

---

# Active Learning in Bayesian Neural Networks: Balanced Entropy Learning Principle - Supplementary material

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Appendix

### 2 A.1 Derivation of marginalized joint entropy

3 The Janossy density function resides in a combination of continuous and discrete domains [8]. For  
4 the Janossy density of  $(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega))$  on  $\Delta^C \times [C]$ , we may follow the classical approach:

$$\begin{aligned} & \mathbb{P}(P_1 \in [p_1 + dp_1], \dots, P_C \in [p_C + dp_C], Y = i) \\ & \approx \mathbb{P}(Y = i | P_1 = p_1, \dots, P_C = p_C) \mathbb{P}(P_1 \in [p_1 + dp_1], \dots, P_C \in [p_C + dp_C]) \\ & \approx p_i f(p_1, \dots, p_C) dp_1 \dots dp_C, \end{aligned} \quad (1)$$

5 where  $f(\cdot)$  is a density function of  $\Phi(\mathbf{x}, \omega)$ . So we may write the Janossy density of  
6  $(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega))$  as follows:

$$j(p_1, \dots, p_C, y = i) = p_i f(p_1, \dots, p_C). \quad (2)$$

7 Following the point process entropy [29, 17, 30, 8], the joint entropy of  $\Phi(\mathbf{x}, \omega)$  and  $Y(\mathbf{x}, \omega)$  can be  
8 defined as

$$\mathfrak{H}(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega)) = - \sum_{i=1}^C \int_{\Delta^c} j(p_1, \dots, p_C, y = i) \log j(p_1, \dots, p_C, y = i) dp_1 \dots dp_C. \quad (3)$$

9 We note that

$$\int_{\Delta^c} p_i f(p_1, \dots, p_C) dp_1 \dots dp_C = \int_{[0,1]} p_i f(p_i) dp_i = \mathbb{E}P_i. \quad (4)$$

10 We may split the Janossy density into two pieces:

$$j(p_1, \dots, p_C, y = i) = (\mathbb{E}P_i) \left( \frac{p_i}{\mathbb{E}P_i} f(p_1, \dots, p_C) \right). \quad (5)$$

11 Plugging (5) into (3), we have

$$\mathfrak{H}(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega)) = H(Y(\mathbf{x}, \omega)) + \mathbb{E}_Y [h(\Phi(\mathbf{x}, \omega) | Y)]. \quad (6)$$

12 On the other hand,

$$\begin{aligned}
(6) &= - \sum_{i=1}^C \int_{\Delta^c} j(p_1, \dots, p_C, y=i) \log j(p_1, \dots, p_C, y=i) dp_1 \dots dp_C \\
&= - \sum_{i=1}^C \int_{\Delta^c} (\mathbb{E}P_i) \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) \log (\mathbb{E}P_i) \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) dp_1 \dots dp_C \\
&= - \sum_{i=1}^C (\mathbb{E}P_i) \log (\mathbb{E}P_i) - \sum_{i=1}^C \int_{\Delta^c} (\mathbb{E}P_i) \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) \log \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) dp_1 \dots dp_C.
\end{aligned}
\tag{7}$$

We apply Jensen's inequality on the second term (by focusing on each summand). For each  $i \in \{1, \dots, C\}$ ,

$$\begin{aligned}
& - (\mathbb{E}P_i) \int_{\Delta^c} \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) \log \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) dp_1 \dots dp_C \\
&= - (\mathbb{E}P_i) \int_{p_i} \int_{\Delta^c \setminus \{p_i\}} \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) \log \left( \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) \right) dp_{1 \dots C}^{-i} dp_i \\
&\leq - (\mathbb{E}P_i) \int_{p_i} \left( \int_{\Delta^c \setminus \{p_i\}} \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) dp_{1 \dots C}^{-i} \right) \log \left( \int_{\Delta^c \setminus \{p_i\}} \frac{p_i}{\mathbb{E}P_i} p(p_1, \dots, p_C) dp_{1 \dots C}^{-i} \right) dp_i \\
&= - \int_{p_i} p_i f(p_i) \log \left( \frac{p_i}{\mathbb{E}P_i} f(p_i) \right) dp_i = - \mathbb{E}P_i \left[ P_i \log \left( \frac{P_i}{\mathbb{E}P_i} f(P_i) \right) \right],
\end{aligned}
\tag{8}$$

where  $dp_{1 \dots C}^{-i}$  indicates  $dp_1 \dots dp_C$  except  $dp_i$ . By combining all terms together, we have

$$(6) \leq - \sum_{i=1}^C (\mathbb{E}P_i) \log (\mathbb{E}P_i) - \sum_{i=1}^C \mathbb{E}P_i \left[ P_i \log \left( \frac{P_i}{\mathbb{E}P_i} f(P_i) \right) \right] = - \sum_i \mathbb{E}P_i [P_i \log (P_i f(P_i))].
\tag{9}$$

## A.2 Equivalent formulation of marginalized joint entropy

Let us assume that  $P_i \sim \text{Beta}(\alpha_i, \beta_i)$  and  $P_i^+ \sim \text{Beta}(\alpha_i + 1, \beta_i)$ .

$$\begin{aligned}
\text{MJEnt}[\mathbf{x}] &= - \sum_i \mathbb{E}P_i [P_i \log (P_i f(P_i))] = - \sum_i \int_0^1 p_i f(p_i) \log (p_i f(p_i)) dp_i \\
&= - \sum_i (\mathbb{E}P_i) \int_0^1 \frac{p_i f(p_i)}{\mathbb{E}P_i} \log \left( \frac{p_i f(p_i)}{\mathbb{E}P_i} \right) dp_i - \sum_i (\mathbb{E}P_i) \log (\mathbb{E}P_i)
\end{aligned}
\tag{10}$$

$$= \sum_i (\mathbb{E}P_i) [h(P_i^+) - \log (\mathbb{E}P_i)],
\tag{11}$$

where  $h(P_i^+)$  is the differential entropy of  $P_i^+$ .

## A.3 Proof of Theorem 3.1

For this appendix to be self-contained, we borrow the proof from [1] which presents the formula stated in Theorem A.1. But we are able to simplify the analytical formula so that it shows  $\text{DirichletBALD}[\mathbf{x}]$  is a function of marginal probabilities, which is one of the key observations in Section 3.4. Let  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_C)$  and  $\boldsymbol{\eta}(i, ++)= (\eta_1, \dots, \eta_{i-1}, \eta_i + 1, \eta_{i+1}, \dots, \eta_C)$ . Let  $B(\boldsymbol{\eta}) = \frac{\Gamma(\eta_1) \dots \Gamma(\eta_C)}{\Gamma(\sum_{k=1}^C \eta_k)}$ , and  $\Gamma(\cdot)$  is a Gamma function.

**Theorem A.1.** [1, Theorem III.1] Assume that  $\Phi(\mathbf{x}, \omega) := (P_1, \dots, P_C) \sim \text{Dirichlet}(\eta_1, \dots, \eta_C)$ . Then the mutual information  $\text{BALD}[\mathbf{x}]$  can be analytically calculated as follows.

$$\begin{aligned}
\text{DirichletBALD}[\mathbf{x}] &:= \left( \sum_{k=1}^C \eta_k - C \right) \Psi \left( \sum_{k=1}^C \eta_k \right) - \sum_{i=1}^C (\eta_i - 1) \Psi(\eta_i) - \sum_{i=1}^C \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \log \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \\
&+ \sum_{i=1}^C \sum_{j \neq i} \frac{(\eta_j - 1) B(\boldsymbol{\eta}(i, ++))}{B(\boldsymbol{\eta})} \left[ \Psi(\eta_j) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right] + \sum_{i=1}^C \frac{\eta_i B(\boldsymbol{\eta}(i, ++))}{B(\boldsymbol{\eta})} \left[ \Psi(\eta_i + 1) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right],
\end{aligned}$$

where  $\Psi(\cdot)$  is a Digamma function.

31 Note that the density function  $f(\cdot)$  of  $\text{Dirichlet}(\eta_1, \dots, \eta_C)$  is given by

$$f(\mathbf{p}) := f(p_1, \dots, p_C) = \frac{1}{B(\boldsymbol{\eta})} \prod_{i=1}^C p_i^{\eta_i-1}. \quad (12)$$

32 Then to derive the analytical form, we shall calculate each term as below [1]:

$$\text{DirichletBALD}[\mathbf{x}] = h(\Phi(\mathbf{x}, \omega)) + H(Y(\mathbf{x}, \omega)) - \mathfrak{H}(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega)). \quad (13)$$

33 Given  $\Phi(\mathbf{x}, \omega) := (P_1, \dots, P_C) \sim \text{Dirichlet}(\eta_1, \dots, \eta_C)$ , the first differential entropy of Dirichlet  
34 distribution is well-known [11, 27].

$$\begin{aligned} h(\Phi(\mathbf{x}, \omega)) &= - \int_{\Delta^C} f(\mathbf{p}) \log f(\mathbf{p}) d\mathbf{p} \\ &= \log B(\boldsymbol{\eta}) + \left( \sum_{k=1}^C \eta_k - C \right) \Psi \left( \sum_{k=1}^C \eta_k \right) - \sum_{i=1}^C (\eta_i - 1) \Psi(\eta_i). \end{aligned} \quad (14)$$

35 For the second entropy term, we first need to use a simple property of Dirichlet distribution.

$$\mathbb{E}P_i = \frac{\eta_i}{\sum_{k=1}^C \eta_k}. \quad (15)$$

36 Then the second term can be obtained by following the Shannon entropy with the equation (15).

$$H(Y(\mathbf{x}, \omega)) = - \sum_{i=1}^C \mathbb{E}P_i \log \mathbb{E}P_i = - \sum_{i=1}^C \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \log \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right). \quad (16)$$

37 For the third joint entropy term, we need to prove the following lemma.

38 **Lemma 1.** Assume that  $\Phi(\mathbf{x}, \omega) := (P_1, \dots, P_C) \sim \text{Dirichlet}(\eta_1, \dots, \eta_C)$ .

$$\mathbb{E}[P_i \log P_j] = \begin{cases} \frac{\eta_i}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_i + 1) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right] & \text{if } i = j, \\ \frac{\eta_i}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_j) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right] & \text{if } i \neq j. \end{cases}$$

39 To prove the Lemma 1, first we consider the  $i = j$  case.

$$\begin{aligned} \mathbb{E}[P_i \log P_i] &= \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} (p_i \log p_i) \prod_{k=1}^C p_k^{\eta_k-1} d\mathbf{p} = \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} p_i^{\eta_i} \log p_i \prod_{k \neq i} p_k^{\eta_k-1} d\mathbf{p} \\ &= \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} \frac{d}{d\eta_i} p_i^{\eta_i} \prod_{k \neq i} p_k^{\eta_k-1} d\mathbf{p} = \frac{1}{B(\boldsymbol{\eta})} \frac{d}{d\eta_i} \int_{\Delta^C} p_i^{\eta_i} \prod_{k \neq i} p_k^{\eta_k-1} d\mathbf{p} \\ &= \frac{1}{B(\boldsymbol{\eta})} \frac{d}{d\eta_i} B(\boldsymbol{\eta}(i, ++)) = \frac{B(\boldsymbol{\eta}(i, ++))}{B(\boldsymbol{\eta})} \left[ \Psi(\eta_i + 1) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right] \\ &= \frac{\eta_i}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_i + 1) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right]. \end{aligned}$$

