

# Improving Evidential Deep Learning via Multi-Task Learning

Dongpin Oh,<sup>1</sup> Bonggun Shin<sup>2</sup>

<sup>1</sup>Deargen Inc., Seoul, South Korea

<sup>2</sup>Deargen USA Inc., Atlanta, GA

<sup>1</sup>dongpin.oh@deargen.me, <sup>2</sup>bonggun.shin@deargen.me

## Abstract

The Evidential regression network (ENet) estimates a continuous target and its predictive uncertainty without costly Bayesian model averaging. However, it is possible that the target is inaccurately predicted due to the gradient shrinkage problem of the original loss function of the ENet, the negative log marginal likelihood (NLL) loss. In this paper, the objective is to improve the prediction accuracy of the ENet while maintaining its efficient uncertainty estimation by resolving the gradient shrinkage problem. A multi-task learning (MTL) framework, referred to as MT-ENet, is proposed to accomplish this aim. In the MTL, we define the Lipschitz modified mean squared error (MSE) loss function as another loss and add it to the existing NLL loss. The Lipschitz modified MSE loss is designed to mitigate the gradient conflict with the NLL loss by dynamically adjusting its Lipschitz constant. By doing so, the Lipschitz MSE loss does not disturb the uncertainty estimation of the NLL loss. The MT-ENet enhances the predictive accuracy of the ENet without losing uncertainty estimation capability on the synthetic dataset and real-world benchmarks, including drug-target affinity (DTA) regression. Furthermore, the MT-ENet shows remarkable calibration and out-of-distribution detection capability on the DTA benchmarks.

## Introduction

The essential task in the safety-critical deep learning systems is to quantify uncertainty (Gal 2016; Kendall and Gal 2017). The factors contributing to uncertainty can be classified into two types: irreducible observation noise (*aleatoric uncertainty*) and the uncertainty of model parameters (*epistemic uncertainty*) (Gal 2016; Guo et al. 2017). In particular, the difficulty and expense involved in representing the uncertainty of the model parameters make it challenging to quantify epistemic uncertainty.

Common approaches used to estimate epistemic uncertainty are Ensemble-based methods and Bayesian neural networks (BNNs) (Wilson and Izmailov 2020). Ensemble-based methods and BNNs have been shown to produce impressive results in terms of both accuracy and the robustness of uncertainty estimation (Welling and Teh 2011; Neal 2012; Blundell et al. 2015; Lakshminarayanan, Pritzel, and

Blundell 2017; Gal and Ghahramani 2016; Maddox et al. 2019). However, since multiple numbers of models are required in ensemble models, and expansive approximations are necessary in BNNs for the intractable posterior, neither of these methodologies can be considered cost-effective in real-world applications (Amini et al. 2020).

In contrast, the evidential neural network (ENet) is a cost-effective *deterministic* neural network, designed to accomplish uncertainty estimation by generating conjugate prior parameters as its outputs (Malinin and Gales 2018; Sensoy, Kaplan, and Kandemir 2018; Gurevich and Stuke 2020; Malinin et al. 2020). The ENet can model both epistemic and aleatoric uncertainty based on the uncertainty of the generated conjugate prior. The ability of the ENet to estimate uncertainty without an ensemble or an expensive posterior approximation is remarkable.

On the other hand, the fundamental goal of deep learning is to accomplish state-of-the-art predictive accuracy, not only to estimate uncertainty. Although the ENet architecture can achieve outstanding and practical uncertainty estimations, NLL loss (negative log marginal likelihood)—the original loss function of the ENet—may result in the high mean squared error (MSE) of the ENet’s prediction in a certain condition. Intuitively speaking, this problem is caused by the fact that the NLL loss could finish training by determining that the target values are unknown instead of correcting the prediction of the ENet. We show that this phenomenon occurs when the estimated epistemic uncertainty is high. This high epistemic uncertainty implies that the given training samples are sparse or biased (Gal 2016), which is a common situation in real-world applications, such as drug-target affinity prediction tasks (Ezzat et al. 2016; Yang et al. 2021).

One possible solution to resolve this issue is reformulating the training objective of the ENet as the Multi-task learning (MTL) loss with an additional loss function which only optimizes the predictive accuracy, such as the MSE loss. However, in the MTL, the conflicting gradients problem could occur, which negatively impact performance (Sener and Koltun 2018; Lin et al. 2019; Yu et al. 2020). In particular, for the safety-critical systems, it can be harmful if the uncertainty estimation capability is degraded despite improved accuracy due to the gradient conflict between the losses. Therefore, to identify and determine the reason for

the gradient conflict in our MTL optimization, the gradient between the loss functions was analyzed. Our analysis shows the certain condition that this gradient conflict could occur.

