Deep Hybrid Models for Out-of-Distribution Detection Supplementary Material

A

A.1 Derivation of Equation (1)

With the assumption that $\mathbf{x}$ and θ are independent, the posterior joint distribution $p(y, \mathbf{x}, \theta|D)$ can be factorized as follows:

$$\begin{aligned} p(y, \mathbf{x}, \theta|D) &= p(y, \mathbf{x}|\theta, D)p(\theta|D) \\ &= p(y|\mathbf{x}, \theta)p(\mathbf{x}|\theta, D)p(\theta|D) \\ &= \underbrace{p(y|\mathbf{x}, \theta)}_{\text{data}} \underbrace{p(\mathbf{x}|D)}_{\text{distributional}} \underbrace{p(\theta|D)}_{\text{model}} \text{ ($\mathbf{x}$ and θ are independent)} \end{aligned}$$

Note that, when constructing a hybrid model, $p(\mathbf{x}|D)$ is generally modeled by $p(\mathbf{x}|\theta_2)$ parameterized by θ_2. Therefore, the hybrid model is essentially $p(y, \mathbf{x}|\theta) = p(y|\mathbf{x}, \theta)p(\mathbf{x}|\theta) = p(y|\mathbf{x}, \theta_1)p(\mathbf{x}|\theta_2)$ where $\theta = (\theta_1, \theta_2)$.

A.2 Proof of Proposition 3.3

The first half of the theorem is Theorem 3.1.6 of [1]. Thus, we only provide the proof for the second half of the theorem: *if a function ϕ is L-bi-Lipschitz continuous, then the singular values of its Jacobian lie in the interval (L^{-1}, L).*

Proof. Consider two metric spaces $(X, \|.\|_X)$ and $(H, \|.\|_H)$. A function $\phi : X \to H$ is L-bi-Lipschitz continuous, which means $\frac{1}{L} * \|\mathbf{x}_1 - \mathbf{x}_2\|_X \leq \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|_H \leq L * \|\mathbf{x}_1 - \mathbf{x}_2\|_X$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in X$. For simplicity, we use l^2-norm $\|.\|_2$ for both X and H.

Let $\mathbf{x}_1 = \mathbf{x}_2 + t * \mathbf{s}$ where $t \in R$, $\mathbf{s} \in X$, and $\|\mathbf{s}\|_2 = 1$; then we have

$$\frac{1}{L} \leq \frac{\|\phi(\mathbf{x}_2 + t * \mathbf{s}) - \phi(\mathbf{x}_2)\|_2}{t} \leq L$$

When $t \to 0$, we have

$$\frac{1}{L} \leq \|\frac{\partial \phi(\mathbf{x}_2)}{\partial \mathbf{x}}\mathbf{s}\|_2 \leq L \tag{6}$$

Let $J_\phi(\mathbf{x}) = \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}}$, since (6) is true for all $\mathbf{x}_2 \in X$, we have

$$\frac{1}{L} \leq \inf_{\mathbf{x} \in X, \|\mathbf{s}\|_2 = 1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2 \leq \|J_\phi(\mathbf{x})\mathbf{s}\|_2 \leq \sup_{\mathbf{x} \in X, \|\mathbf{s}\|_2 = 1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2 \leq L \tag{7}$$

Now we show that $\sigma_1 = \inf_{\mathbf{x} \in X, \|\mathbf{s}\|_2 = 1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2$ and $\sigma_r = \sup_{\mathbf{x} \in X, \|\mathbf{s}\|_2 = 1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2$ where σ_1 is the smallest singular value of $J_\phi(\mathbf{x})$ and σ_r is its largest singular value.