40 Note that we may interchange the differentiation and the integral operator by applying Lebesgue's  
41 dominated convergence theorem [13]. The second last equality can be derived by the definition of the  
42 Digamma function [3]. Similarly, for the  $i \neq j$  case,

$$\begin{aligned} \mathbb{E}[P_i \log P_j] &= \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} (p_i \log p_j) \prod_{k=1}^C p_k^{\eta_k-1} d\mathbf{p} = \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} p_i^{\eta_i} p_j^{\eta_j-1} \log p_j \prod_{k \neq i, j} p_k^{\eta_k-1} d\mathbf{p} \\ &= \frac{1}{B(\boldsymbol{\eta})} \int_{\Delta^C} p_i^{\eta_i} \frac{d}{d\eta_j} p_j^{\eta_j-1} \prod_{k \neq i, j} p_k^{\eta_k-1} d\mathbf{p} = \frac{1}{B(\boldsymbol{\eta})} \frac{d}{d\eta_j} \int_{\Delta^C} p_i^{\eta_i} p_j^{\eta_j-1} \prod_{k \neq i, j} p_k^{\eta_k-1} d\mathbf{p} \\ &= \frac{1}{B(\boldsymbol{\eta})} \frac{d}{d\eta_j} B(\boldsymbol{\eta}(i, ++)) = \frac{B(\boldsymbol{\eta}(i, ++))}{B(\boldsymbol{\eta})} \left[ \Psi(\eta_j) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right] \\ &= \frac{\eta_i}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_j) - \Psi \left( \left( \sum_{k=1}^C \eta_k \right) + 1 \right) \right]. \end{aligned}$$

On the other hand, following the usual point process entropy calculation, we may write a Janossy density function of  $(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega))$  on  $\Delta^C \times [C]$  as follows:

$$j(\mathbf{p}, y = i) = p_i f(\mathbf{p}) = \frac{1}{B(\boldsymbol{\eta})} p_i^{\eta_i} \prod_{j \neq i} p_j^{\eta_j - 1}, \quad (17)$$

where  $\mathbf{p} := (p_1, \dots, p_C)$  and  $f(\cdot)$  is a density function of  $\Phi(\mathbf{x}, \omega)$ . Then the joint entropy of  $\Phi(\mathbf{x}, \omega)$  and  $Y(\mathbf{x}, \omega)$  can be defined as

$$\mathfrak{H}(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega)) = - \sum_{i=1}^C \int_{\Delta^C} j(\mathbf{p}, y = i) \log j(\mathbf{p}, y = i) d\mathbf{p}. \quad (18)$$

Then we have the following identity by plugging the Janossy density (17) of  $(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega))$  into the equation (18).

$$\begin{aligned} & \mathfrak{H}(\Phi(\mathbf{x}, \omega), Y(\mathbf{x}, \omega)) \\ &= (\log B(\boldsymbol{\eta})) \sum_{i=1}^C \mathbb{E}[P_i] - \sum_{i=1}^C \sum_{j \neq i} (\eta_j - 1) \mathbb{E}[P_i \log P_j] - \sum_{i=1}^C \eta_i \mathbb{E}[P_i \log P_i] =: (*). \end{aligned}$$

By applying Lemma 1, we have

$$\begin{aligned} (*) &= \log B(\boldsymbol{\eta}) - \sum_{i=1}^C \sum_{j \neq i} \frac{\eta_i(\eta_j - 1)}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_j) - \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \right] \\ &\quad - \sum_{i=1}^C \frac{\eta_i^2}{\sum_{k=1}^C \eta_k} \left[ \Psi(\eta_i + 1) - \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \right]. \end{aligned} \quad (19)$$

By combining three terms (14), (16), and (19) in the equation (13), we have the following simplified formula:

$$\begin{aligned} & \text{DirichletBALD}[\mathbf{x}] \\ &= \left( \sum_{k=1}^C \eta_k - C \right) \Psi\left(\sum_{k=1}^C \eta_k\right) - \sum_{i=1}^C (\eta_i - 1) \Psi(\eta_i) - \sum_{i=1}^C \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \log \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \\ &\quad + \sum_{i=1}^C \sum_{j \neq i} \left( \frac{\eta_i(\eta_j - 1)}{\sum_{k=1}^C \eta_k} \right) \left[ \Psi(\eta_j) - \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \right] \\ &\quad + \sum_{i=1}^C \left( \frac{\eta_i^2}{\sum_{k=1}^C \eta_k} \right) \left[ \Psi(\eta_i + 1) - \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \right] \\ &= \left( \sum_{k=1}^C \eta_k - C \right) \Psi\left(\sum_{k=1}^C \eta_k\right) - \sum_{i=1}^C \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \log \left( \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \\ &\quad - \sum_{i=1}^C \frac{\eta_i(\eta_i - 1)}{\sum_{k=1}^C \eta_k} \Psi(\eta_i) - \sum_{i=1}^C (\eta_i - 1) \left( 1 - \frac{\eta_i}{\sum_{k=1}^C \eta_k} \right) \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \\ &\quad + \sum_{i=1}^C \left( \frac{\eta_i^2}{\sum_{k=1}^C \eta_k} \right) \left[ \Psi(\eta_i + 1) - \Psi\left(\left(\sum_{k=1}^C \eta_k\right) + 1\right) \right]. \end{aligned}$$

Therefore DirichletBALD is a function of marginals of  $\Phi(\mathbf{x}, \omega)$  with Dirichlet distribution parameters  $\eta_i$  and  $\sum_{i=1}^C \eta_i$ . Under Beta marginal distribution assumption, by letting  $\eta_i = \alpha_i$  and  $\sum_{k=1}^C \eta_k = \alpha_i + \beta_i$  for any  $i$  since each marginal distribution of Dirichlet distribution follows Beta distribution, we have

$$\begin{aligned} \text{BetaMarginalBALD}[\mathbf{x}] &:= \sum_{i=1}^C (\alpha_i - 1) \Psi(\alpha_i + \beta_i) - \sum_{i=1}^C \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) \log \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) - \sum_{i=1}^C \frac{\alpha_i(\alpha_i - 1)}{\alpha_i + \beta_i} \Psi(\alpha_i) \\ &\quad - \sum_{i=1}^C \frac{\beta_i(\alpha_i - 1)}{\alpha_i + \beta_i} \Psi(\alpha_i + \beta_i + 1) + \sum_{i=1}^C \left( \frac{\alpha_i^2}{\alpha_i + \beta_i} \right) [\Psi(\alpha_i + 1) - \Psi(\alpha_i + \beta_i + 1)]. \end{aligned}$$

Therefore Theorem 3.1 follows.

On the other hand, since Beta marginal distributions are sufficient to calculate the mutual information, the same idea can be applied to the aleatoric uncertainty.

**Corollary 1.** *Under Beta marginal distribution approximation, let  $P_i \sim \text{Beta}(\alpha_i, \beta_i)$  in  $\Phi(\mathbf{x}, \omega)$ . Then the aleatoric uncertainty can be estimated as follows:*

$$\begin{aligned} \text{BetaMarginalAleatoricUncertainty}[\mathbf{x}] := & - \sum_{i=1}^C (\alpha_i - 1) \Psi(\alpha_i + \beta_i) + \sum_{i=1}^C \frac{\alpha_i (\alpha_i - 1)}{\alpha_i + \beta_i} \Psi(\alpha_i) \\ & + \sum_{i=1}^C \frac{\beta_i (\alpha_i - 1)}{\alpha_i + \beta_i} \Psi(\alpha_i + \beta_i + 1) - \sum_{i=1}^C \left( \frac{\alpha_i^2}{\alpha_i + \beta_i} \right) [\Psi(\alpha_i + 1) - \Psi(\alpha_i + \beta_i + 1)]. \end{aligned}$$

#### A.4 Proof of Theorem 4.1

First let a positive integer  $\Delta^{-1} > 0$  be given and let  $\Upsilon := \{I_n\}$ , a collection of evenly divided intervals in  $[0, 1]$  where  $I_n := [(n-1)\Delta, n\Delta]$  for  $n = 1, \dots, (\Delta^{-1} - 1)$  and  $I_{\Delta^{-1}} := [1 - \Delta, 1]$ . Let  $\bar{P}_i$  be a discretized random variable over  $\Upsilon$  of  $P_i$  from  $\Phi(\mathbf{x}, \omega)$ . i.e.,  $\bar{P}_i = (n - \frac{1}{2})\Delta$  if  $P_i \in I_n$  such that  $\mathbb{P}[\bar{P}_i = (n - \frac{1}{2})\Delta] = \mathbb{P}[P_i \in I_n]$ . For any estimator  $\hat{P}_i$  of  $\bar{P}_i$  given the label  $\{Y = i\}$ , by applying Fano's inequality [12][7, Theorem 2.10.1], we have (note that our log has a base  $e$ )

$$\mathbb{P}[\hat{P}_i \neq \bar{P}_i | Y = i] \geq \frac{H(\bar{P}_i | Y = i) - \log 2}{\log \Delta^{-1}} = \frac{H(\bar{P}_i | Y = i) - \log 2}{-\log \Delta}. \quad (20)$$

We note that Shannon entropy and the differential entropy have the following connection [7, Theorem 8.3.1]:

$$H(\bar{P}_i | Y = i) + \log \Delta = h(P_i | Y = i) + \epsilon_i = h(P_i^+) + \epsilon_i, \quad (21)$$

where  $\epsilon_i$  is an adjustment constant depending on  $\Delta$  such that  $\epsilon_i \rightarrow 0$  as  $\Delta \rightarrow 0$ . Note that  $\epsilon_i$  does not have to be non-negative. Then we can rewrite the inequality as follows:

$$\mathbb{P}[\hat{P}_i \neq \bar{P}_i | Y = i] \geq \frac{h(P_i^+) - \log \Delta - \log 2}{-\log \Delta} + \frac{\epsilon_i}{-\log \Delta}. \quad (22)$$

Taking the expectation with respect to  $Y$ , we have

$$\mathbb{E} \left[ \mathbb{P}[\hat{P}_i \neq \bar{P}_i | Y = i] \right] \geq \frac{\sum_i (\mathbb{E} P_i) h(P_i^+) - \log \Delta - \log 2}{-\log \Delta} + \frac{\sum_i (\mathbb{E} P_i) \epsilon_i}{-\log \Delta} =: (**).$$

