** | **97.01 ± 0.02** | **85.16 ± 0.03** | **86.37 ± 0.11** | **81.29 ± 0.02** | **82.50 ± 0.03** |

## 5.1 CLASSIFICATION PERFORMANCE

Specifically, for each dataset, we conducted the following experiments:

***(a) Performance and Uncertainty Evaluation*** We replaced the decision layers in the network with BNDL and performed supervised training from scratch. The results are shown in Table 1. The baseline models are grouped into two categories: ***1)*** Uncertainty estimation networks, including Bernoulli MC Dropout(Gal & Ghahramani, 2016), BM (Joo et al., 2020) and CARD (Han et al., 2022) ***2)*** Dense decision layer baselines: including ViT-Base (Dosovitskiy et al., 2021) (We used the pretrained weights for the vit-base-patch16-224 only, modifying the decision layer for continued training.) and ResNet (He et al., 2016). It is important to note that the goal of BNDL is not to achieve a substantial improvement in the model's performance, but rather to preserve the model's performance while enhancing its interpretability and uncertainty estimation capabilities.

***(b) Impact of Sparsity.*** We replaced the decision layers of a pre-trained model with BNDL, froze the existing feature layers, and fine-tuned only the parameters of the BNDL. The results across different datasets are illustrated in Figure 3, where we compare BNDL (shown in blue) with the Debuggable Network(Wong et al., 2021) (shown in orange), both of them utilize the same backbone with different sparse decision layers.

### 5.1.1 PERFORMANCE AND UNCERTAINTY EVALUATION

In Table 1 , we show accuracy and PAvPU. Our model reports the mean and variance across 5 different random seeds, while the results of other models are reported from previous papers if available. Since we directly used [1] to test on ImageNet-1k, the variance term is not provided in the table. We can observe that: ***1)*** By leveraging stochastic latent variables to capture complex dependencies, BNDL consistently outperformed all datasets and demonstrated improved performance across various widely-used architectures, including ResNet and ViT. ***2)*** The integration of BNDL endowed the model with the capability for uncertainty estimation, as evidenced by the improvements in PAvPU metrics when compared to several strong baselines. ***3)*** BNDL exhibits scalability and can be extended to larger datasets, such as ImageNet-1k, as demonstrated by the complexity analysis.

**Relation between Uncertainty and Accuracy** To further investigate the relationship between prediction uncertainty and downstream performance, we conducted an ablation study. Specifically, we chose ResNet-BNDL to visualize and sorted the test set of imagenet based on their evaluation uncertainty and divided the set into 10 subsets accordingly. We then calculated the average accuracy and corresponding average uncertainty for each subset, and plotted the results in the line chart shown in Figure 2, where each point represents the result of a subset. Next, we select images from the subsets with the highest and lowest uncertainty, and visualize their LIME explanations (we visualize the features with the maximum weights of the image activations), as shown on the right side of Figure 2. The line chart illustrates a clear ***negative correlation*** between uncertainty and accuracy: higher uncertainty corresponds to lower accuracy. This suggests that the model provides reliable uncertainty estimates, helping to avoid potential misclassifications. In the visualization, we observe that the model made correct predictions for images with low uncertainty, while for images with

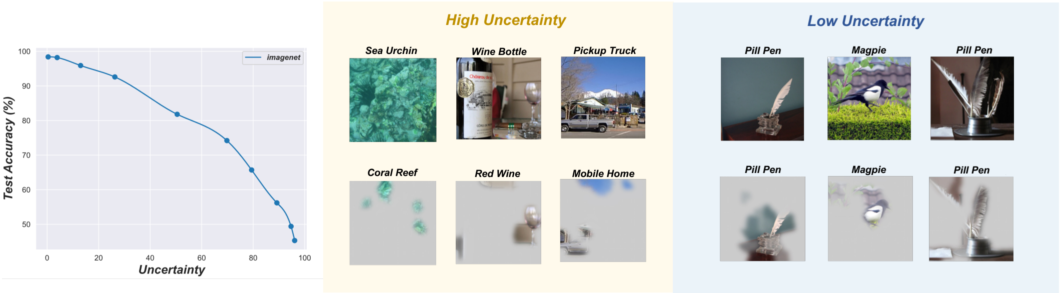
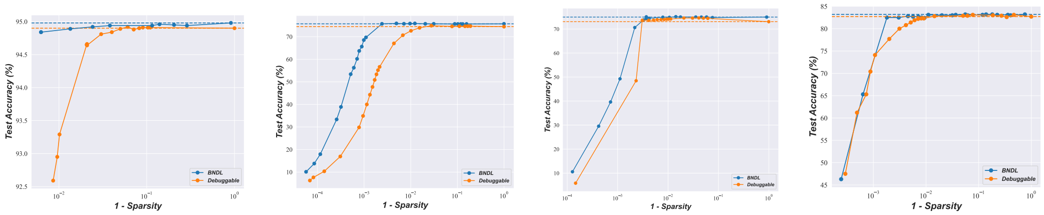


Figure 2: The leftmost line chart illustrates the average uncertainty and accuracy across subsets of the ImageNet test set. The middle and right panels sample images from the subsets with the highest and lowest uncertainty, as defined by the curve. The top row shows the original images with ground truth labels, while the bottom row displays the model's predictions alongside LIME visualizations.

high uncertainty, the visualizations reveal the causes of misclassification, *e.g.*, in the image of a wine bottle, the model primarily focused on the wine glass filled with red wine in the background, leading to a misclassification as red wine. The Uncertainty Vs Acc results for additional datasets and ViT-BNDL are provided in the Appendix A.3.