Suppose that, for a fixed $\mathbf{x} \in X$, the singular value decomposition of $J_\phi(\mathbf{x}) = U(\mathbf{x})\Sigma(\mathbf{x})V^T(\mathbf{x})$ where $U^T(\mathbf{x})U(\mathbf{x}) = I$ and $V^T(\mathbf{x})V(\mathbf{x}) = I$, then we have

$$\sup_{\|\mathbf{s}\|_2 = 1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2 = \sup_{\|\mathbf{s}\|_2 = 1} \|U(\mathbf{x})\Sigma(\mathbf{x})V^T(\mathbf{x})\mathbf{s}\|_2 = \sup_{\|\mathbf{s}\|_2 = 1} \|\Sigma(\mathbf{x})V^T(\mathbf{x})\mathbf{s}\|_2$$

since $U(\mathbf{x})$ is unitary, that is, $\|U(\mathbf{x})\mathbf{s}_0\|_2^2 = \mathbf{s}_0^T U^T(\mathbf{x})U(\mathbf{x})\mathbf{s}_0 = \mathbf{s}_0^T\mathbf{s}_0 = \|\mathbf{s}_0\|_2^2$, for any $\mathbf{s}_0 \in X$.

Let $y = V^T(\mathbf{x})\mathbf{s}$, then we have

$$\sup_{\|\mathbf{s}\|_2=1} \|\Sigma(\mathbf{x})V^T(\mathbf{x})\mathbf{s}\|_2 = \sup_{\|y\|_2=1} \|\Sigma(\mathbf{x})y\|_2 \tag{8}$$

since $\|y\|_2 = \|V^T(\mathbf{x})\mathbf{s}\|_2 = \|\mathbf{s}\|_2 = 1$, as $V(\mathbf{x})$ is unitary.

Let $\Sigma(\mathbf{x}) = \text{diag}(\sigma_1(\mathbf{x}), ..., \sigma_r(\mathbf{x}))$ where $\sigma_1(\mathbf{x})$ is the smallest singular value and $\sigma_r(\mathbf{x})$ is the largest singular value. Next, we solve $\sup_{\|y\|_2=1}\|\Sigma(\mathbf{x})y\|_2^2$ using a Lagrange Multiplier.

$$L(y, \lambda) = \|\Sigma(\mathbf{x})y\|_2^2 - \lambda(\|y\|_2^2 - 1) = y^T\Sigma^T(\mathbf{x})\Sigma(\mathbf{x})y - \lambda(y^Ty - 1)$$

$$\frac{\partial L}{\partial y} = 2\Sigma^T(\mathbf{x})\Sigma(\mathbf{x})y - 2\lambda y = 2(\Sigma^T(\mathbf{x})\Sigma(\mathbf{x})y - \lambda y) = 0$$

Thus, the maximum (minimum) of $\|\Sigma(\mathbf{x})y\|_2^2$ subjected to $\|y\|_2 = 1$ is the largest (smallest) eigenvalue of $\Sigma^T(\mathbf{x})\Sigma(\mathbf{x})$ which is $\sigma_r^2(\mathbf{x})$ ($\sup_{\|y\|_2=1}\|\Sigma(\mathbf{x})y\|_2^2 = \sigma_r^2(\mathbf{x})$). Thus, $\sup_{\|\mathbf{s}\|_2=1}\|J_\phi(\mathbf{x})\mathbf{s}\|_2 = \sup_{\|y\|_2=1}\|\Sigma(\mathbf{x})y\|_2 = \sigma_r(\mathbf{x})$.

Similarly (replacing sup with inf), we can also prove that $\inf_{\|\mathbf{s}\|_2=1}\|J_\phi(\mathbf{x})\mathbf{s}\|_2 = \sigma_1(\mathbf{x})$. Now let $\mathbf{x}$ vary; then we have

$$\sup_{\mathbf{x}\in X, \|\mathbf{s}\|_2=1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2 = \sup_{\mathbf{x}\in X} \sigma_r(\mathbf{x}) = \sigma_r$$

$$\inf_{\mathbf{x}\in X, \|\mathbf{s}\|_2=1} \|J_\phi(\mathbf{x})\mathbf{s}\|_2 = \inf_{\mathbf{x}\in X} \sigma_1(\mathbf{x}) = \sigma_1$$