If we let  $\Delta^{-1} = \lfloor 2e^{H(Y)} \rfloor$ , there exists a  $\delta \geq 0$  such that

$$H(Y) + \log 2 - \delta = -\log \Delta = \log \lfloor 2e^{H(Y)} \rfloor \leq H(Y) + \log 2. \quad (23)$$

We also note that  $\delta \rightarrow 0$  as  $H(Y) \rightarrow \infty$  (or equivalently  $\Delta \rightarrow 0$ ). Therefore, when  $\Delta^{-1} = \lfloor 2e^{H(Y)} \rfloor$ , we have

$$\begin{aligned} (**) &= \frac{\sum_i (\mathbb{E} P_i) h(P_i^+) + H(Y) - \delta}{H(Y) + \log 2 - \delta} + \frac{\sum_i (\mathbb{E} P_i) \epsilon_i}{H(Y) + \log 2 - \delta} \\ &\geq \frac{\sum_i (\mathbb{E} P_i) h(P_i^+) + H(Y)}{H(Y) + \log 2} \left( 1 + \frac{\delta}{H(Y) + \log 2 - \delta} \right) - \frac{\sum_i (\mathbb{E} P_i) |\epsilon_i| + \delta}{H(Y) + \log 2}. \end{aligned} \quad (24)$$

Let  $\epsilon_1 = \frac{\delta}{H(Y) + \log 2 - \delta}$  and  $\epsilon_2 = \frac{\sum_i (\mathbb{E} P_i) |\epsilon_i| + \delta}{H(Y) + \log 2} \geq 0$ . Since  $\epsilon_i \rightarrow 0$  and  $\delta \rightarrow 0$  as  $\Delta \rightarrow 0$ ,  $\epsilon \rightarrow 0$  as  $\Delta \rightarrow 0$ . Therefore Theorem 4.1 follows.

#### A.5 More Beta marginal formulations

With Beta approximation, we are able to describe Beta marginal formulation of MeanSD. Since we are matching the variance of each marginal distribution, the empirical value of MeanSD should be the same as BetaMarginalMeanSD.

$$\text{BetaMarginalMeanSD}[\mathbf{x}] := \frac{1}{C} \sum_{i=1}^C \sqrt{\frac{\alpha_i \beta_i}{(\alpha_i + \beta_i)^2 (\alpha_i + \beta_i + 1)}} = \text{MeanSD}[\mathbf{x}]. \quad (25)$$

82 In [15], the expected information gain has been proposed and studied. We may  
83 also formulate the expected information gain with Beta marginal distributions.

$$\text{BetaMarginalEIG}[\mathbf{x}] := H(Y) - \mathbb{E}H(Y^+ | Y = i)$$

$$84 \quad = \sum_{i=1}^C \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) \left[ \sum_{j=1}^C \left( \frac{\alpha_j + \delta_i(j)}{\alpha_j + \beta_j + 1} \right) \log \left( \frac{\alpha_j + \delta_i(j)}{\alpha_j + \beta_j + 1} \right) - \log \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) \right],$$

(26)

85 where  $Y^+$  is a categorical random variable over the posterior probability given  $Y = i$ ,  $\delta_i(j) = 1$  if  
86  $i = j$ , and  $\delta_i(j) = 0$  otherwise.

## 87 A.6 Beta marginal approximation visualization examples

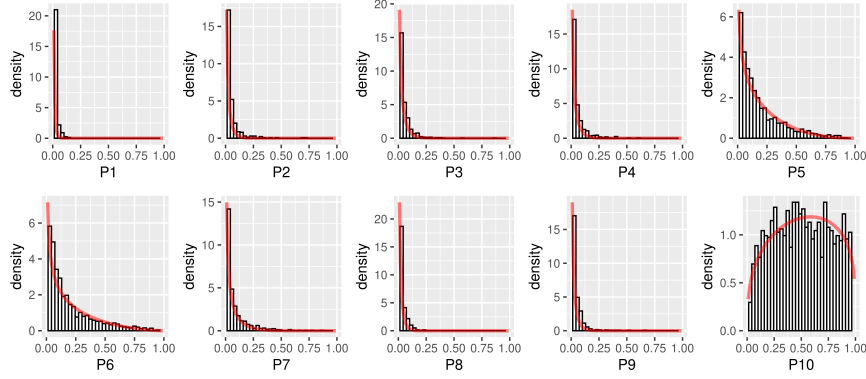


Figure 1: An example of Beta approximations (red lines) for each marginal distribution after applying softmax layer in MNIST dataset. Each Beta distribution is estimated by calculating the sample mean and sample variance of the histogram generated by the Bayesian deep learning model.

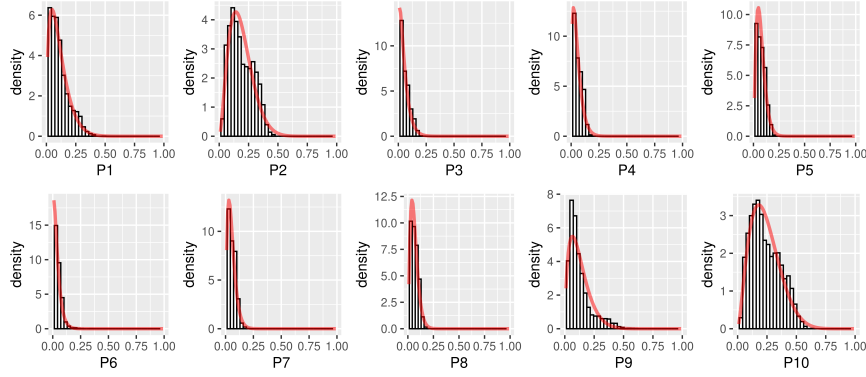


Figure 2: An example of Beta approximations (red lines) for each marginal distribution after applying softmax layer in CIFAR-10 dataset.

88 Figure 1 and Figure 2 shows an example of Beta approximations obtained from the MNIST and  
89 CIFAR-10 datasets.  $P_1, \dots, P_{10}$  show each marginal distribution of the predictive probability of  
90 each digit. We observe that the Beta approximation is a reasonable approximation.

## 91 A.7 Rank correlation study with BetaMarginalEIG

92 A good advantage of explicit formula is that we can study the behavior of each measure directly. For  
93 example, if  $C = 2$  and  $\Phi(\mathbf{x}, \omega) \sim \text{Dirichlet}(\alpha, \beta)$  such that  $P_1 \sim \text{Beta}(\alpha, \beta)$  and  $P_2 \sim \text{Beta}(\beta, \alpha)$ ,  
94 we are able to plot the behavior of each Beta marginal measure.

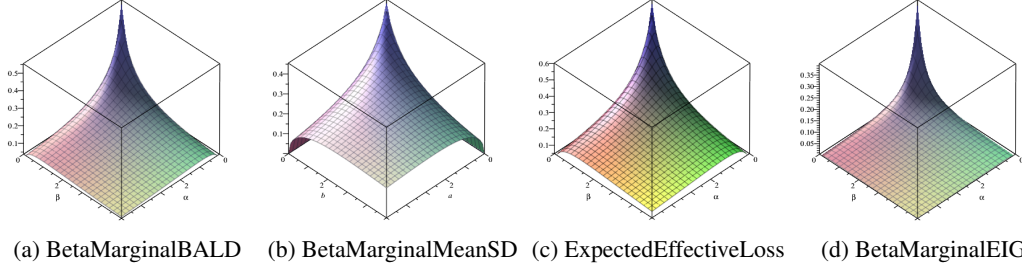


Figure 3: 3D plot of each uncertainty measure when Beta marginal assumption holds.

95 With BetaMarginalEIG, we are able to generate the same type of plot shown in Figure 1. EIG shows  
 96 positive correlations with BALD and MeanSD, but the correlation is around 70% implying that EIG  
 97 might show more variations.

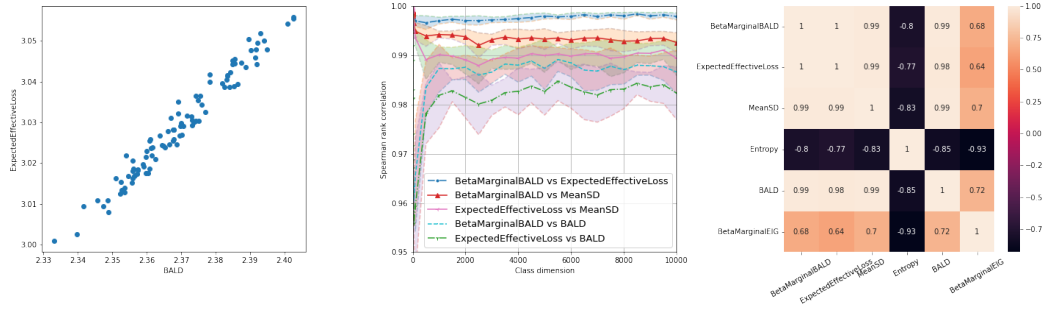


Figure 4: Same experiments with BetaMarginalEIG described in Figure 1 from the main article. This is another independent experiment (as a validation), so the captured correlation values are slightly different.

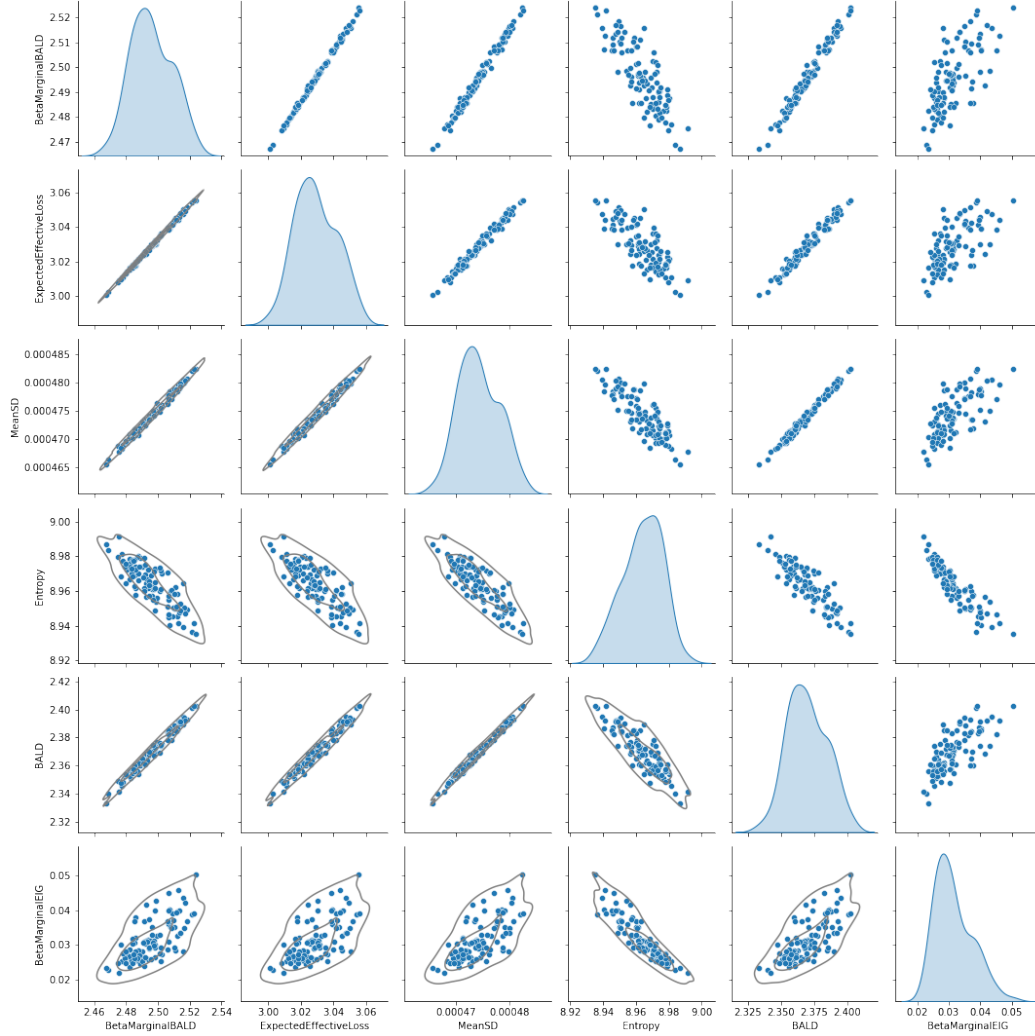


Figure 5: Pairwise scatter plot at  $C = 10,000$  companion with Figure 4.