### 5.1.2 IMPACT OF SPARSITY



(a) Cifar-10 (b) Imagenet-1k (c) Cifar-100 (d) Places-10

Figure 3: Sparsity-accuracy trade-offs for BNDL and Debuggable Network (Wong et al., 2021). Each point on the curve represents a BNDL classifier. Horizontal dashed lines indicate the fully dense accuracy for each network. The x-axis shows the proportion of non-sparse weights, with higher values indicating denser distributions, while the y-axis represents test accuracy on each dataset.

The results of fine-tuning on the original model are shown in Figure 3. For each dataset, we control the sparsity according to the activation function mentioned in Eq.11, resulting in the Sparsity vs. accuracy curve. The blue lines represent our method, while the orange lines represent the Debuggable Network (Wong et al., 2021), which also employs the idea of a sparse decision layer. It can be observed that at the same level of sparsity, our model overall outperforms the Debuggable Network across different datasets. Additionally, across different datasets, the decision layer can be made substantially sparser—by up to two orders of magnitude—with minimal impact on accuracy (cf. Figure 3(a)). For instance, when the sparsity of the ImageNet-1k classifier is 0.0024, the network's classification accuracy still reaches 75.7%.

## 5.2 INTERPRETABILITY EVALUATION

**Disentangled representation learning** To validate the disentanglement on real-world data, we adopt an unsupervised disentanglement metric SEPIN@$k$ (Do & Tran, 2019). SEPIN@$k$ measures how each feature $\boldsymbol{\theta}_i$ is disentangled from others $\boldsymbol{\theta}_{\neq i}$ by computing their conditional mutual information with the top $k$ features, *i.e.*, SEPIN@$k = 1/k \sum_{i=1}^{k} I(x, \boldsymbol{\theta}_{r_i} | \boldsymbol{\theta}_{\neq r_i})$, which are estimated with InfoNCE lower bound (Oord et al., 2018) implemented following (Wang et al., 2024).
As shown in Table 2, BNDL features exhibit much better disentanglement than ResNet-50 across all top-$k$ dimensions. The advantage is more pronounced when considering the top features, as learned features also contain noise dimensions. This verifies the disentanglement of learned features, as

Table 2: Feature disentanglement score on Imagenet-1k, where @k denotes the top-k dimensions. Values are scaled by $10^2$, we use ResNet-50 as the baseline for BNDL

| | SEPIN@1 | SEPIN@10 | SEPIN@100 | SEPIN@1000 | SEPIN@all |
|---|---|---|---|---|---|
| ResNet50 | 1.50 ± 0.02 | 1.03 ± 0.01 | 0.60 ± 0.01 | 0.31 ± 0.01 | 0.23 ± 0.01 |
| ResNet50-BNDL | **2.59** ± **0.03** | **2.12** ± **0.01** | **1.30** ± **0.01** | **0.65** ± **0.01** | **0.44** ± **0.01** |

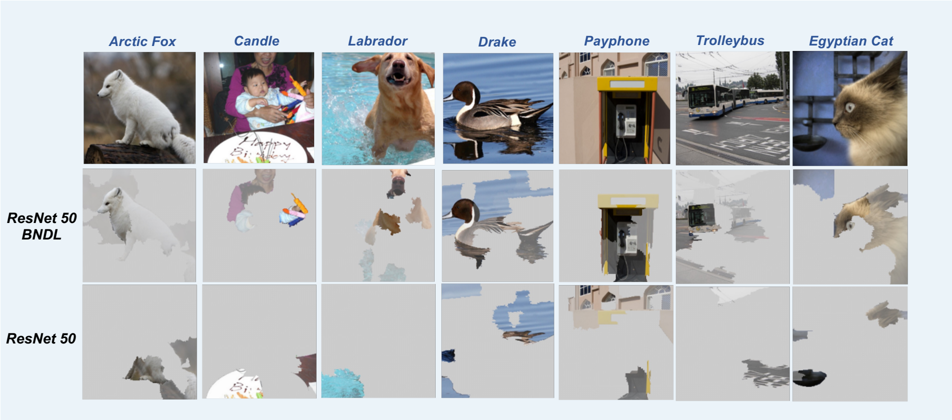


Figure 4: The LIME visualizations for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, show that BNDL's features align more closely with the semantic meaning of true labels, suggesting more disentangled representations. As we selected the top-10 super-pixels for visualization, the results may include some less significant super-pixels; this issue is alleviated when we reduce the number of top-$k$ super-pixels.

analyzed in Sec.4. BNDL indeed provides better feature disentanglement on real-world data. The disentanglement results on other datasets can be found in Appendix A.3.