From Equation (7), we have

$$\frac{1}{L} \le \sigma_1 \le ... \le \sigma_r \le L \tag{9}$$

□

A.3 Proof of Theorem 3.5

Proof. Consider two measure spaces $(X, \mathscr{A}, vol_X)$ and $(H, \mathscr{H}, vol_H)$ where $X = R^n$ and $H = R^m$ with $m < n$. Suppose that function $\phi : R^n \to R^m$ is $L^{1/m}$-bi-Lipschitz continuous ($L > 1$), and under mild conditions, its Jacobian $J_\phi(\mathbf{x})$ is full rank. From Proposition 3.3, the singular values of $J_\phi(\mathbf{x})$ are double-bounded by $L^{1/m}$ and $L^{-1/m}$:

$$\frac{1}{L^{1/m}} \le \sigma_1 \le ... \le \sigma_m \le L^{1/m} \tag{10}$$

For any measurable (open) set $B \in \mathscr{A}$, let $B = \cup B_\mathbf{x}$ where $B_\mathbf{x}$ are "almost" disjoint closed cubes, in the sense that only the boundaries of the cubes can overlap. Each $B_\mathbf{x}$ is an infinitesimal open cover of point $\mathbf{x} \in X$. From Equation (5) we see that, for each $B_\mathbf{x}$, the volume ratio $vol_H(\phi(B_\mathbf{x}))/vol_X(B_\mathbf{x}) = dV_\phi(\mathbf{h})/d\mathbf{x} = volJ_\phi = \prod_{i=1}^m \sigma_i$. Since each $\sigma_i (i = 1, ..., m)$ is double-bounded by $L^{1/m}$ and $L^{-1/m}$, their product $\prod_{i=1}^m \sigma_i$ is double-bounded by L and L^{-1}. Thus, we have

$$\frac{1}{L} \le \frac{vol_H(\phi(B_\mathbf{x}))}{vol_X(B_\mathbf{x})} = \prod_{i=1}^m \sigma_i \le L, \forall \mathbf{x} \in X \tag{11}$$

We notice that $\phi(B_\mathbf{x})$ are also disjoint open sets since ϕ is bijective. Thus, we have $vol_X(B) = \sum vol_X(B_\mathbf{x})$ and $vol_H(\phi(B)) = \sum vol_H(\phi(B_\mathbf{x}))$. From (11) and Lemma 1, we have

$$\frac{1}{L} \le \min \frac{vol_H(\phi(B_\mathbf{x}))}{vol_X(B_\mathbf{x})} \le \frac{vol_H(\phi(B))}{vol_X(B)} = \frac{\sum vol_H(\phi(B_\mathbf{x}))}{\sum vol_X(B_\mathbf{x})} \le \max \frac{vol_H(\phi(B_\mathbf{x}))}{vol_X(B_\mathbf{x})} \le L \tag{12}$$

Thus, we have $\frac{1}{L} * vol_X(B) \le vol_H(\phi(B)) \le L * vol_X(B), \forall B \in \mathscr{A}$. □

Lemma 1 Given two sequences of positive numbers $a_1, ..., a_k$ and $b_1, ..., b_k$,

$$\min_i \frac{a_i}{b_i} \leq \frac{\sum_i a_i}{\sum_i b_i} \leq \max_i \frac{a_i}{b_i}$$

Proof. Let $M = \max_i a_i/b_i$. This means that for all i, $a_i/b_i \leq M$, and thus $a_i \leq Mb_i$. Thus, we have

$$\sum_i a_i \leq \sum_i Mb_i \leq M \sum_i b_i$$

which is the same as

$$\frac{\sum_i a_i}{\sum_i b_i} \leq M = \max_i \frac{a_i}{b_i}$$

The other side of the inequality can be obtained similarly. □

A.4 Proof of Proposition 3.6

Proof. Consider a residual DNN $\phi = \phi_d \circ ... \circ \phi_2 \circ \phi_1$ where $\phi_l(\mathbf{x}) = \mathbf{x} + g_l(\mathbf{x})$ for $l = 1, \ldots, d$. For simplicity, we use l^2-norm $\|.\|_2$ for both X and H. All $g_l(\mathbf{x})$ are β-Lipschitz continuous where $0 < \beta < 1$. Thus, we have $\|g_l(\mathbf{x}_1) - g_l(\mathbf{x}_2)\|_2 \leq \beta * \|\mathbf{x}_1 - \mathbf{x}_2\|_2, \forall \mathbf{x}_1, \mathbf{x}_2 \in X$.