## 98 A.8 BALD and BetaMarginalBALD

99 Although BALD and BetaMarginalBALD shows high rank-correlation (under softmax applied  
100 Gaussian distribution assumption), we might wonder how much different they are in the value.  
101 We plot the RMSE (rooted mean square error) between two measures. Under Dirichlet distribu-  
102 tion assumption, RMSE between BALD and BetaMarginalBALD is  $< 0.002$  upto  $C \leq 1,000$ .  
103 However, under softmax-applied Gaussian distribution assumption, RMSE between BALD and  
104 BetaMarginalBALD shows  $< 0.07$  upto  $C \leq 1,000$ . This implies that Beta marginal approxi-  
105 mation still preserves a high rank-correlation, but the absolute values are slightly shifted. e.g.,  
106  $\text{BALD}[\mathbf{x}] \approx \text{BetaMarginalBALD}[\mathbf{x}] + \text{err}$  for some constant  $\text{err} \in \mathbb{R}$ . This study also implies that  
107 Beta marginal approximation is a reasonable assumption.

## 108 A.9 Toy example with ExpectedEffectiveLoss and BetaMarginalEIG

109 As observed by high rank-correlation in Figure 4, BALD, ExpectedEffectiveLoss, and Beta-  
110 MarginalEIG show similar selections.

111 Figure 8 shows the active learning curves for ExpectedEffectiveLoss and BetaMarginalEIG with  
112 MNIST and  $3 \times \text{CIFAR-100}$ . This experiment also confirms that BALD and ExpectedEffectiveLoss  
113 are tightly aligned as we show that both are highly correlated. In MNIST, BetaMarginalEIG per-



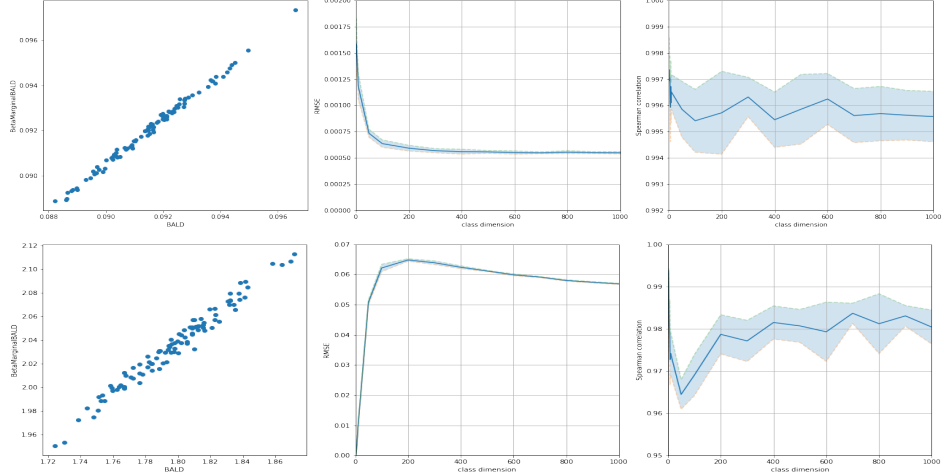


Figure 6: Scatter plot at  $C = 1,000$  between BALD and BetaMarginalBALD (left), RMSE between BALD and BetaMarginalBALD over various class dimensions (middle), and Spearman’s rank correlations over various class dimensions (right). The first row is the result from  $C$ -dimensional 100 random Dirichlet samples. The second row is obtained from softmax-applied 100 random Gaussian samples. Then we repeat the process 10 times. In both cases, we observe  $> 96\%$  rank correlations as well.

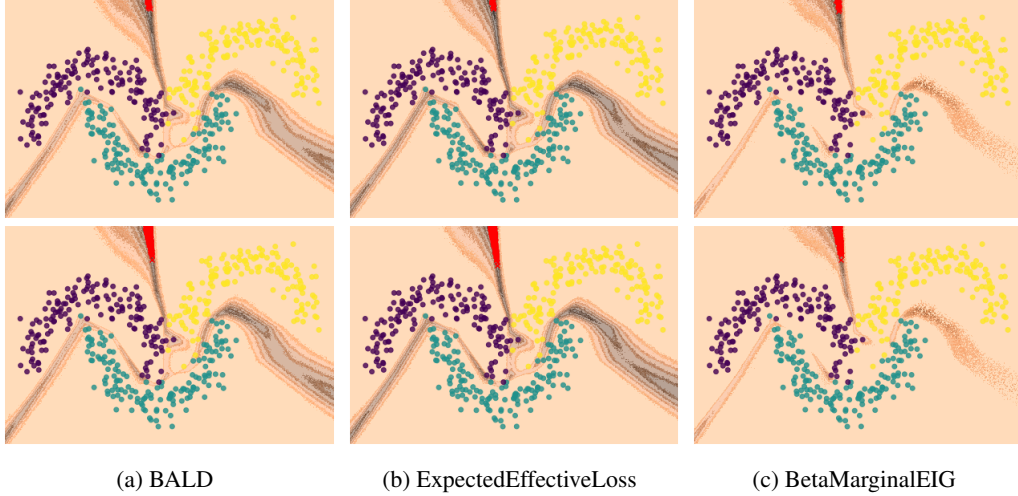


Figure 7: Top- $K$  selected points are marked by red color. The first row shows the top  $K = 25$  point selections. The second row shows the top  $K = 500$  point selections among around 0.6 million uniform grid points. The same experiments shown in Figure 2 in the main article.

114 forms similar to BALD and ExpectedEffectiveLoss. However, in  $3 \times \text{CIFAR-100}$ , BetaMarginalEIG  
 115 performs similar to BALD at first, but it essentially performs better than BALD and similar to the  
 116 random case. Recall that the rank correlation between BALD and BetaMarginalEIG is around 70%.

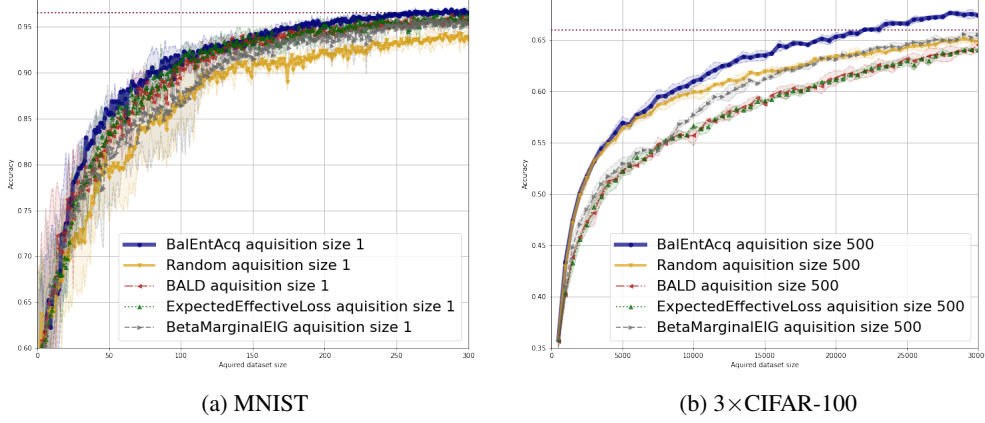


Figure 8: Active learning curves for ExpectedEffectiveLoss and BetaMarginalEIG with MNIST and 3×CIFAR-100.

### 117 A.10 Non-negative BalEntAcq region

118 In this section, we study the non-negative region of  $\text{BalEntAcq}[\mathbf{x}]$ .  $\text{BalEntAcq}[\mathbf{x}]$   
 119 is non-negative when  $\text{MJEnt}[\mathbf{x}] \geq 0$ . Under Beta marginal distribution approxima-  
 120 tion, let  $P_i \sim \text{Beta}(\alpha_i, \beta_i)$  in  $\Phi(\mathbf{x}, \omega)$ . Then we can fully write  $\text{MJEnt}[\mathbf{x}]$  as follows:

$$\begin{aligned}
 \text{MJEnt}[\mathbf{x}] &= \sum_i (\mathbb{E}P_i) h(P_i^+) + H(Y) \\
 &= \sum_{i=1}^C \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) \left[ \log B(\alpha_i + 1, \beta_i) - \alpha_i \Psi(\alpha_i + 1) - (\beta_i - 1) \Psi(\beta_i) - (\alpha_i + \beta_i - 1) \Psi(\alpha_i + \beta_i + 1) - \log \left( \frac{\alpha_i}{\alpha_i + \beta_i} \right) \right].
 \end{aligned}$$

122 Then, we are able to generate a 3D plot and a contour plot of  $\text{BalEnt}[\mathbf{x}]$  when  $C = 2$ . i.e.,  
 123  $\Phi(\mathbf{x}, \omega) \sim \text{Dirichlet}(\alpha, \beta)$  such that  $P_1 \sim \text{Beta}(\alpha, \beta)$  and  $P_2 \sim \text{Beta}(\beta, \alpha)$ .