**Visualization** We visualized the feature representation $\boldsymbol{\theta}$ of BNDL and the baseline model (ResNet-50) for the same images in ImageNet-1k, as illustrated in Figure 4. Specifically, we selected the feature $\boldsymbol{\theta}$ with the highest activation for each image and applied the LIME visualization, in line with the approach used in Figure 2. The top row of Figure 4 shows the true categories of the corresponding images, the second row presents our visualization results, and the third row displays the visualization results of the baseline model. Overall, the visualization results of BNDL are more semantically meaningful compared to those of ResNet-50. For instance, in the image of a candle, BNDL successfully captures parts of the candle, while ResNet-50 only identifies the cake. Similar observations occur in other categories, and we provide additional visualization results in Appendix A.3. This finding aligns with the conclusions drawn in 4, suggesting that BNDL has learned more identifiable features through the constraint of sparsity.

## 6 CONCLUSION

We introduce BNDL as a simple and scalable Bayesian decision layer that excels in both uncertainty estimation and interpretability, while maintaining or improving accuracy across a range of tasks, including large-scale applications. With an efficient parameterization of the covariance-dependent variational distribution, BNDL enhances the flexibility of DNNs with only a slight increase in memory and computational cost. We demonstrate the broad applicability of BNDL on both ResNet-based and ViT-based models and show that BNDL achieves superior performance compared to these baselines. Notably, we provide both practical and theoretical guarantees for BNDL's ability to learn more disentangled and identifiable features. Based on these results, we believe BNDL can serve as an efficient alternative to decision layer in the versatile tool box of modules.

## 7 REPRODUCILTLY STATEMENT

The novel methods introduced in this paper are accompanied by detailed descriptions ( Sec.3), and their implementations are provided as anonymous downloadable source code in the supplementary materials.

## REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Shehzaad Dhuliawala, Mrinmaya Sachan, and Carl Allen. Variational classification. *Trans. Mach. Learn. Res.*, 2024, 2023a. URL `https://api.semanticscholar.org/CorpusID:258740937`.

Shehzaad Dhuliawala, Mrinmaya Sachan, and Carl Allen. Variational classification. *arXiv preprint arXiv:2305.10406*, 2023b.

Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. *arXiv preprint arXiv:1908.09961*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations (ICLR 2021)*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Zhibin Duan, Dongsheng Wang, Bo Chen, Chaojie Wang, Wenchao Chen, Yewen Li, Jie Ren, and Mingyuan Zhou. Sawtooth factorial topic embeddings guided gamma belief network. In *International Conference on Machine Learning*, pp. 2903–2913. PMLR, 2021.

Zhibin Duan, Tiansheng Wen, Muyao Wang, Bo Chen, and Mingyuan Zhou. A non-negative vae: the generalized gamma belief network. *arXiv preprint arXiv:2408.03388*, 2024.

XINJIE FAN, Shujian Zhang, Korawat Tanwisuth, Xiaoning Qian, and Mingyuan Zhou. Contextual dropout: An efficient sample-dependent dropout module. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=ct8_a9h1M`.

Xiao Fu, Kejun Huang, and Nicholas D. Sidiropoulos. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters*, 25:328–332, 2017. URL `https://api.semanticscholar.org/CorpusID:3483466`.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Nicolas Gillis. *Nonnegative matrix factorization*. SIAM, 2020.

Nicolas Gillis and Róbert Rajkó. Partial identifiability for nonnegative matrix factorization. *SIAM Journal on Matrix Analysis and Applications*, 44(1):27–52, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *ArXiv*, abs/2206.07275, 2022. URL `https://api.semanticscholar.org/CorpusID:249674651`.

Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Neural Information Processing Systems*, 2021. URL `https://api.semanticscholar.org/CorpusID:235367888`.

James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. Continuous meta-learning without tasks. *Advances in neural information processing systems*, 33:17571–17581, 2020a.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online bayesian regression. In *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*, pp. 318–337. Springer, 2020b.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. *arXiv preprint arXiv:2404.11599*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pp. 1861–1869. PMLR, 2015.

Kejun Huang, Nicholas D Sidiropoulos, and Ananthram Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2013.

Taejong Joo, Uijung Chung, and Minji Seo. Being bayesian about categorical probability. In *International Conference on Machine Learning*, 2020. URL `https://api.semanticscholar.org/CorpusID:211171911`.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second international conference on learning representations, ICLR*, volume 19, pp. 121, 2014.

DP Kingma. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*, 2014.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

David A Knowles. Stochastic gradient variational bayes for gamma approximating distributions. *arXiv preprint arXiv:1509.01631*, 2015.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pp. 5436–5446. PMLR, 2020.

A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Matthew L Leavitt and Ari Morcos. Towards falsifiable interpretability research. *arXiv preprint arXiv:2010.12016*, 2020.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Shiwei Liu, Tianlong Chen, Zahra Atashgahi, Xiaohan Chen, Ghada Sokar, Elena Mocanu, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Deep ensembling with no overhead for either training or testing: The all-round blessings of dynamic sparsity. *arXiv preprint arXiv:2106.14568*, 2021.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pp. 2218–2227. PMLR, 2017.

Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. Regression prior networks. *arXiv preprint arXiv:2006.11590*, 2020.

Arshia Soltani Moakhar, Eugenia Iofinova, and Dan Alistarh. Spade: Sparsity-guided debugging for deep neural networks. *arXiv preprint arXiv:2310.04519*, 2023.

Gemma E Moran, Dhanya Sridhar, Yixin Wang, and David M Blei. Identifiable deep generative models via sparse decoding. *arXiv preprint arXiv:2110.10804*, 2021.

Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jayneel Parekh, Sanjeel Parekh, Pavlo Mozharovskyi, Florence d'Alché Buc, and Gaël Richard. Listen to interpret: Post-hoc interpretability for audio networks with nmf. *Advances in Neural Information Processing Systems*, 35:35270–35283, 2022.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Francisco R Ruiz, Titsias RC AUEB, David Blei, et al. The generalized reparameterization gradient. *Advances in neural information processing systems*, 29, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. Kernel implicit variational inference. *arXiv preprint arXiv:1705.10119*, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Marcin Tomczak, Siddharth Swaroop, Andrew Foong, and Richard Turner. Collapsed variational bounds for bayesian neural networks. *Advances in Neural Information Processing Systems*, 34: 25412–25426, 2021.

Yifei Wang, Qi Zhang, Yaoyu Guo, and Yisen Wang. Non-negative contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016.

Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *International Conference on Machine Learning*, pp. 11205–11216. PMLR, 2021.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. Whai: Weibull hybrid autoencoding inference for deep topic modeling. *arXiv preprint arXiv:1803.01328*, 2018.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Mingyuan Zhou, Lauren Hannah, David Dunson, and Lawrence Carin. Beta-negative binomial process and poisson factor analysis. In *Artificial Intelligence and Statistics*, pp. 1462–1471. PMLR, 2012.

Mingyuan Zhou, Yulai Cong, and Bo Chen. The poisson gamma belief network. *Advances in Neural Information Processing Systems*, 28, 2015.

# A APPENDIX / SUPPLEMENTAL MATERIAL

## A.1 EXPERIMENTAL SETTING

All experiments are conducted on Linux servers equipped with 32 AMD EPYC 7302 16-Core Processors and 2 NVIDIA 3090 GPUs. Models are implemented in PyTorch version 1.12.1, scikit-learn version 1.0.2 and Python 3.7. The CIFAR-10, CIFAR-100, and ImageNet-1k datasets we used are all publicly available standard datasets. As for Places-10, it is a subset of Places365 (Zhou et al., 2017) containing the classes “airport terminal”, “boat deck”, “bridge”, “butcher’s shop”, “church-outdoor”, “hotel room”, “laundromat”, “river”, “ski slope” and “volcano”. We chose this dataset to ensure a fair comparison with DebuggableNetworks (Wong et al., 2021), as they also selected this subset.

- **Training from scratch setup** For ResNet-18 on CIFAR-10 and CIFAR-100, we set the batch size to 128, learning rate to 0.1, training epochs to 150, and weight decay to 5e-4. For ResNet-50 on ImageNet-1k, we set the batch size to 256, weight decay to 1e-4, epochs to 200, and learning rate to 0.1. For the calculation of PAvPU, we sampled 20 times to obtain 20 different logits, and then selected the top-2 classes based on their mean logits, performing a two-sample t-test to obtain the p-value. Following the approach used by CARD and MC Dropout models, we set the p-value threshold to 0.05 to determine whether a sample is classified as certain.
- **Fine-tuning setup** For Places-10, we set the learning rate to 0.1, batch size to 128, and epochs to 100. For ImageNet-1k, CIFAR-10, and CIFAR-100, we set the learning rate to 0.001 and epochs to 200.
- **Sparsity Vs Accuracy in Sec. 5.1.2** We follow the default settings of (Wong et al., 2021) when running the Debuggable baselines, and we run BNDL using the settings from the fine-tuning setup. It is worth mentioning that in Debuggable Networks, the sparsity of the decision layer is controlled via the elastic net, which adjusts the sparsity of decision layer weights through the regularization path, this kind of parameter norm often leads to performance degradation. In contrast, BNDL increases the sparsity of the decision layer by using a Gamma distribution as the prior. Additionally, we control the sparsity of the weights by applying an activation function to the weights, $w' = \text{ReLU}(w - \alpha)$, where $\alpha$ is a predefined constant set as a hyperparameter.

**Visualization tool (LIME)** Traditionally, LIME is used to obtain instance-specific explanations—i.e., to identify the super-pixels in a given test image that are most responsible for the model’s prediction. In our setting, we follow this intuition and use the following two step-procedure to obtain LIME-based feature interpretations: ***(i)*** We randomly select an image category and then randomly choose $K$ images from that category. For each image, we identify the feature with the maximum activation for visualization. Among them, our definition of maximum activation is that we select the maximum weight of the category index corresponding to the image in $\mathbf{\Phi}$, and select the feature neuron multiplied by this weight. ***(ii)*** Run LIME on each of these examples to identify relevant super-pixels. At a high level, this involves performing linear regression to map image super-pixels to the (normalized) activation of the deep feature (rather than the probability of a specific class as is typical).