Based on this analysis, we define the Lipschitz modified MSE loss, which is designed to mitigate the gradient conflict with the original NLL loss function of the ENet. We also propose an MTL framework using the Lipschitz MSE, named Multi-task-based evidential network (MT-ENet)<sup>1</sup>. In particular, our contributions are as follows:

- We thoroughly show that (1) the NLL loss alone is not sufficient to optimize the prediction accuracy of the ENet, (2) and adding the MSE-based loss into the training objective can resolve this issue.
- We establish the condition that the MSE-based loss function could conflict with the NLL loss. To avoid this gradient condition, the Lipschitz modified MSE loss function is designed. Based on our novel Lipschitz loss function, we propose an MTL framework, MT-ENet, that improves the prediction accuracy of the ENet while maintaining its uncertainty estimation capability.
- Our experiments show that MT-ENet outperforms other strong baselines on the real-word regression benchmark datasets. And the MT-ENet shows remarkable performance in uncertainty estimation and out-of-distribution detection on the drug-target affinity datasets.

## Background

### Problem Setup

Assume we have a dataset of the regression task,  $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^N$ , where  $X_i \in \mathbb{R}^d$  are the i.i.d. input data points,  $d$  is the dimension of an input vector,  $y_i \in \mathbb{R}$  are the real-valued targets, and  $N$  is the number of data samples. We consider resolving a regression task by modeling a predictive distribution,  $p(y|f_\theta(X))$ , where  $f$  is a neural network and  $\theta$  is its parameters.

### Evidential Regression Network

An evidential regression network (ENet) (Amini et al. 2020) considers a target value,  $y$ , as a sample drawn from a normal distribution with unknown parameters,  $\mu, \sigma$ . These parameters,  $\mu$  and  $\sigma$ , are drawn from a Normal-Inverse-Gamma (NIG) distribution, which is the conjugate prior to the normal distribution:

$$y \sim \mathcal{N}(\mu, \sigma^2), \quad \mu \sim \mathcal{N}(\gamma, \frac{\sigma^2}{\nu}), \quad \sigma^2 \sim \text{Gamma}^{-1}(\alpha, \beta) \quad (1)$$

where  $\gamma \in \mathbb{R}, \nu > 0, \alpha > 1, \beta > 0$ , and  $\text{Gamma}^{-1}$  is the inverse-gamma distribution. The NIG distribution in Eq 1. is parameterized by  $\mathbf{m} = \{\gamma, \nu, \alpha, \beta\}$ , which is the output of the ENet,  $\mathbf{m} = f_\theta(X)$ , where  $\theta$  is a trainable parameter of the ENet (Fig 1).

With the NIG distribution in Eq 1, the model prediction ( $\mathbb{E}[\mu]$ ), aleatoric ( $\mathbb{E}[\sigma^2]$ ), and epistemic ( $\text{Var}[\mu]$ ) uncertainty of the ENet can be calculated by the following:

$$\mathbb{E}[\mu] = \gamma, \quad \mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}, \quad \text{Var}[\mu] = \frac{\beta}{\nu(\alpha - 1)}. \quad (2)$$

<sup>1</sup>Code is at <https://github.com/deargen/MT-ENet>.

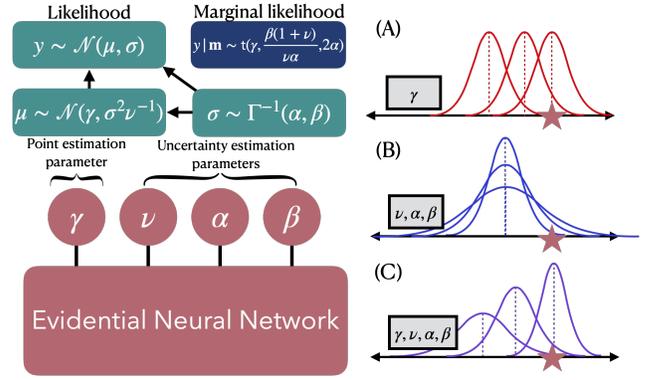


Figure 1: A scheme of the architecture of the MT-ENet and ENet and simplified examples of roles of the outputs ( $\mathbf{m} = \{\gamma, \nu, \alpha, \beta\}$ ). (Left) An overview of the ENet architecture. The likelihood and marginal likelihood distribution can be calculated via the outputs of the ENet. (A)  $\gamma$  determines the model prediction (point estimation); (B)  $\nu, \alpha, \beta$  determine the model uncertainty (uncertainty estimation); (C) The predictive distribution (Marginal likelihood) of the ENet can be derived by  $\gamma, \nu, \alpha, \beta$ .