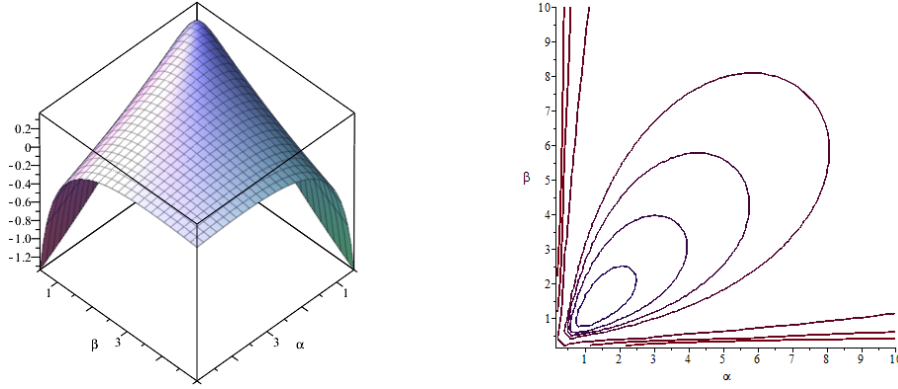


Figure 9: BalEnt 3D plot (left) and Positive BalEnt contour plot (right) over parameters  $(\alpha, \beta)$ . For the contour plot, starting from the outside, contours are generated when  $\text{BalEnt}[\mathbf{x}] = -3, -2, -1, 0, 0.1, 0.2, 0.3$ .

124 Figure 9 suggests that in Dirichlet distribution’s parameter space, there exist (uncountably and)  
 125 infinitely many parameters which produce non-negative  $\text{BalEnt}[\mathbf{x}]$  values. Then we also plot the  
 126 non-negative region (red shaded) of  $\text{BalEntAcq}$  in our toy example.

127 Figure 10-(b) illustrates that there exist infinitely many points which produce the same  $\text{BalEntAcq}[\mathbf{x}]$   
 128 values. Therefore we may imagine that we are conducting a uniform sampling on each contour  
 129 surface  $\{\text{BalEnt}[\mathbf{x}] = \lambda\}$  for each  $\lambda \geq 0$ , then moving to the surface for each  $\lambda \geq 0$ . This observation  
 130 also explains how  $\text{BalEnt}[\mathbf{x}]$  diversifies the selection near the decision boundary.

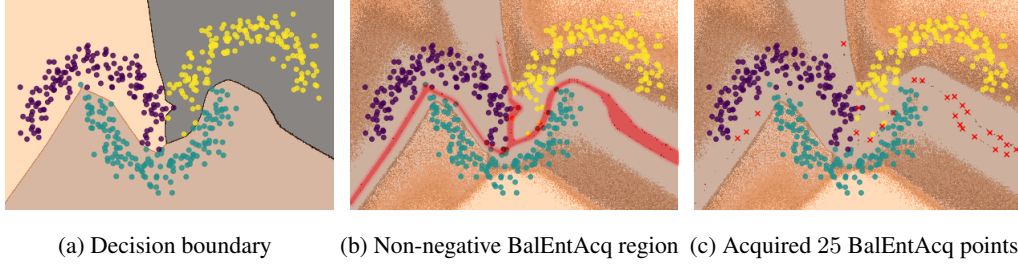


Figure 10: Non-negative BalEntAcq region illustration from the toy example

### A.11 Implementation of balanced entropy active learning

Implementation of BalEntAcq active learning is almost the same as the usual MC dropout-based uncertainty methods. The difference is the dropout samples at inference time; BalEntAcq requires an additional estimation step of Beta parameters for each marginal distribution. Algorithm 1 explains the whole steps of BalEntAcq active learning. Moreover, we do not apply any early stopping criteria in each training because we understand that early stopping conflicts with dropout-based model training. In other words, if we stop too early in model training by observing validation accuracy or loss, we observe that model weights cannot reach to the fully mixing states of the randomness in MC dropouts, similar to the early stage of Markov Chain Monte Carlo (MCMC).

---

#### Algorithm 1: BalEntAcq active learning algorithm

---

- 1 **Input:** 1) Unlabeled dataset  $\mathcal{D}_{\text{pool}}$ , 2) initially labelled dataset  $\mathcal{D}_{\text{training}}^{(0)}$ , 3) the number of dropout samples  $M$  at inference time, 4) active learning budget  $K$  for each iteration, 5) total active learning budget  $K^{\text{tot}}$
  - 2 **Initialize** all weights of Bayesian neural network  $\Phi$  and set  $n \leftarrow 0$
  - 3 **Repeat** at iteration  $n \geq 0$
  - 4    Train the model  $\Phi$  with  $\mathcal{D}_{\text{training}}^{(n)}$
  - 5    For each  $\mathbf{x} \in \mathcal{D}_{\text{pool}} \setminus \mathcal{D}_{\text{training}}^{(n)}$ ,
  - 6     Generate  $M$  dropout samples
  - 7     Estimate Beta parameters  $(\alpha_i, \beta_i)$  for each marginal distribution
  - 8     Calculate  $\text{BalEntAcq}[\mathbf{x}]$
  - 9    Set  $\mathcal{D}_{\text{training}}^{(n+1)} \leftarrow \mathcal{D}_{\text{training}}^{(n)} \cup \left\{ \text{top } K \text{ BalEntAcq-valued } \mathbf{x} \in \mathcal{D}_{\text{pool}} \setminus \mathcal{D}_{\text{training}}^{(n)} \right\}$ , and  $n \leftarrow n + 1$
  - 10 **Until**  $|\mathcal{D}_{\text{training}}^{(n-1)}|$  reaches to  $K^{\text{tot}}$
- 

### A.12 More experimental details

Table 1 shows a summary of dataset, configurations, and hyperparameters used in our experiments. For each experiment, we repeat 3 times to generate the full active learning accuracy curve.

Scenario	Dataset	# Classes	$K$	$K^{\text{tot}}$	Backbone	Loss	Image size	Batch size	Optimizer	Epochs	Learning rate	Dropout	MC trials
Full dropouts	MNIST	10	1	300	CNN	Cross-entropy	$28 \times 28$	128	Adam	150	0.01	50%	100
Fixed feature	CIFAR-100	100	500	10,000	ResNet-50	Cross-entropy	$224 \times 224$	128	Adam	150	0.0003	20%	100
Redundant images	CIFAR-100	100	500	30,000	ResNet-50	Cross-entropy	$224 \times 224$	128	Adam	150	0.0003	20%	100
Pre-trained backbone	TinyImageNet	200	1,500	30,000	ResNet-50	Cross-entropy	$64 \times 64$	128	Adam	100	0.0003	20%	100

Table 1: Detailed configurations used in our experiments.

In SimCLR [6] feature training, we trained ResNet-50 with  $224 \times 224$  image size, 192 batch size, 500 epochs, and 0.0003 learning rate with Adam optimizer for CIFAR-10/CIFAR-100.

### A.12.1 Simply last+ layer Bayesian

Dropout-based Bayesian neural network typically requires adding dropout layer with ReLU activation for each convolutional or linear layer to approximate a Gaussian process [18]. But this requires a high computational cost. Therefore we adopt several additional last layer dropout architecture to build a Bayesian neural network equipped with Beta approximation. There exist several different lines of works to justify the effectiveness of this simple last layer modification [33, 36, 4, 26, 21]. More precisely, similar to Laplace approximation applied at the last layer [26, See Theorem 2.4] and [21], we may replace several last linear layers with a dropout applied and ReLU activated linear layers.

For example, we may add two or more dropout layers after ResNet-50 fixed backbone in our CIFAR-100 experiments to avoid any pathological cases [14]. In practice, we observe a single dropout layer application is sufficient to achieve our Beta approximated marginals as shown below. We note that in MNIST experiment, we use 50% dropout rate and for all other our experiments, we use 20% dropout rate.

### A.12.2 Choices of prioritization in BalEntAcq

In this section, we study the impact of the prioritization in  $\text{BalEnt}[\mathbf{x}]$ .

P1.  $P1[\mathbf{x}] = -\text{BalEnt}[\mathbf{x}]$ . This is the case where we put higher priority when the posterior uncertainty captures very small values. Note that this also includes high epistemic uncertainty (BALD) valued case. For example, when  $C = 2$  with  $\Phi(\mathbf{x}, \omega) \sim \text{Dirichlet}(\alpha, \beta)$ , the posterior uncertainty goes to  $-\infty$  as  $\alpha \rightarrow 0$  and  $\beta \rightarrow 0$ . Therefore  $\text{BalEnt}[\mathbf{x}] \rightarrow -\infty$ . But this case also achieves the highest epistemic uncertainty.

P2.  $P2[\mathbf{x}] = \begin{cases} \text{BalEnt}[\mathbf{x}]^{-1} & \text{if } \text{BalEnt}[\mathbf{x}] \geq 0, \\ \text{BalEnt}[\mathbf{x}] & \text{if } \text{BalEnt}[\mathbf{x}] < 0 \end{cases}$ . This is the same case as our proposed acquisition measure.

P3.  $P3[\mathbf{x}] = \text{BalEnt}[\mathbf{x}]$ . This is the case where we put higher priority when the posterior uncertainty captures very high values (close to zero). As discussed in Section 4.1, we want to prioritize more when the information imbalance gap is higher.

Figure 11 and Table 2 show that selecting the points near  $\text{BalEnt}[\mathbf{x}] \approx 0$  is a better way to improve the accuracy as we discussed in Section 4.1. When we prioritize the small posterior uncertainty case, P1 shows a very poor performance in a fixed feature scenario. However, under the backbone and augmentation scenario, the performance of P1 is similar to the high posterior uncertainty case of P3. This could be because of the evolution of the feature space and the batch normalization during the active learning process. i.e. previously captured uncertainty values will not be preserved under the backbone with augmentation scenario.

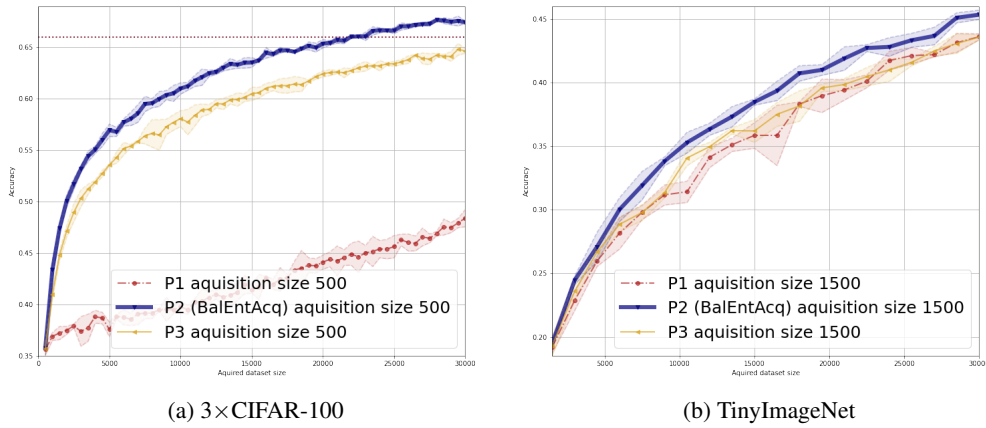


Figure 11: Active learning curves depending on different prioritization.