We first show that, for all $l = 1, \ldots, d$:

$$(1 - \beta) * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2 \leq (1 + \beta) * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{13}$$

We first show the left hand side of (13):

$$\begin{aligned}
\|\mathbf{x}_1 - \mathbf{x}_2\|_2 &= \|\mathbf{x}_1 - \mathbf{x}_2 + (\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)) - (\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2))\|_2 \\
&\leq \|(\phi_l(\mathbf{x}_2)) - \mathbf{x}_2) - (\phi_l(\mathbf{x}_1)) - \mathbf{x}_1)\|_2 + \|(\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2 \\
&\leq \|(g_l(\mathbf{x}_2) - g_l(\mathbf{x}_1)\|_2 + \|(\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2 \\
&\leq \beta * \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \|(\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2
\end{aligned}$$

Thus, we obtain:

$$(1 - \beta) * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2 \tag{14}$$

Next, we show the right hand side of (13):

$$\begin{aligned}
\|\phi_l(\mathbf{x}_1) - \phi_l(\mathbf{x}_2)\|_2 &= \|\mathbf{x}_1 + g_l(\mathbf{x}_2) - (\mathbf{x}_2 + g_l(\mathbf{x}_1))\|_2 \\
&\leq \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \|g_l(\mathbf{x}_1) - g_l(\mathbf{x}_2)\|_2 \\
&\leq (1 + \beta) * \|\mathbf{x}_1 - \mathbf{x}_2\|_2
\end{aligned} \tag{15}$$

Combining (14) and (15), we prove (13).

Since $\phi = \phi_d \circ ... \circ \phi_2 \circ \phi_1$, we have

$$(1 - \beta)^d * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|_2 \leq (1 + \beta)^d * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{16}$$

Let $L = max\{(1 - \beta)^{-d}, (1 + \beta)^d\}$, we have

$$\frac{1}{L} * \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|_2 \leq L * \|\mathbf{x}_1 - \mathbf{x}_2\|_2$$

□

A.5 Proof of Corollary 3.7

Proof. This proof is informal. We only provide an approximate analysis of the true bi-Lipschitz bound for residual DNNs in the real world. This can only be done under some mild assumptions. To start with, it is helpful to consider an equivalent definition of the bi-Lipschitz continuous condition in our following analysis. In Definition 3.2 of our paper, let $\|\mathbf{x}_1 - \mathbf{x}_2\|_X \to 0$, we get the following conclusion: a function $\phi : X \to H$ is L-bi-Lipschitz continuous if and only if there exists a constant $L \geq 1$, s.t. the local distance ratio $\frac{\|d\phi(\mathbf{x})\|}{\|d\mathbf{x}\|}$ (norm of the local derivative) lies in the interval $(\frac{1}{L}, L)$ for any $\mathbf{x} \in X$.

Suppose that our model is a DNN ϕ composed of d residual blocks, each of which is β-Lipschitz continuous where $0 < \beta < 1$. Following Proposition 3.6, ϕ is guaranteed to be at least L-bi-Lipschitz continuous where $L = max\{(1-\beta)^{-d}, (1+\beta)^d\}$. Without loss of generality, let $L = (1+\beta)^d$. In practice, however, it is possible to obtain a much tighter bi-Lipschitz bound (much smaller than the value of $L = (1+\beta)^d$) by considering the entire DNN as a whole rather than each layer in isolation. Specifically, in order to reach the bi-Lipschitz constant $L = (1+\beta)^d$, all d residual blocks must reach the bi-Lipschitz constant $(1+\beta)$ at the same time, which means that all local distance ratios $\frac{\|d\phi(\mathbf{x})\|}{\|d\mathbf{x}\|}$ of each block reach their maximum values $(1+\beta)$ at the same location $\mathbf{x}$. For a flexible DNN, this is not likely to happen in practice. What happens, in general, is that all local distance ratios $\frac{\|d\phi(\mathbf{x})\|}{\|d\mathbf{x}\|}$ of the blocks approximately lie uniformly in the interval $(\frac{1}{1+\beta}, 1+\beta)$ at a particular point $\mathbf{x}$.