With these equations, we define **point estimation** as estimating the model prediction of the ENet,  $\mathbb{E}[\mu] = \gamma$  and **uncertainty estimation** as estimating the uncertainties,  $\mathbb{E}[\sigma^2]$  and  $\text{Var}[\mu]$ .

In addition to the uncertainty estimates in Eq 2, **pseudo observation** can be used as an alternative interpretation of the predictive uncertainty,  $\mathbb{E}[\sigma^2]$  and  $\text{Var}[\mu]$  (Murphy 2007, 2012). This is widely used in Bayesian statistics literatures (Lee 1989).

**Definition 1.** We define  $\alpha$  and  $\nu$ , the outputs of the ENet, as the **pseudo observations**.

Note that the aleatoric and the epistemic uncertainty increase as the pseudo observations decrease because they are inversely proportional ( $\mathbb{E}[\sigma^2] \propto \frac{1}{\alpha}$ ,  $\text{Var}[\mu] \propto \frac{1}{\alpha\nu}$ ).

### Training the Enet With the Marginal Likelihood

The ENet learns its trainable parameters  $\theta$  by maximizing a *marginal likelihood* with respect to unknown Gaussian parameters,  $\mu$  and  $\sigma$ . By analytically marginalizing the NIG distribution (Eq 1) over these parameters  $\mu$  and  $\sigma$ , we can derive the following marginal likelihood:

$$\begin{aligned} p(y|\mathbf{m}) &= \int_{\sigma^2=0}^{\sigma^2=\infty} \int_{\mu=-\infty}^{\mu=\infty} p(y|\mu, \sigma^2) p(\mu, \sigma^2|\mathbf{m}) d\mu d\sigma^2 \\ &= \text{t}(y; \gamma, \frac{\beta(1+\nu)}{\nu\alpha}; 2\alpha) \end{aligned} \quad (3)$$

where  $\text{t}(x; l, s, n)$  is the student-t distribution with the location parameter ( $l$ ), the scale parameter ( $s$ ), and the degrees of freedom ( $n$ ).

The training procedure for the ENet is to minimize the negative log marginal likelihood (NLL) loss function,

$L_{NLL}(y, \mathbf{m}) = -\log(p(y|\mathbf{m}))$ , as summarized in the following equation:

$$\begin{aligned} \arg \min_{\theta} (L_{NLL}(y, \mathbf{m})) &= \arg \min_{\theta} \frac{1}{2} \log\left(\frac{\pi}{\nu}\right) - \alpha \log \Lambda \\ &+ \left(\alpha + \frac{1}{2}\right) \log((y - \gamma)^2 \nu + \Lambda) + \log\left(\frac{\Gamma(\alpha)}{\Gamma(\alpha + \frac{1}{2})}\right) \end{aligned} \quad (4)$$

where  $\Gamma(\cdot)$  is the gamma function and  $\Lambda = 2\beta(1 + \nu)$ . Besides allowing the ENet to predict a proper point estimate  $\gamma$ , the NLL loss function allows the ENet to quantify the aleatoric and epistemic uncertainty via Eq 4.

## Gradient Shrinking of the NLL Loss

In this section we show that the NLL loss alone is not sufficient to optimize the prediction accuracy of the ENet. Even though the model trained on NLL loss can predict the proper point estimate,  $\gamma$ , the model can circumvent to achieve lower NLL loss by increasing the predictive uncertainty **instead of achieving a higher predictive accuracy**. This limitation can be resolved by using an additional loss function. To formally state this, we first define the gradient vector of the loss function:

**Definition 2.** Consider a loss function  $L : \mathbb{R}^n \rightarrow \mathbb{R}$  and a set of outputs of a model,  $f_{\theta}(X) \in \mathbb{R}^n$ , with parameters  $\theta$ . Let  $\Omega_{\theta}$  be the subset of the model outputs,  $\Omega_{\theta} \subseteq f_{\theta}(X)$ , then  $\mathbf{g}_{L, \Omega} = \sum_{\omega_{\theta} \in \Omega_{\theta}} \frac{\partial L}{\partial \omega_{\theta}} \nabla_{\theta} \omega_{\theta}$  denotes the **gradient vector** with respect to the subset of the model outputs,  $\Omega_{\theta}$ . We exclude the reliance of  $\theta$  to simplify notations.