Scenario	Redundant images + Fixed feature					Backbone + Augmentation			
Dataset/Acq. Size/Test size	3×CIFAR-100/500/10,000					TinyImageNet/1,500/10,000			
Train Size/Pool Size	5,000/150,000	15,000/150,000	25,000/150,000	30,000/150,000	6,000/100,000	15,000/100,000	24,000/100,000	30,000/100,000	
P1	37.6 ± 0.7%	41.5 ± 0.4%	45.6 ± 1.1%	48.4 ± 0.8%	28.2 ± 1.2%	35.8 ± 1.0%	41.7 ± 0.7%	43.6 ± 0.2%	
P2 (BalEntAcq)	<b>56.9 ± 0.6%</b>	<b>63.5 ± 0.4%</b>	<b>66.6 ± 0.3%</b>	<b>67.4 ± 0.1%</b>	<b>30.0 ± 0.9%</b>	<b>38.5 ± 0.2%</b>	<b>42.8 ± 0.7%</b>	<b>45.3 ± 0.4%</b>	
P3	53.6 ± 0.1%	60.5 ± 0.6%	63.4 ± 0.2%	64.7 ± 0.2%	28.9 ± 0.5%	36.2 ± 0.9%	41.0 ± 0.9%	43.6 ± 0.2%	

Table 2: Selected accuracy table depending on different prioritization. Mean and standard deviation are from 3 repeated experiments. The best performance in each column is shown in **bold**.

### 177 A.12.3 Behavior of different precision levels

178 As shown in the proof of Theorem 4.1, we may have some freedom to choose the level of the  
179 precision in the  $P_i$  estimation. Therefore we report the active learning behavior for other precision  
180 choices. It is not clear which precision level achieves the optimal performance, but our preference  
181 of  $-\log \Delta \approx H(Y) + \log 2$  shows a reasonably superior performance in any scenario as shown in  
182 Figure 12 and Table 3.

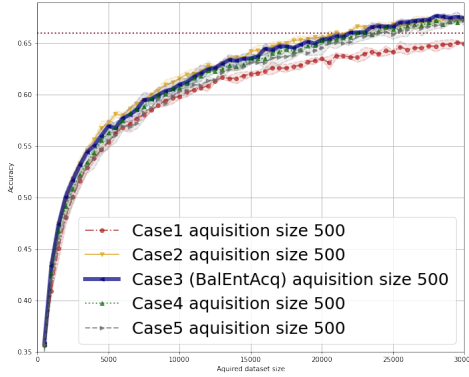
183 Case 1.  $-\log \Delta \approx H(Y) - \log 2$ ,

184 Case 2.  $-\log \Delta \approx H(Y)$ ,

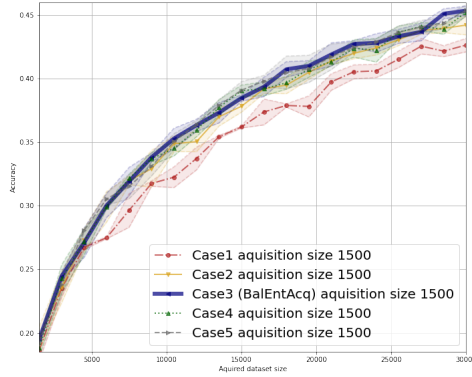
185 Case 3.  $-\log \Delta \approx H(Y) + \log 2$  (our choice),

186 Case 4.  $-\log \Delta \approx H(Y) + 2 \log 2$ ,

187 Case 5.  $-\log \Delta \approx H(Y) + 3 \log 2$ .



(a) 3×CIFAR-100



(b) TinyImageNet

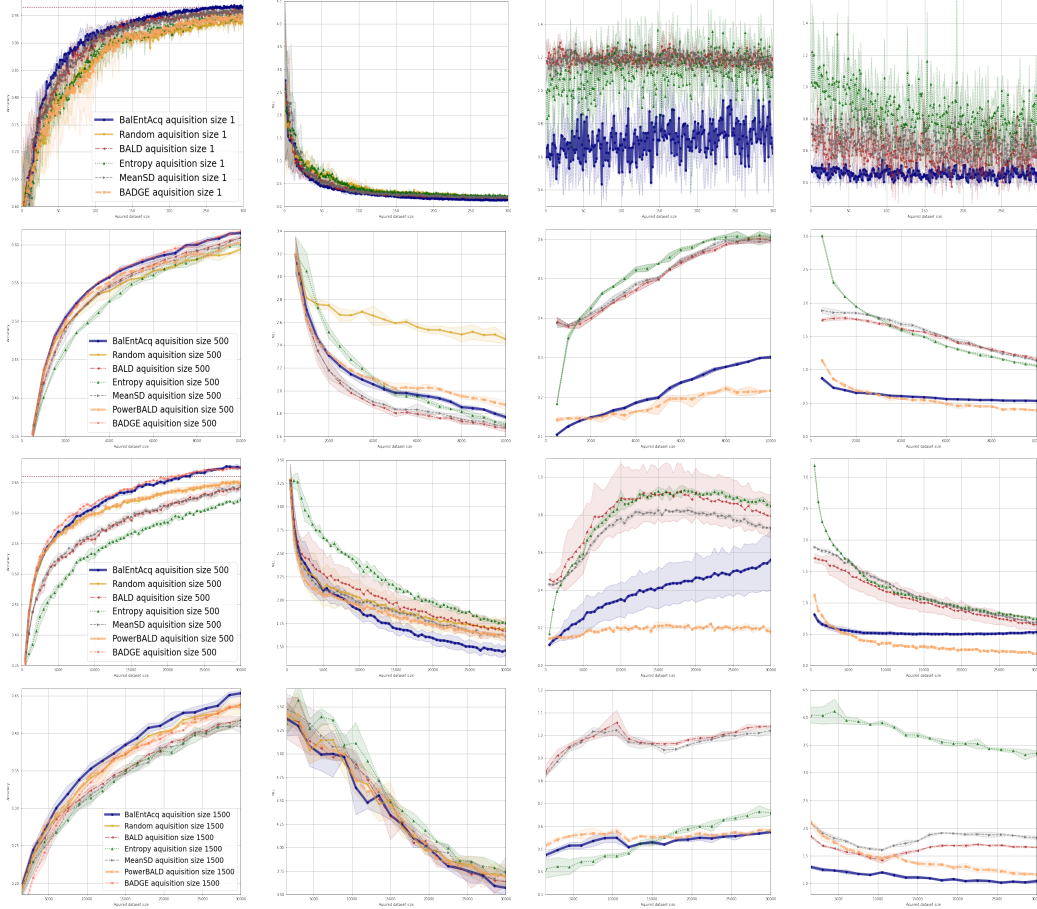
Figure 12: Active learning curves depending on different precision levels.

Scenario	Redundant images + Fixed feature					Backbone + Augmentation			
Dataset/Acq. Size/Test size	3×CIFAR-100/500/10,000					TinyImageNet/1,500/10,000			
Train Size/Pool Size	5,000/150,000	15,000/150,000	25,000/150,000	30,000/150,000	6,000/100,000	15,000/100,000	24,000/100,000	30,000/100,000	
Case1	55.4 ± 0.7%	62.0 ± 0.3%	64.2 ± 0.1%	65.0 ± 0.1%	27.5 ± 0.1%	36.2 ± 0.1%	40.6 ± 0.5%	42.6 ± 0.5%	
Case2	<b>57.2 ± 0.3%</b>	<b>64.2 ± 0.2%</b>	<b>66.9 ± 0.3%</b>	<b>67.4 ± 0.2%</b>	29.9 ± 1.2%	37.8 ± 0.5%	42.4 ± 0.5%	44.2 ± 0.8%	
Case3 (BalEntAcq)	56.9 ± 0.6%	63.5 ± 0.4%	66.6 ± 0.3%	<b>67.4 ± 0.1%</b>	30.0 ± 0.9%	38.5 ± 0.2%	42.8 ± 0.7%	45.3 ± 0.4%	
Case4	56.3 ± 0.7%	63.6 ± 0.2%	66.4 ± 0.3%	<b>67.4 ± 0.2%</b>	29.9 ± 0.2%	39.0 ± 0.5%	42.2 ± 0.9%	45.2 ± 0.3%	
Case5	55.6 ± 0.6%	63.2 ± 0.1%	65.9 ± 0.5%	67.2 ± 0.5%	<b>30.5 ± 0.5%</b>	<b>39.1 ± 0.7%</b>	<b>42.9 ± 0.3%</b>	<b>45.4 ± 0.5%</b>	

Table 3: Selected accuracy table depending on different precision levels. Mean and standard deviation are from 3 repeated experiments. The best performance in each column is shown in **bold**.

### 188 A.12.4 More details about the main experiment

189 Figure 13 shows the full active learning curves, negative log-likelihood, average epistemic uncertainty  
190 for selected samples, and average aleatoric uncertainty for selected samples. We note that our proposed  
191 method selects neither high epistemic uncertainty (=model uncertainty) nor aleatoric uncertainty  
192 (=data uncertainty) samples. Nevertheless, BelEntAcq shows a good performance improvement  
193 during the active learning iterations. Furthermore, we observe that BelEntAcq keeps choosing low  
194 aleatoric uncertainty points but increasing epistemic uncertainty points.



(a) Accuracy (b) Negative log-likelihood (c) Epistemic uncertainty (d) Aleatoric uncertainty

Figure 13: Full active learning curves obtained from different scenarios. From the top row, each row represents a result of MNIST, CIFAR-100, 3×CIFAR-100, and TinyImageNet.