To be more specific, let $\phi = \phi_d \circ ... \circ \phi_2 \circ \phi_1$ and $h_i = \phi_i \circ ... \circ \phi_2 \circ \phi_1$, where $\phi_i(\mathbf{x}) = \mathbf{x} + g_i(\mathbf{x})$ is the i-th residual block and $i = 1, \ldots, d$. Since each g_i is β-Lipschitz continuous, each ϕ_i is $(1+\beta)$-bi-Lipschitz continuous, and thus the distance ratio $\frac{\|d\phi_i(h_{i-1}(\mathbf{x}))\|}{\|dh_{i-1}(\mathbf{x})\|}$ lies in the interval $(\frac{1}{1+\beta}, 1+\beta)$. For simplicity of analysis, we assume that all d distance ratios $\frac{\|d\phi_i(h_{i-1}(\mathbf{x}))\|}{\|dh_{i-1}(\mathbf{x})\|}$ (at a particular point $\mathbf{x}$) of the d residual blocks are approximately statistically independent and all lie uniformly in the interval $(\frac{1}{1+\beta}, 1+\beta)$. This assumption is generally valid for the following reason: DNNs are universal approximators. Thus, each layer's shape should be diverse in order to be flexible enough to approximate complicated functions. Therefore, the norms of the local derivatives (distance ratios) of each layer should be diverse as well (after the DNN ϕ is trained to approximate a highly complex function). Therefore, it is reasonable to assume that the distance ratios are approximately statistically independent (in other words, the distance ratios should not be the same or even close at one particular location $\mathbf{x}$). Note that statistical independence is different from independence (e.g., pseudo-random number generators).

Suppose, for each i, that $\frac{\|d\phi_i(h_{i-1}(\mathbf{x}))\|}{\|dh_{i-1}(\mathbf{x})\|}$ reaches its maximum value $(1+\beta)$ at $\mathbf{x} = x_i^*$, the probability that $x_1^* = x_2^* = ... = x_d^*$ approaches to zero when d is large. Therefore, it is not very likely that the true bi-Lipschitz constant of ϕ reaches the value $L = (1+\beta)^d$. Actually, the expectation of the true bi-Lipschitz constant L should be $E[L] = 0.5^d(1+\beta+\frac{1}{1+\beta})^d$ with standard deviation $D[L] = \sqrt{\frac{1}{12}^d(1+\beta-\frac{1}{1+\beta})^{2d}}$ (d independent uniform distributions). Furthermore, following Theorem 3.5, if the data $\mathbf{x}$ takes values on an m-manifold, we would expect our model ϕ to be $0.5^{md}(1+\beta+\frac{1}{1+\beta})^{md}$-measure-preserving (after it is trained on $\mathbf{x}$).

□

A.6 More Related Work

Uncertainty Factorization In the literature of uncertainty factorization, the main goal is to factorize the model into two or three types of uncertainty so that we can directly access the uncertainty information once we learn the model. A good uncertainty factorization should be general (widely useful in various models) and semantically accurate (the results should be consistent with the definitions presented in Section 2.1 in our paper).