In order to increase the accuracy of the ENet, the model prediction ( $\gamma$ ) should be trained to get closer to the true value via the corresponding gradient  $\mathbf{g}_{L_{NLL}, \gamma}$ . For the ideal loss function, its gradient magnitude becomes zero only when the model prediction and the true value are identical. However, **despite an inaccurate prediction** if the prediction uncertainty is severely increased by the NLL loss, the gradient for the model prediction ( $\mathbf{g}_{L_{NLL}, \gamma}$ ) could be significantly small. Thus, we aim to verify that the gradient magnitude for the model prediction,  $\|\mathbf{g}_{L_{NLL}, \gamma}\|$ , is insignificant despite of the incorrect model prediction  $\gamma$ . Specifically, we prove that the  $\|\mathbf{g}_{L_{NLL}, \gamma}\|$  is converged to zero under the condition that the pseudo observation ( $\nu$ ) becomes zero as shown in *Theorem 1*:

**Theorem 1.** (Shrinking NLL gradient)<sup>2</sup>. Let  $\nu > 0, \gamma \in \mathbb{R}$  be the output of the ENet, and assume that  $(y - \gamma)^2 > 0$ , then for every real  $\epsilon > 0$ , there exists a real  $\delta > 0$  such that for every  $\nu$ ,  $0 < |\nu| < \delta$  implies  $|\frac{\partial L_{NLL}}{\partial \gamma}| < \epsilon$ . Therefore,  $\lim_{\nu \rightarrow 0^+} \frac{\partial}{\partial \gamma} L_{NLL}(y, \mathbf{m}) = 0 \Rightarrow \lim_{\nu \rightarrow 0^+} \|\mathbf{g}_{L_{NLL}, \gamma}\| = 0$ .

*Theorem 1* states that the NLL loss itself is insufficient to correct the point estimate ( $\gamma$ ) because it cannot fully utilize the error value,  $(y - \gamma)^2$ , when pseudo observation ( $\nu$ )

<sup>2</sup>Proofs and details of all the theorems and propositions of this paper are given in Appendix A.



Figure 2: Simplified examples of the gradient vectors. The red arrows represent the gradient of MSE; The blue arrows characterize the point estimation gradient of NLL; The green arrows signify the uncertainty estimation gradient of NLL. (a) Non-conflicting gradients. (b) Slightly conflicting gradients. (c) Conflicting gradient by  $\mathbf{g}_{L_{NLL}, \{\nu, \alpha, \beta\}}$ . As *Proposition 1* states, the direction of  $\mathbf{g}_{L_{MSE}, \gamma}$  and  $\mathbf{g}_{L_{NLL}, \gamma}$  is always the same.

is very low. This infinitely low pseudo observation signifies that infinitely high epistemic uncertainty of the ENet ( $Var[\mu]$ ) since  $Var[\mu] \propto \frac{1}{\nu}$ . This statement is empirically justified in the experiment on the synthetic dataset as shown in Fig 4.

On the other hand, the simple MSE loss,  $L_{MSE}(y, \mathbf{m}) = (y - \gamma)^2$ , consistently trains the ENet to correctly estimate  $\gamma$ , since  $\frac{\partial}{\partial \gamma} L_{MSE}$  is independent of  $\nu$ . Thus, the MSE-based loss functions—which only updates the target value ( $\gamma$ )—could improve the point estimation capability of the ENet.

## Gradient Conflict Between the NLL and MSE

The previous section established that the simple MSE loss could resolve the gradient shrinkage problem. This finding motivates us to integrate the MSE loss into the NLL loss to form a MTL framework in order to improve the predictive accuracy. When forming this MTL framework, it is important not to have **gradient conflict** between tasks because they could lead to performance degradation. In the MTL literature, this undesirable case can be defined as when gradient vectors are in different directions (Sener and Koltun 2018; Yu et al. 2020).