#### 195 A.12.5 Model architectures

196 In this section, we describe model architectures what we have used in our experiments.

#### 197 Toy example - moon dataset

```

198 BNN (
199     (classifier): Sequential(
200         (0): Linear(in_features=2, out_features=72, bias=True)
201         (1): ReLU(inplace=True)
202         (2): Linear(in_features=72, out_features=72, bias=True)
203         (3): Dropout(p=0.2, inplace=False)
204         (4): ReLU(inplace=True)
205         (5): Linear(in_features=72, out_features=72, bias=True)
206         (6): Dropout(p=0.2, inplace=False)
207         (7): ReLU(inplace=True)
208         (8): Linear(in_features=72, out_features=3, bias=False)
209     )
210 )

```

#### 211 MNIST

212 CNNBNN (



```

213 (features): Sequential(
214   (0): CNN2D(in_channel=1, out_channel=32, kernel_size=5,
215             stride=1, dropout_p=0.5, apply_max_pool=True,
216             apply_relu=True)
217   (1): CNN2D(in_channel=32, out_channel=64, kernel_size=5,
218             stride=1, dropout_p=0.5, apply_max_pool=True,
219             apply_relu=True)
220 )
221 (classifier): Sequential(
222   (0): Linear(in_features=1024, out_features=128, bias=True)
223   (1): Dropout(p=0.5, inplace=False)
224   (2): ReLU(inplace=True)
225   (3): Linear(in_features=128, out_features=10, bias=False)
226 )
227 )

```

## 228 **SVHN**

```

229 RESNETBNN(
230   (features): ResNet18(remove_last_fully_connected_layer=True)
231   (classifier): Sequential(
232     (0): Linear(in_features=512, out_features=512, bias=True)
233     (1): Dropout(p=0.2, inplace=False)
234     (2): ReLU(inplace=True)
235     (3): Linear(in_features=512, out_features=10, bias=False)
236   )
237 )

```

## 238 **CIFAR-100 and 3×CIFAR-100**

```

239 RESNETCLASSIFIER(
240   (classifier): Sequential(
241     (0): Linear(in_features=2048, out_features=2048, bias=True)
242     (1): Dropout(p=0.2, inplace=False)
243     (2): ReLU(inplace=True)
244     (3): Linear(in_features=2048, out_features=100, bias=False)
245   )
246 )

```

## 247 **TinyImageNet**

```

248 RESNETBNN(
249   (features): ResNet50(remove_last_fully_connected_layer=True)
250   (classifier): Sequential(
251     (0): Linear(in_features=2048, out_features=2048, bias=True)
252     (1): Dropout(p=0.2, inplace=False)
253     (2): ReLU(inplace=True)
254     (3): Linear(in_features=2048, out_features=200, bias=False)
255   )
256 )

```

## 257 **A.13 Relationship with the efficient active learning algorithm with abstention**

258 It is well-known that any active learning method cannot improve the label complexity better than  
 259 passive learning (random acquisition) in general [35, 23, 5]. Therefore under some conditions on  
 260 labels or models, it is possible to achieve exponential savings [2, 19, 9, 22, 10, 20, 37, 25, 31]. Zhu and  
 261 Nowak recently proposed an efficient active learning algorithm with abstention in binary classification  
 262 [39] in the parametric setting with high probability. This section illustrates the relationship with the  
 263 recently proposed efficient Algorithm by Zhu and Nowak [39]. Note that we are not proving the  
 264 equivalence between the two algorithms.

As demonstrated in A.10, we can see that our proposed BalEntAcq method shares a high similarity with active learning strategy with abstention [28, 32, 31, 39].

Intuitively, Algorithm 1 in [39] works in the following way. Set an abstention parameter  $\gamma > 0$ . Train a binary classifier  $h(x)$ . For unlabelled point  $x \in \mathcal{X}$ , we can calculate a uncertainty bound,  $UB[x] := [\text{lcb}(x), \text{ucb}(x)]$ . If  $UB[x] \subseteq [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]$ , we abstain the point  $x$ , i.e., we do not query the point  $x$ . If  $\frac{1}{2} \in UB[x]$  and  $UB[x] \not\subseteq [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]$ , we query the point  $x$ . At each iteration  $m$ , we add geometrically increasing  $2^m$  queried points.

The key insight of this Algorithm 1 to achieve exponential label savings is to abstain from the point very close to the decision boundary. Similarly, as we demonstrated in A.10, our BalEntAcq finds a margin by focusing on the positive sign of BalEntAcq[x] which corresponds to finding  $x$  outside the abstention region such that  $|x - \frac{1}{2}| > \gamma$  near the decision boundary. Then following the positive BalEntAcq[x] values, we acquire points toward the decision boundary direction, which corresponds to the condition  $\frac{1}{2} \in UB[x]$ . We know that the point near the decision boundary should have high aleatoric uncertainty. On the other hand, Corollary 1 implies that aleatoric uncertainty is increasing as  $\alpha, \beta \rightarrow +\infty$ . So MJEnt[x]  $\rightarrow -\infty$ . Then BalEntAcq[x]  $\rightarrow -\infty$ . Therefore, our BalEntAcq[x] will acquire points near the decision boundary but will not acquire the point if it's too close to the decision boundary. This strategy in our BalEntAcq[x] exactly matches the key insight of Algorithm 1. So we may be able to theoretically guarantee that our proposed acquisition function BalEntAcq[x] could be a universally working active learning algorithm by achieving exponential label savings.

#### A.14 Acquisition time complexity

We note the time complexity of the acquisition calculation for each active learning iteration. We denote by  $N$  number of unlabelled points,  $C$  number of classes,  $K$  the acquisition size. For BADGE, we use the last layer feature vector and then apply k-means++.

Method	BalEntAcq	BALD	Entropy	MeanSD	PowerBALD	BADGE	Random
Time Complexity	$O(CN)$	$O(CN)$	$O(CN)$	$O(CN)$	$O(CN)$	$O(CNK)$	$O(N)$
Case	Average Elapsed Time (sec)						
MNIST with Acq. size 1	7.1	6.9	6.8	6.3	—	—	0.1
CIFAR-100 with Acq. size 500	10.2	9.4	9.5	9.6	9.6	302.5	0.1
3×CIFAR-100 with Acq. size 500	18.9	18.5	18.4	18.2	18.4	1227.4	0.3
SVHN with Acq. size 2500	20.7	19.7	19.3	19.4	20.3	85.4	0.1
TinyImageNet with Acq. size 1500	183.4	178.7	178.4	176.4	182.0	4936.1	0.2

Table 4: First two rows show the theoretical time complexity. Remaining rows present the average calculation time what we observed in our experiments.

#### A.15 More experiments with smaller acquisition size

In this section, we conduct more experiments with 3×MNIST and 3×CIFAR-10 by adding more baselines such as VarRatio [x] :=  $1 - \max_i \mathbb{E}P_i$  [16], BatchBALD, and CoreSet. The main purpose of these experiments is to test the relatively smaller acquisition size. We acquire 25 points for each active learning iteration. For 3×MNIST, we use CNN architectures. For 3×CIFAR-10, we fix the feature space obtained from SimCLR [6], the same setting we used in our main experiments. Overall, the additional experimental results are well-aligned with our main results. We observe that BADGE is the best performing baseline. However, we note that BADGE is not linearly scalable, and it requires more computational costs. Figure 14 and Table 5 show full results.



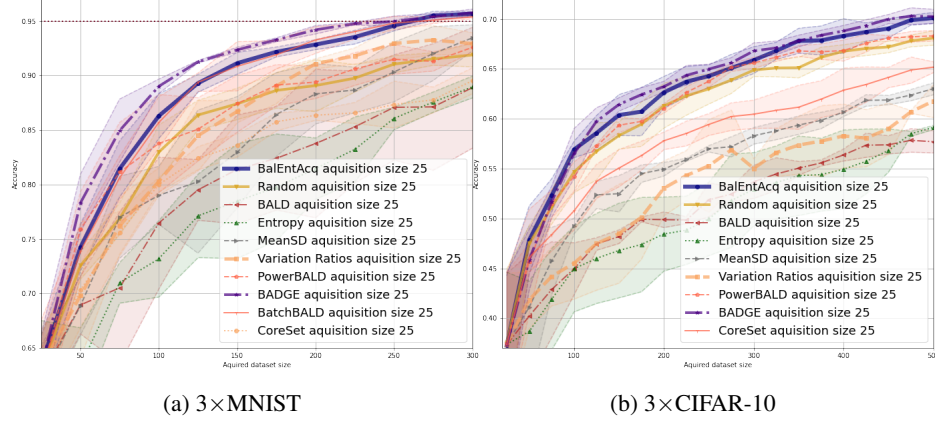


Figure 14: Active learning curves of smaller batch size with 3×MNIST and 3×CIFAR-10.

Scenario	Full dropouts + CNN			Fixed feature		
Dataset/Acq. Size/Test size	3×MNIST/25/10,000			3×CIFAR-10/25/10,000		
Train Size/Pool Size	100/180,000	200/180,000	300/180,000	200/150,000	300/150,000	500/150,000
Random	83.0 ± 2.5%	89.1 ± 2.4%	91.9 ± 1.1%	61.3 ± 2.3%	64.9 ± 0.6%	68.1 ± 0.7%
BALD	76.5 ± 6.0%	83.8 ± 2.8%	88.9 ± 5.5%	49.8 ± 0.8%	53.7 ± 1.7%	57.7 ± 1.1%
Entropy	73.2 ± 3.5%	81.5 ± 2.8%	89.0 ± 1.0%	48.5 ± 3.7%	53.0 ± 3.7%	59.1 ± 0.7%
MeanSD	79.1 ± 2.7%	88.3 ± 2.8%	93.5 ± 1.2%	55.0 ± 0.7%	58.3 ± 0.8%	63.0 ± 0.6%
Variation Ratios	80.3 ± 1.3%	91.1 ± 0.5%	92.9 ± 1.0%	53.0 ± 2.7%	55.0 ± 1.0%	61.8 ± 1.6%
PowerBALD	83.8 ± 4.5%	89.4 ± 1.5%	92.6 ± 0.4%	61.0 ± 0.5%	65.6 ± 0.6%	68.3 ± 0.5%
BADGE (not-scalable)	<b>89.0 ± 0.7%</b>	<b>94.2 ± 0.4%</b>	<b>95.8 ± 0.3%</b>	<b>63.2 ± 0.5%</b>	<b>66.8 ± 0.8%</b>	<b>70.3 ± 0.7%</b>
BatchBALD (not-scalable)	85.7 ± 2.4%	93.3 ± 1.2%	95.4 ± 0.2%	—	—	—
CoreSet (not-scalable)	80.0 ± 0.7%	86.4 ± 0.5%	89.5 ± 1.2%	57.8 ± 0.9%	60.5 ± 1.4%	65.2 ± 0.5%
BalEntAcq (ours)	<b>86.3 ± 2.0%</b>	<b>92.8 ± 0.5%</b>	<b>95.7 ± 0.2%</b>	<b>62.7 ± 1.6%</b>	<b>65.9 ± 1.4%</b>	<b>70.1 ± 0.5%</b>

Table 5: Selected accuracy table. Mean and standard deviation are from 3 repeated experiments.

## 297 A.16 Miscellaneous

298 In our prioritization experiment, one might be interested in the reciprocal form of  $\text{BalEnt}[\mathbf{x}]$  itself.  
 299 But in this case, the majority of acquired points have positive values. So there’s no meaningful  
 300 difference with  $\text{P2}[\mathbf{x}]$ . However, as the result of  $\text{P1}[\mathbf{x}]$  suggests, adding negative values close to  $-\infty$   
 301 should not be helpful to improve the active learning performance. Figure 15 shows the additional  
 302 prioritization active learning curve.

303 P4.  $\text{P4}[\mathbf{x}] = \text{BalEnt}[\mathbf{x}]^{-1}$ . This is the case where we put the reverse priority for the negative  
 304 value case after acquiring all positive values.

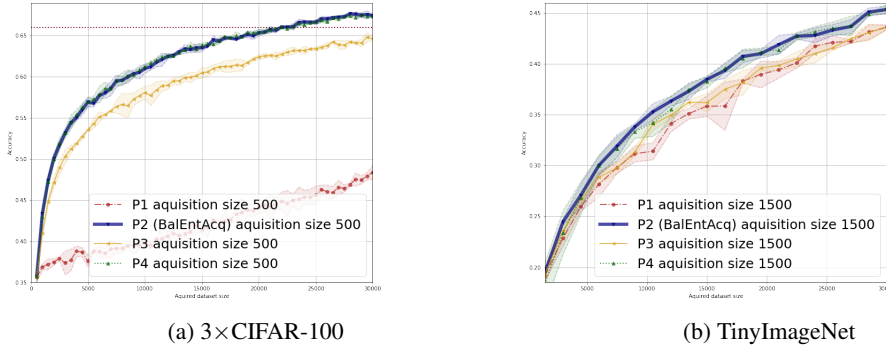


Figure 15: Active learning curves depending on different prioritization.