## A.2 THEORETICAL GUARANTEES FOR BNDL

**Problem Formulation** As described in Sec.3.3, BNDL can be viewed as a Non-negative Matrix Factorization (NMF) problem. From this perspective, its objective function $\boldsymbol{p}\left(Y \mid \{\mathbf{\Phi}, \boldsymbol{\theta}, X\}\right)$ can be further reformulated as:

$$\min_{\boldsymbol{\theta} \geq 0, \mathbf{\Phi} \geq 0} \|Y - \boldsymbol{\theta}\mathbf{\Phi}\|_F^2 \tag{15}$$

This property enables NMF to accurately recover the ground truth factors that generated the data. Following the definition in (Gillis & Rajkó, 2023), we first define an exact NMF (that is, an errorless reconstruction) as follows:

**Definition 1.** *(Excat NMF of size r) Given a nonnegative matrix* $Y \in \mathbb{R}^{m\times n}$*, the decomposition* $\boldsymbol{\theta}\mathbf{\Phi}$ *where* $\boldsymbol{\theta} \in \mathbb{R}_+^{m\times r}$ *and* $\mathbf{\Phi} \in \mathbb{R}_+^{r\times n}$ *is an exact NMF of* $Y$ *of size r if* $Y = \boldsymbol{\theta}\mathbf{\Phi}$.

The formally defined identifiability of an Exact NMF is as follows:

**Definition 2.** *(Identifiability of Exact NMF) The Exact NMF of $Y = \boldsymbol{\theta}_* \boldsymbol{\Phi}_*$ of size $r$ is identifiable if and only if for any other Exact NMF of $Y = \boldsymbol{\theta}\boldsymbol{\Phi}$ of size $r$, there exists a permutation matrix $\Pi \in \{0,1\}^{r \times r}$ and a nonsingular diagonal scaling matrix $D$ such that:*

$$\boldsymbol{\theta} = \boldsymbol{\theta}_* \Pi D \quad \textit{and} \quad \boldsymbol{\Phi} = D^{-1} \Pi \boldsymbol{\Phi}_* \tag{16}$$

Intuitively, Definition 2 indicates that all columns of $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ must be identifiable. Achieving this typically requires very stringent conditions, such as the requirement for both $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ to satisfy the so-called sufficiently scattered condition (SSC) (Huang et al., 2013). Therefore, we concentrate on the partial identifiability of BNDL, which similarly ensures the identifiability and uniqueness of a subset of columns of $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ under more relaxed conditions.

**Partially Identifiable Features** To demonstrate that BNDL is partially identifiable, we first present the definition of partial identifiability in exact NMF.

**Definition 3.** *(Partial identifiability in Exact NMF) Let $Y = \boldsymbol{\theta}_* \boldsymbol{\Phi}_*$ be an exact NMF of $Y$ of size $r$. The $k$-th column of $\boldsymbol{\theta}_*$ is identifiable if and only if for any other Exact NMF of $Y = \boldsymbol{\theta}\boldsymbol{\Phi}$ of size $r$, there exits an index set $j$ and a scalar $\alpha > 0$ such that:*

$$\boldsymbol{\theta}(:, j) = \alpha \boldsymbol{\theta}_*(:, k) \tag{17}$$

*Similarly, we can define the identifiability of the kth column of $\boldsymbol{\Phi}_*$ using symmetry.*

Previous work (Gillis & Rajkó, 2023) has shown that Definition 3 can hold under two relatively relaxed assumptions.

**Proposition 2.** *(Gillis & Rajkó, 2023) Let $Y = \boldsymbol{\theta}_* \boldsymbol{\Phi}_*$ where $\boldsymbol{\theta}_* \in \mathbb{R}_+^{m \times r}$ and $\boldsymbol{\Phi}_* \in \mathbb{R}_+^{r \times n}$ with rank(Y) = r. Without loss of generality, assume $Y, \boldsymbol{\Phi}_*$ and $\boldsymbol{\theta}$ are column stochastic. The $k$-th column of $\boldsymbol{\theta}_*$ is identifiable if it satisfies the following two conditions:*

- *(**Selective Window**) There exists a row of $\boldsymbol{\Phi}_*$, say the $j$-th, such that $\boldsymbol{\Phi}_*(j, :) = \alpha e_{(k)}^T$ for $\alpha > 0$.*
- *(**Sparsity Constrain**) There exists a subset $\mathcal{J}$ of $r-1$ columns of $Y$, namely $Y(:, \mathcal{J})$, such that $rank(Y(:, \mathcal{J})) = r - 1$ and for all $j \in \mathcal{J}$:*

  $$\mathcal{F}_{\boldsymbol{\theta}_*}(\boldsymbol{\theta}_*(:, k)) \cap \mathcal{F}_{\boldsymbol{\theta}_*}(R(:, j)) = \emptyset \tag{18}$$

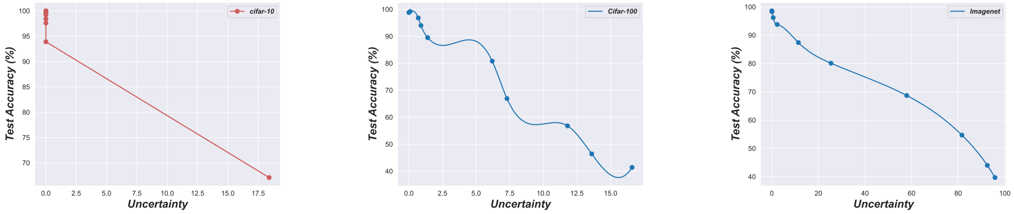
  *which means that the minimal face containing the $k$-th column of $\boldsymbol{\theta}_*$ does not intersect with the minimal faces containing the columns of $Y(:, \mathcal{J})$.*