The existing literature [2–4] focuses on factorizing the posterior predictive distribution $p(y|\mathbf{x}, \theta, D)$. One shortcoming of this approach is that the proposed factorizations are often complicated and not general, only suitable for specific models. Furthermore, many factorizations fail to capture the semantics of different sources of uncertainty correctly. For example, [2] proposes to factorize the posterior predictive distribution as $P(\omega_c|x^*, D) = \int\int p(\omega_c|\mu)p(\mu|x^*, \theta)p(\theta|D)d\mu\, d\theta$ where μ is an auxiliary variable the authors introduce. According to the paper, $p(\omega_c|\mu)$ is the data uncertainty, $p(\mu|x^*, \theta)$ is the distributional uncertainty, and $p(\theta|D)$ is the model uncertainty. There are two shortcomings of this factorization. (1) It is not general since the authors introduce a new variable μ, which means that if we want to model the distributional uncertainty or the model uncertainty, we have to incorporate the variable μ into our model design. (2) It is not semantically accurate. For example, the distributional uncertainty defined by $p(\mu|x^*, \theta)$ does not capture the discrepancy between x^* and D. The data uncertainty defined by $p(\omega_c|\mu)$ is solely determined by a model variable μ. Thus, it cannot capture the inherent randomness of the data when x varies. The uncertainty factorization introduced by [4] is not general since it is only applicable in their proposed Bayesian Nonparametric

Ensemble (BNE) models. The uncertainty factorization proposed by [3] is also not general. It only works with Bayesian Neural Networks with latent variables (BNN+LV), where latent variables z come from Gaussian distributions. Furthermore, this uncertainty factorization is very complicated.

To our knowledge, this is the first work to attempt to factorize the posterior joint distribution $p(y|\mathbf{x}, \theta, D)$. Our uncertainty factorization is quite neat while remaining semantically accurate. It is also quite general since it does not introduce auxiliary variables or assume a specific form of the model. Therefore, this uncertainty factorization can be used in a wide variety of models.

B Experiment Details

Training In all the vision experiments, the initial learning rate is set to 0.05, which drops by 0.2 at 60, 120, and 160 epochs. We use SGD with Nesterov momentum, and the batch size is set to 64, momentum to 0.9, and weight decay to 5e-4. We train all the models for 200 epochs. We only apply the standard data augmentation (horizontal flips and random cropping with 4x4 padding). Following [5], we set power iteration to 1 and SN upper bound c to 6. For NF, we use the same architecture as [6] except that the hidden dimension of the residual blocks is set to 640 rather than 256. We find that a larger weight decay (16e-4) for the NF weights leads to better OoD detection performance.

In the text experiments, we pre-tokenize the sentences using the standard XLNet tokenizer[1] with a maximum sequence length of 32 and initialize the model from the official XLNetBase checkpoint[2]. For fine-tuning, we use AdamW [7] optimizer with weight decay rate 1e-6, and the batch size is set to 256. The initial learning rate is set to 3e-6, which drops by 0.5 at 60, 120, and 160 epochs. We train the models for 200 epochs. For SN, following [5], we set power iteration to 1, and SN upper bound c to 0.95, and only apply it to the pooler dense layer of the classification token. For NF, the hidden dimension of the residual blocks is set to 768.

All models are implemented in Pytorch and are trained on a V-100 GPU.

Evaluation Metrics For the vanilla DNN, Deep Ensembles, and SNGP, we compute their OoD uncertainty scores using the maximum value of the logits. We compute the kernel distance for DUQ and NF probability for DHM, respectively. To evaluate the model's calibration performance on ID data, we use the empirical estimate of Expected Calibration Error $ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} \big|acc(B_m) - conf(B_m)\big|$ [8] where M is set to 15 in this paper, reflecting the difference in the model's predictive accuracy and its confidence. To evaluate the model's OoD detection performance, we use Area Under Receiver Operating Characteristic (AUROC) and Area Under Precision-Recall (AUPR). A ROC curve is plotted with the TPR against the FPR, while a PR curve plots the relationship between precision and recall. Each point on a ROC curve or a PR curve represents one possible classifier (threshold). The AUROC and AUPR are holistic metrics that summarize the performance of detectors at all possible thresholds. The value of AUROC and AUPR vary between 0 and 1, with an uninformative detector yielding 0.5. An excellent OoD detector has an AUROC or AUPR value near 1, which means that the model is well capable of distinguishing between ID and OoD classes.