### 305 A.17 Comparison with CoreMSE

306 Recently, the Bayesian active learning framework considering the Expected Loss Reduction (ELR)  
 307 for the optimal Bayes classifier has been proposed [38, 34]. Under this framework, they attempt to  
 308 optimize the loss reduction in a holistical way, accounting for average loss reduction from all points.  
 309 However, this non-parametric approach requires a very expensive computational cost. With a large  
 310 dataset size, ELR [38], wMOCU [38], CoreLog [34], and CoreMSE [34] require a vast memory size  
 311 unless we apply size reductions on the data space and MC samples [34]. If the number of classes  
 312 is large, running the algorithm in practice is impossible. Therefore the naive application of the  
 313 ELR-based algorithm is not scalable. Moreover, both works have pitfalls in the convergence proof  
 314 by assuming the finite data and parameter space. Both pieces of the works end up with null proof.  
 315 Nevertheless, we tested the performance of CoreMSE [34] with MNIST, seemingly the best method  
 under this framework. Figure 16 and Table 6 show the full active learning results.

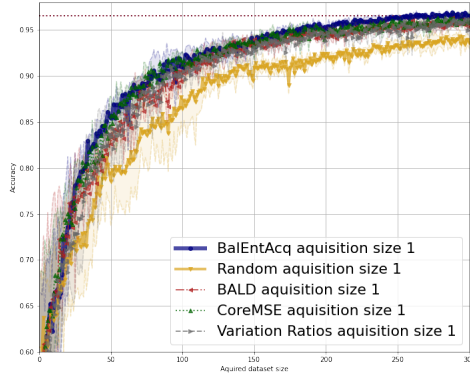


Figure 16: Active learning curves with CoreMSE in MNIST

316

Scenario	Full dropouts + CNN		
Dataset/Acq. Size/Test size	MNIST/1/10,000		
Train Size/Pool Size	50/60,000	100/60,000	300/60,000
Random	78.6 $\pm$ 4.9%	86.4 $\pm$ 2.7%	93.6 $\pm$ 0.7%
BALD	82.6 $\pm$ 1.3%	90.5 $\pm$ 0.8%	95.3 $\pm$ 0.4%
Variation Ratios	83.4 $\pm$ 3.2%	90.0 $\pm$ 1.2%	96.2 $\pm$ 0.2%
CoreMSE	85.3 $\pm$ 2.1%	91.3 $\pm$ 1.3%	95.8 $\pm$ 0.8%
BalEntAcq (ours)	<b>85.4 <math>\pm</math> 1.0%</b>	<b>91.4 <math>\pm</math> 1.3%</b>	<b>96.5 <math>\pm</math> 0.1%</b>

Table 6: Selected accuracy table. Mean and standard deviation are from 3 repeated experiments.

### 317 A.18 Application to another Bayesian neural network with variational dropouts

318 In this section, we report our active learning experiment when we train a Bayesian neural network  
 319 with variational dropouts [24] with 3×CIFAR-10 with acquisition size 50 under a fixed feature  
 320 scenario.

321 We use an Adam optimizer with a learning rate of 0.0003 and 500 epochs in each experiment.  
 322 Compared to MC-dropout Bayesian neural network models, we observe that the convergence with  
 323 variational dropouts is not stable, so it requires much longer epochs if we newly train the model at  
 324 each active learning iteration. Therefore, we continue to train the model from the previously trained  
 325 model at each iteration except the initial iteration so that the convergence can be more stable.

326 Here is the architecture we used for our 3×CIFAR-10 experiment.

```
327 VARIATIONAL_DROPOUT_CLASSIFIER(  
328     (classifier): Sequential(  

```

```

329         (0): VariationalDropout(in_features=2048, out_features=1024)
330         (1): VariationalDropout(in_features=1024, out_features=1024)
331         (2): Linear(in_features=1024, out_features=10, bias=False)
332     )
333 )

```

334 We observe a similar result from the MC-dropout Bayesian neural networks. Our BalEntAcq consistently outperforms other linearly scalable baselines and is eventually on par with BADGE.

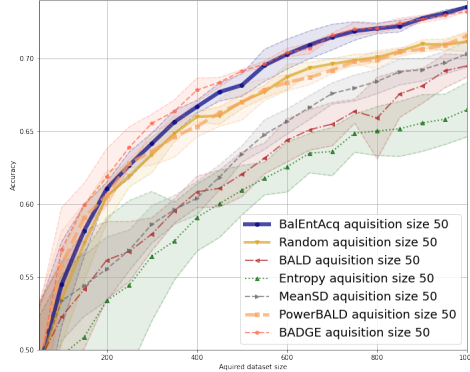


Figure 17: Active learning curves with variational dropouts in 3×CIFAR-10

335

Scenario	Fixed feature + variational dropouts		
Dataset/Acq. Size/Test size	3×CIFAR-10/1/10,000		
Train Size/Pool Size	500/50,000	750/50,000	1000/50,000
Random	67.0 ± 0.5%	69.9 ± 0.4%	71.2 ± 0.7%
BALD	62.1 ± 1.5%	66.4 ± 0.8%	69.5 ± 0.8%
Entropy	60.8 ± 1.7%	64.9 ± 1.3%	66.5 ± 1.9%
MeanSD	63.5 ± 0.6%	68.0 ± 0.9%	70.3 ± 0.9%
PowerBALD	67.0 ± 1.2%	69.8 ± 0.2%	71.2 ± 0.2%
BADGE (not-scalable)	<b>69.1 ± 0.1%</b>	<b>72.0 ± 0.4%</b>	<b>73.2 ± 0.1%</b>
BalEntAcq (ours)	<b>68.2 ± 0.8%</b>	<b>71.9 ± 0.6%</b>	<b>73.5 ± 0.2%</b>

Table 7: Selected accuracy table. Mean and standard deviation are from 3 repeated experiments.

## 336 References

- 337 [1] Anonymous. Analytic mutual information in bayesian neural networks. *To appear in 2022*  
338 *IEEE International Symposium on Information Theory (ISIT)*, 2022.
- 339 [2] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In  
340 *International Conference on Computational Learning Theory*, pages 35–50. Springer, 2007.
- 341 [3] Frits Beukers. Special functions (encyclopedia of mathematics and its applications 71). *Bulletin*  
342 *of the London Mathematical Society*, 33(1):116–127, 2001.
- 343 [4] Nicolas Brosse, Carlos Riquelme, Alice Martin, Sylvain Gelly, and Éric Moulines. On last-layer  
344 algorithms for classification: Decoupling representation from uncertainty estimation. *arXiv*  
345 *preprint arXiv:2001.08049*, 2020.
- 346 [5] Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions*  
347 *on Information Theory*, 54(5):2339–2353, 2008.

- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [7] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- [8] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*, volume 2. Springer Science & Business Media, 2007.
- [9] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pages 249–263. Springer, 2005.
- [10] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *The Journal of Machine Learning Research*, 13(1):2655–2697, 2012.
- [11] Nader Ebrahimi, Ehsan S Soofi, and Shaoqiong Zhao. Information measures of dirichlet distribution with applications. *Applied Stochastic Models in Business and Industry*, 27(2):131–150, 2011.
- [12] Robert M Fano. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.
- [13] Gerald B Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.
- [14] Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020.
- [15] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32, 2019.
- [16] L.C. Freeman. *Elementary Applied Statistics: For Students in Behavioral Science*. For Students in Behavioral Science. Wiley, 1965.
- [17] J. Fritz. An approach to the entropy of point processes. *Periodica Mathematica Hungarica*, 3(1-2):73–83, 1973.
- [18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- [19] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007.
- [20] Steve Hanneke. Theory of active learning. *Foundations and Trends in Machine Learning*, 7(2-3), 2014.
- [21] Marius Hobbhahn, Agustinus Kristiadi, and Philipp Hennig. Fast predictive uncertainty for classification with bayesian deep networks. *arXiv preprint arXiv:2003.01227*, 2020.
- [22] Daniel Joseph Hsu. *Algorithms for active learning*. PhD thesis, UC San Diego, 2010.
- [23] Matti Kääriäinen. Active learning in the non-realizable case. In *International Conference on Algorithmic Learning Theory*, pages 63–77. Springer, 2006.
- [24] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.
- [25] Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. Active learning for cost-sensitive classification. In *International Conference on Machine Learning*, pages 1915–1924. PMLR, 2017.

- 394 [26] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes  
395 overconfidence in relu networks. In *International Conference on Machine Learning*, pages  
396 5436–5446. PMLR, 2020.
- 397 [27] Jiayu Lin. On the dirichlet distribution. *Master’s Thesis*, 2016.
- 398 [28] Andrea Locatelli, Alexandra Carpentier, and Samory Kpotufe. An adaptive strategy for active  
399 learning with smooth decision boundary. In *Algorithmic Learning Theory*, pages 547–571.  
400 PMLR, 2018.
- 401 [29] J. McFadden. The entropy of a point process. *Journal of the Society for Industrial & Applied*  
402 *Mathematics*, 13(4):988–994, 1965.
- 403 [30] F. Papangelou. On the entropy rate of stationary point processes and its discrete approximation.  
404 *Probability Theory and Related Fields*, 44(3):191–211, 1978.
- 405 [31] Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through  
406 abstention. In *Conference on Learning Theory*, pages 3806–3832. PMLR, 2021.
- 407 [32] Shubhanshu Shekhar, Mohammad Ghavamzadeh, and Tara Javidi. Active learning for classifi-  
408 cation with abstention. *IEEE Journal on Selected Areas in Information Theory*, 2(2):705–719,  
409 2021.
- 410 [33] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram,  
411 Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep  
412 neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR,  
413 2015.
- 414 [34] Wei Tan, Lan Du, and Wray Buntine. Diversity enhanced active learning with strictly proper  
415 scoring rules. *Advances in Neural Information Processing Systems*, 34:10906–10918, 2021.
- 416 [35] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.
- 417 [36] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel  
418 learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.
- 419 [37] Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning.  
420 *Advances in Neural Information Processing Systems*, 27, 2014.
- 421 [38] Guang Zhao, Edward Dougherty, Byung-Jun Yoon, Francis Alexander, and Xiaoning Qian.  
422 Uncertainty-aware active learning for optimal bayesian classifier. In *International Conference*  
423 *on Learning Representations (ICLR 2021)*, 2021.
- 424 [39] Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *arXiv preprint*  
425 *arXiv:2204.00043*, 2022.