To avoid this degradation, it is necessary to identify the cause of the gradient conflict. In this section, the gradient conflict between the simple MSE and NLL loss function is considered. To do this, the gradient of NLL is considered by dividing it into two separate gradients: uncertainty estimation gradient and point estimation gradient. Here, the *uncertainty* and *point estimation gradient* of the NLL are clarified with the following definition:

**Definition 3.** We define the *point estimation gradient* of the NLL as  $\mathbf{g}_{L_{NLL}, \gamma}$ , since  $\gamma$  performs the point estimation. The *uncertainty estimation gradient* of the NLL is defined as  $\mathbf{g}_{L_{NLL}, \{\nu, \alpha, \beta\}}$ , since  $\{\nu, \alpha, \beta\}$  perform the uncertainty estimation. Finally, we define the *total gradient* of the NLL as  $\mathbf{g}_{L_{NLL}, \mathbf{m}}$ .

Then, we show that the *point estimation gradient* of the NLL from the Definition 3 never conflicts with the gradient of the MSE as shown in the following proposition:

**Proposition 1.** (Non-conflicting point estimation)<sup>1</sup>. If gradient magnitudes of  $\mathbf{g}_{L_{MSE}, \gamma}$ ,  $\mathbf{g}_{L_{NLL}, \gamma}$  are not zero, then the cosine similarity  $s(\cdot)$  between  $\mathbf{g}_{L_{MSE}, \gamma}$  and

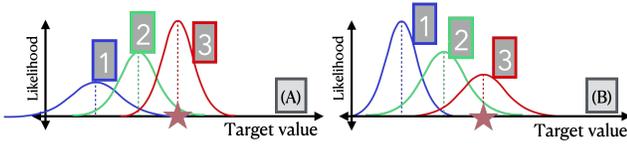


Figure 3: Illustration of the trained predictive distribution via the MSE and NLL losses. The dashed lines represent the model predictions ( $\mathbb{E}[\mu] = \gamma$ ); The stars are the ground truth. (A) The NLL decreases the uncertainty; (B) The NLL increases the uncertainty. The optimization procedure is (1)→(2)→(3). The NLL loss of (B) is higher than A due to the gradient conflict between the MSE and NLL.

$\mathbf{g}_{L_{NLL},\gamma}$  is always one:  $s(\mathbf{g}_{L_{MSE},\gamma}, \mathbf{g}_{L_{NLL},\gamma}) = 1$ .

*Proposition 1* states that providing the model updates its parameters only through  $\mathbf{g}_{L_{MSE},\gamma}$  and  $\mathbf{g}_{L_{NLL},\gamma}$ , the two gradients will never conflict. This is indicated by the red arrows and blue arrows in Fig 2. Then, a trivial consequence of the *Proposition 1* is the following corollary:

**Corollary 1.** *If the total gradients of the NLL and the gradient of MSE are not in the same direction, then the uncertainty estimation gradients of the NLL and the gradients of the MSE are also not in the same direction:  $s(\mathbf{g}_{L_{MSE},\mathbf{m}}, \mathbf{g}_{L_{NLL},\mathbf{m}}) < 1 \Rightarrow s(\mathbf{g}_{L_{MSE},\mathbf{m}}, \mathbf{g}_{L_{NLL},\{\nu,\alpha,\beta\}}) < 1$ .*

Since *Proposition 1* states that there is no gradient conflict between  $\mathbf{g}_{L_{MSE},\gamma}$  and  $\mathbf{g}_{L_{NLL},\gamma}$ , *Corollary 1* implies that the **only cause of gradient conflict** between the MSE and NLL loss is the **uncertainty estimation gradient** of the NLL. Therefore, if the gradient conflict between the uncertainty estimation gradient and the gradient of the MSE loss can be avoided, the MTL can be performed more effectively.

### Conjecture of the Gradient Conflict

In the previous section, it was shown that the *uncertainty estimation gradient* of the NLL is the only cause of the gradient conflict. In this section, we illustrate and focus on how the increased predictive uncertainty due to uncertainty estimation gradient can result in large gradient conflict. Fig 3 shows the examples of the predictive distributions of the ENet trained on the MSE and NLL loss during its training processes. According to the multi-task learning literature, it is a common premise that the more severe the gradient conflict, the greater the loss (Chen et al. 2018; Lin et al. 2019; Yu et al. 2020; Chen et al. 2020; Javaloy and Valera 2021). In this sense, since (B) has a higher loss than (A) as long as the model prediction is approaching its true value, we can consider that the gradient conflict of (B) is more significant.