$i)$***For the Selective Window assumption***: Intuitively, this assumption means that the column in $\boldsymbol{\theta}$ (latent class) corresponding to the $\boldsymbol{\Phi}_*(j, :)$ appears uniquely in the dataset, which is reasonable in many applications (Gillis, 2020). E.g., in classification tasks (Wang et al., 2024), the authors achieve full identifiability by positing that each latent class has a unique sample. In the context of selective window assumption, we only need to assume the presence of a single latent class with a unique sample to satisfy the selective window assumption, which makes it more feasible and easier to achieve. $ii)$***For the Sparsity Constrain***: This condition implies that the $k$-th column of $\boldsymbol{\Phi}_*$ contains at least $r-1$ entries equal to zero, namely, $\boldsymbol{\Phi}_*(\mathcal{J}, k) = 0$. Due to the use of a Gamma prior and a ReLU function in $\boldsymbol{\Phi}$ within BNDL, as shown in 5 and 11 respectively, $\boldsymbol{\Phi}$ is enforced to be sparse during the training process. Additionally, in Section 5.1.1, we demonstrate the high sparsity of BNDL, for instance, the 1-sparsity of decision weights for ImageNet is only 0.04. Only a small portion of the weights have a decisive impact on the final results, indicating that BNDL satisfies the sparsity constraint.

In summary, our theory demonstrates that BNDL satisfies the assumptions outlined in 2, facilitating the partial identifiability of the learned features.

### A.3 ADDITIONAL EXPERIMENTAL RESULTS

**Uncertainty Evaluation** We provide additional ablation study results on Figure 5, with experimental settings consistent with Sec.5.1.1. For plotting the curves, we used B-spline interpolation to generate smooth curves, setting $k = 3$ (cubic spline). The line charts illustrate a clear negative correlation between uncertainty and accuracy: higher uncertainty corresponds to lower accuracy. This suggests that the model provides reliable uncer- tainty estimates, helping to avoid potential misclassifications.



(a) ResNet-18 Cifar-10 (b) ResNet-18 Cifar-100 (c) ViT Imagenet-1k

Figure 5: Uncertainty Vs Test Acc Curve on other datasets and model.

Table 3: Feature score on Cifar-100, where @k denotes the top-k dimensions. Values are scaled by $10^2$

| Cifar-100 | SEPIN@1 | SEPIN@10 | SEPIN@100 | SEPIN@1000 | SEPIN@all |
|---|---|---|---|---|---|
| ResNet18 | $3.17 \pm 0.06$ | $2.40 \pm 0.03$ | $1.79 \pm 0.03$ | $1.21 \pm 0.01$ | $1.21 \pm 0.01$ |
| BM | $3.19 \pm 0.05$ | $2.53 \pm 0.02$ | $1.17 \pm 0.02$ | $1.28 \pm 0.02$ | $1.24 \pm 0.01$ |
| BNDL | $\mathbf{3.91 \pm 0.06}$ | $\mathbf{3.43 \pm 0.03}$ | $\mathbf{2.77 \pm 0.03}$ | $\mathbf{1.69 \pm 0.03}$ | $\mathbf{1.69 \pm 0.02}$ |

**Disentangled Measurement** Disentanglement Result on BNDL, BM and ResNet18. It can be observed that BNDL consistently achieves the best disentanglement metrics, aligning with the conclusions drawn in Section 5.2 of the paper. This suggests that BNDL has successfully learned more identifiable features through the constraint of sparsity.

**Additional Visualization Results** We visualized the feature representation $\boldsymbol{\theta}$ of BNDL and the baseline model (ResNet-50) for the same images in ImageNet-1k, as illustrated in Figures 7 to 11. Specifically, we selected the feature $\boldsymbol{\theta}$ with the highest activation for each image and applied the LIME method using the top-10 super-pixels for visualization, in line with the approach used in Figure 2. We have added comparative visualization results of BNDL and several models on ImageNet-1k, including the BLL model BM (Joo et al., 2020) and the sparse decision-layer model Debuggable Networks (Wong et al., 2021), the result is shown in Fig.6. The top row of Figures shows the true categories of the corresponding images, the second row presents our visualization results, and the third row displays the visualization results of the baseline model. Overall, the visualization results of BNDL are more semantically meaningful compared to those of ResNet-50.