Assets The datasets used in this paper include CIFAR-10/100 [9], SVHN [10], TinyImageNet[3] [11], LSUN [12], iSUN [13], and CLINC150[4] [14]. They are all open source available and widely used in various domains and, to our knowledge, do not contain personally identifiable information or offensive content. We use the official implementations of DUQ [5] [15] and SNGP [6] [5] in our experiments. Furthermore, our implementation of DHM adapts code from [16] for the NF [7].

C Additional Experiment Results

Additional results are shown in Table 1-3. The results are consistent with the previous experiments.

[1] `https://huggingface.co/transformers/model_doc/xlnet.html` (Apache License, Version 2.0)

[2] `https://huggingface.co/transformers/model_doc/xlnet.html` (Apache License, Version 2.0)

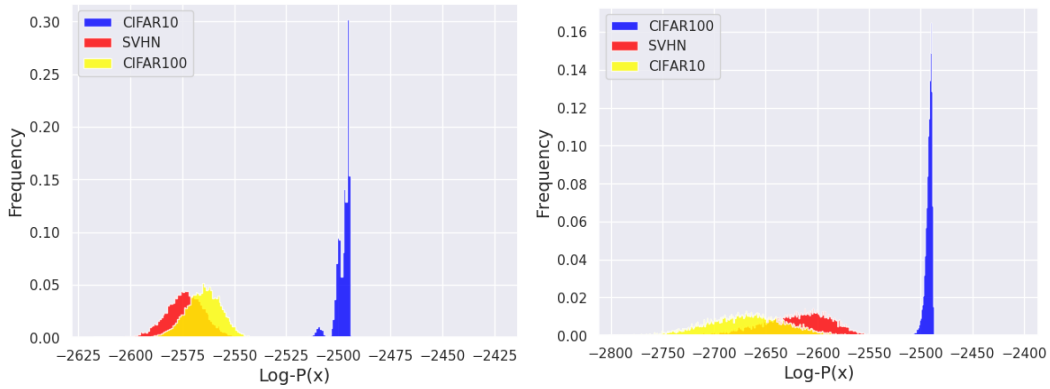
[3] `https://github.com/rmccorm4/Tiny-Imagenet-200` (MIT License)

[4] `https://github.com/clinc/oos-eval` (Creative Commons Public License)

[5] `https://github.com/y0ast/deterministic-uncertainty-quantification` (MIT License)

[6] `https://github.com/google/uncertainty-baselines/tree/master/baselines` (Apache License, Version 2.0)

[7] `https://github.com/rtqichen/residual-flows` (MIT License)



(a) CIFAR-10 (b) CIFAR-100

Figure 1: (a) and (b) show the histograms of $\log p(\mathbf{x})$ for both ID and OoD datasets.

	In-distribution			OoD: SVHN		OoD: CIFAR-100	
	Accuracy (↑)	ECE (↓)	NLL (↓)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)
Deterministic	94.6 ± 0.01	0.033 ± 0.002	0.214 ± 0.02	0.921 ± 0.01	0.918 ± 0.01	0.863 ± 0.01	0.816 ± 0.01
Deep Ensembles	**95.6 ± 0.01**	**0.020 ± 0.001**	**0.178 ± 0.01**	0.952 ± 0.01	0.948 ± 0.01	0.911 ± 0.01	0.902 ± 0.01
DUQ	94.6 ± 0.01	0.032 ± 0.002	0.203 ± 0.02	0.950 ± 0.01	0.945 ± 0.01	0.884 ± 0.01	0.879 ± 0.01
SNGP	94.5 ± 0.01	0.025 ± 0.002	0.215 ± 0.02	0.961 ± 0.01	0.957 ± 0.01	0.907 ± 0.01	0.895 ± 0.01
DNN+SN	94.6 ± 0.01	0.033 ± 0.002	0.214 ± 0.02	0.933 ± 0.01	0.925 ± 0.01	0.865 ± 0.01	0.827 ± 0.01
DNN+NF	94.8 ± 0.01	0.030 ± 0.002	0.195 ± 0.02	0.995 ± 0.01	0.989 ± 0.02	0.941 ± 0.03	0.924 ± 0.04
DHM (Ours)	95.3 ± 0.01	0.028 ± 0.002	0.183 ± 0.02	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**

Table 1: Results for ResNet-18 on CIFAR-10, averaged over 10 independent seeds.

	In-distribution			OoD: SVHN		OoD: CIFAR-10	
	Accuracy (↑)	ECE (↓)	NLL (↓)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)
Deterministic	77.5 ± 0.01	0.052 ± 0.002	0.901 ± 0.02	0.896 ± 0.01	0.915 ± 0.01	0.794 ± 0.01	0.814 ± 0.01
Deep Ensembles	**79.3 ± 0.01**	**0.030 ± 0.001**	**0.754 ± 0.01**	0.932 ± 0.01	0.938 ± 0.01	0.865 ± 0.01	0.883 ± 0.01
DUQ	77.6 ± 0.01	0.051 ± 0.002	0.898 ± 0.02	0.891 ± 0.01	0.899 ± 0.01	0.843 ± 0.01	0.839 ± 0.01
SNGP	77.5 ± 0.01	0.034 ± 0.002	0.885 ± 0.01	0.901 ± 0.01	0.907 ± 0.01	0.856 ± 0.01	0.867 ± 0.01
DNN+SN	77.5 ± 0.01	0.053 ± 0.003	0.902 ± 0.02	0.895 ± 0.01	0.912 ± 0.01	0.804 ± 0.01	0.817 ± 0.01
DNN+NF	78.0 ± 0.01	0.055 ± 0.003	0.877 ± 0.02	0.976 ± 0.02	0.975 ± 0.02	0.932 ± 0.04	0.921 ± 0.04
DHM (Ours)	78.3 ± 0.01	0.055 ± 0.003	0.876 ± 0.02	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**

Table 2: Results for ResNet-18 on CIFAR-100, averaged over 10 independent seeds.

	OoD: TinyImageNet		OoD: LSUN		OoD: iSUN	
	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)
Deterministic	0.965 ± 0.01	0.972 ± 0.01	0.973 ± 0.01	0.975 ± 0.01	0.956 ± 0.01	0.963 ± 0.01
Deep Ensembles	0.992 ± 0.01	0.993 ± 0.01	0.998 ± 0.01	0.998 ± 0.01	0.991 ± 0.01	0.990 ± 0.01
DUQ	0.988 ± 0.01	0.987 ± 0.01	0.994 ± 0.01	0.995 ± 0.01	0.986 ± 0.01	0.986 ± 0.01
SNGP	0.997 ± 0.01	0.997 ± 0.01	0.998 ± 0.01	0.998 ± 0.01	0.995 ± 0.01	0.996 ± 0.01
DNN+SN	0.966 ± 0.01	0.969 ± 0.01	0.975 ± 0.01	0.981 ± 0.01	0.961 ± 0.01	0.958 ± 0.01
DNN+NF	0.997 ± 0.02	0.997 ± 0.02	0.999 ± 0.02	0.999 ± 0.02	0.998 ± 0.02	0.998 ± 0.02
DHM (Ours)	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**

Table 3: Additional experiment results for Wide ResNet-28-10 on CIFAR-10, averaged over 10 independent seeds.

References

[1] Herbert Federer. Geometric measure theory. In *Classics in Mathematics. Springer-Verlag Berlin Heidelberg*, 1969.

[2] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Neural Information Processing Systems*, 2018.

[3] Stefan Depeweg, Jose Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning.

In *ICML*, 2018.

[4] Jeremiah Liu, John Paisley, Marianthi-Anna Kioumourtzoglou, and Brent Coull. Accurate uncertainty estimation and decomposition in ensemble learning. In *NeurIPS*, 2019.

[5] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Zhang Hongjie, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, 2020.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[10] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[11] `https://tiny-imagenet.herokuapp.com/`.

[12] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong . Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. In *arXiv:1506.03365, 2015.*, 2019.

[13] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. In *arXiv:1504.06755, 2015.*, 2019.

[14] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. . Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *arXiv: 1909.02027.*, 2019.

[15] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[16] Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, 2019.