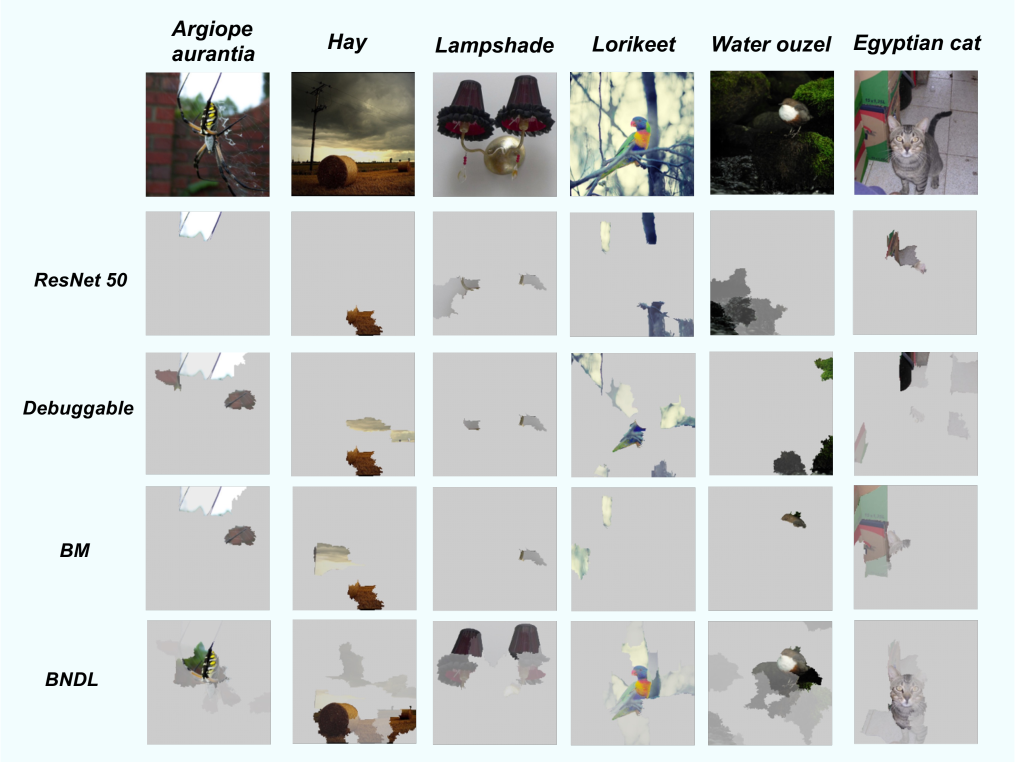


Figure 6: The LIME visualization results for BNDL ,ResNet-50 Debuggable Networks and BM, focusing on the largest $\theta$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels. This suggests that BNDL has learned more identifiable features.

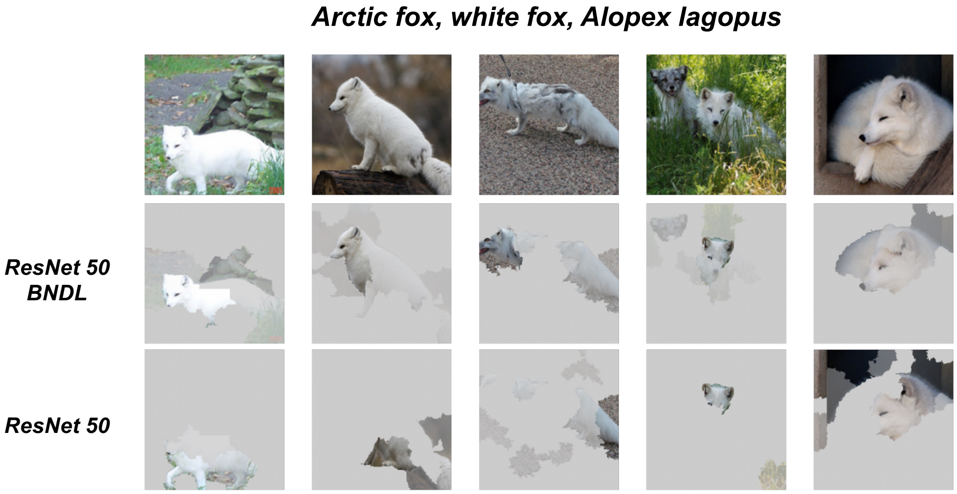


Figure 7: The LIME visualization results for BNDL and ResNet-50, focusing on the largest $\theta$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels compared to ResNet-50. This suggests that BNDL has learned more identifiable features.

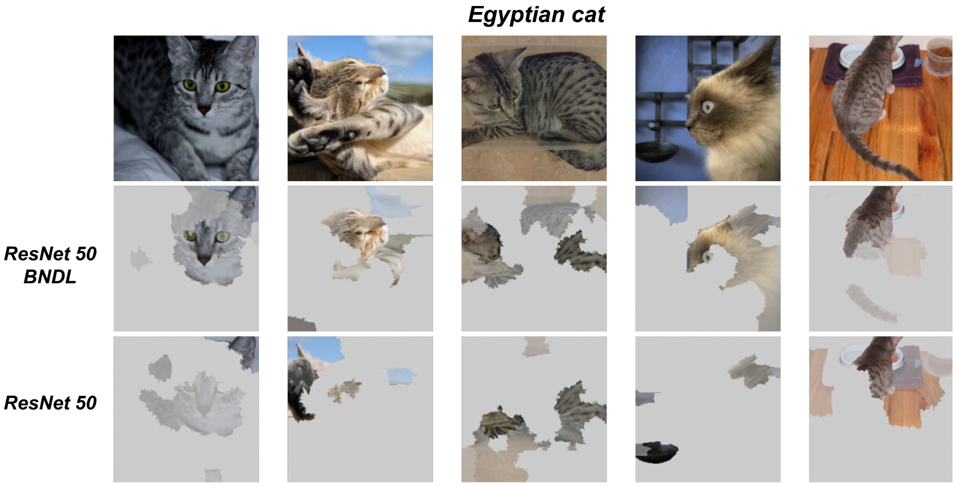


Figure 8: The LIME visualization results for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels compared to ResNet-50. This suggests that BNDL has learned more identifiable features.

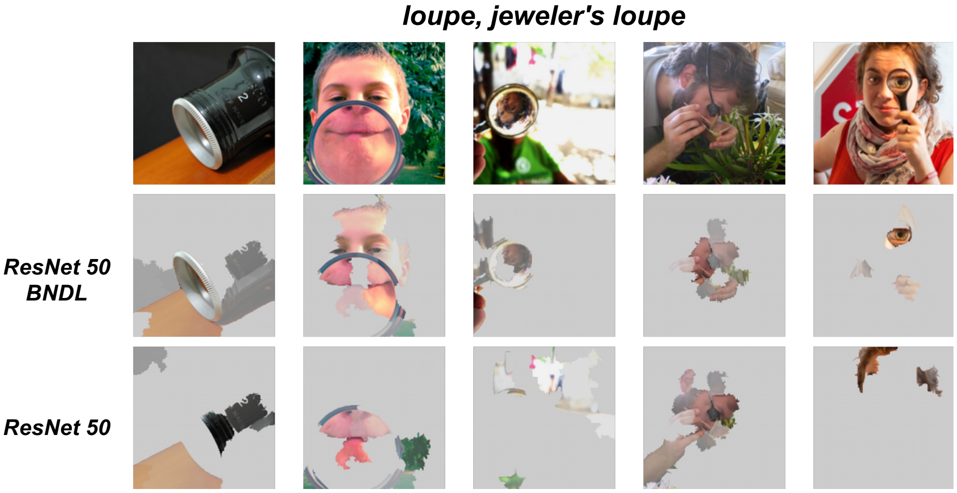


Figure 9: The LIME visualization results for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels compared to ResNet-50. This suggests that BNDL has learned more identifiable features.

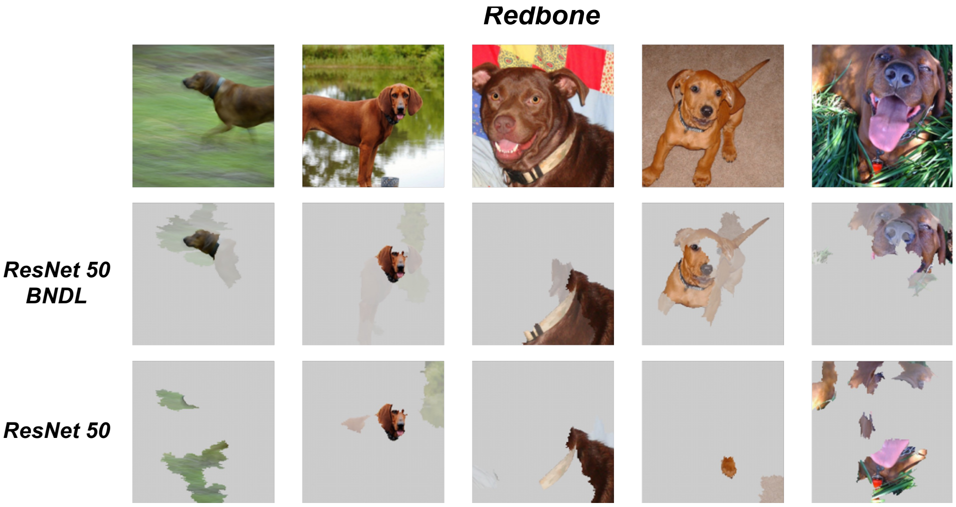


Figure 10: The LIME visualization results for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels compared to ResNet-50. This suggests that BNDL has learned more identifiable features.

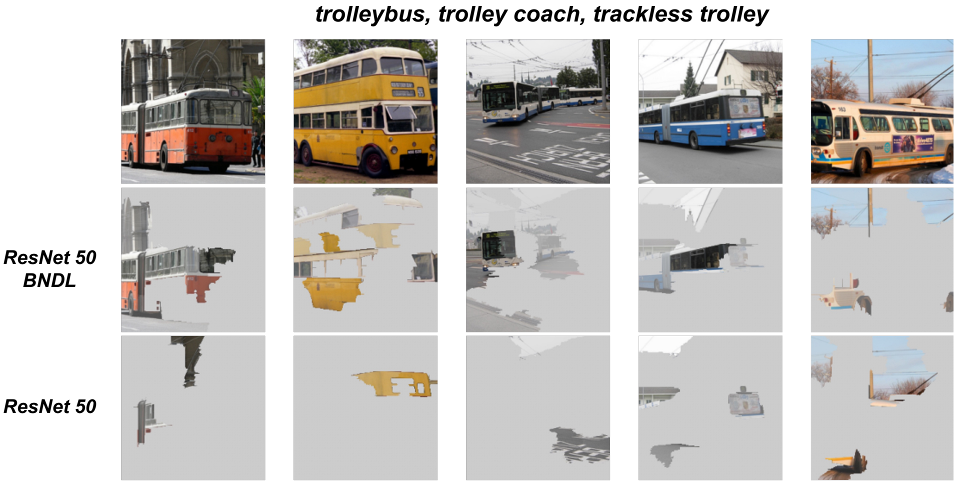


Figure 11: The LIME visualization results for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, demonstrate that BNDL's feature visualization aligns more closely with the semantic meaning of the true labels compared to ResNet-50. This suggests that BNDL has learned more identifiable features.