The fundamental difference between (B) and (A) is that the NLL increases uncertainty in (B), and the NLL decreases uncertainty in (A). Therefore, **we assume that gradient conflicts between the MSE and NLL are more significant when the NLL increases the predictive uncertainty of the**

ENet like the case of (B). In practice, to alleviate the gradient conflict of (B), Lipschitz modified MSE loss function is designed. And we empirically demonstrate that our *Lipschitz modified MSE loss function* successfully reduces the gradient conflict as shown in Fig 5.

### Multi-Task Based Evidential Neural Network

A new multi-task loss function of the ENet is introduced in this section. The ultimate goal of this multi-task-based evidential neural network (MT-ENet) is to *improve the predictive accuracy* of the ENet *while does not interrupt the uncertainty estimation training on the NLL loss*. To accomplish this, the MT-ENet utilizes a novel Lipschitz loss function as an additional loss function which can mitigate gradient conflicts with the NLL loss.

#### Prevent the Conflict via the Lipschitz MSE Loss

In the previous section, we established that two gradients from the MSE and NLL could conflict when the NLL increases predictive uncertainty. In this section, this gradient conflict condition is specified to design the loss function that can resolve this conflict.

Specifically, we reinterpret this conflict situation as the change of **pseudo observations** of the ENet, since an increase in predictive uncertainty is associated with a decrease in the pseudo observation ( $\mathbb{E}[\sigma^2] \propto \frac{1}{\alpha}$ ,  $Var[\mu] \propto \frac{1}{\alpha\nu}$ ). This decrease in the pseudo observations by the NLL loss occurs when the difference between the model prediction and the true value exceeds specific thresholds, as shown in *Proposition 2*:

**Proposition 2.** *Let  $\lambda^2 \stackrel{def}{=} (y - \gamma)^2$ , which is the squared error value of the ENet. then if  $\lambda^2$  is larger than certain thresholds,  $U_\nu$  and  $U_\alpha$ , then the derivative signs of the  $L_{NLL}$  w.r.t  $\nu$  and  $\alpha$  are positive.*

$$\lambda^2 > U_\alpha \Rightarrow \frac{\partial}{\partial \alpha} L_{NLL} > 0, \quad \lambda^2 > U_\nu \Rightarrow \frac{\partial}{\partial \nu} L_{NLL} > 0$$

$$U_\nu = \frac{\beta(\nu+1)}{\alpha\nu}, \quad U_\alpha = \frac{2\beta(1+\nu)}{\nu} [\exp(\Psi(\alpha + \frac{1}{2}) - \Psi(\alpha)) - 1] \quad (5)$$

where  $\Psi(\cdot)$  is the digamma function.

Since both  $\frac{\partial L_{NLL}}{\partial \alpha}$  and  $\frac{\partial L_{NLL}}{\partial \nu}$  contain  $\lambda^2$ , the thresholds ( $U_\nu$  and  $U_\alpha$ ) can be obtained by rearranging  $\frac{\partial L_{NLL}}{\partial \alpha}$ ,  $\frac{\partial L_{NLL}}{\partial \nu}$  to solve for  $\lambda^2$  for each case (See Appendix A).

*Proposition 2* states that when the difference between the predicted and true values is significant, the NLL loss trains the ENet to increase the predictive uncertainty. This increase in predictive uncertainty results in the gradient conflict, which can lead to performance degradation, as explained in the previous section. Thus, our strategy is designing the additional loss function, which regulates its gradient if there is an increase in uncertainty.

**Lipschitz MSE loss** We propose a modified Lipschitz MSE loss based on *Proposition 2* to mitigate the gradient conflict that occurs when the MSE is excessively large. Unlike a commonly used MSE loss—which does not have Lipschitzness (Qi et al. 2020)—we define the  $L$ -Lipschitz con-

tinuous loss function where the Lipschitz constant  $L$  is dynamically determined by  $U_\alpha$  and  $U_\nu$  from Proposition 2.

Consider a minibatch,  $\mathbf{X} = \{(X_1, y_1), \dots, (X_N, y_N)\}$ , extracted from our dataset  $\mathcal{D}$  and the corresponding output of the MT-ENet,  $f_\theta(X_i) = \mathbf{m}_i = \{\gamma_i, \nu_i, \alpha_i, \beta_i\}$ . For each  $(X_i, y_i)$  and  $\mathbf{m}_i$ , the Lipschitz modified MSE  $L_{mse}^*$  is defined as follows:

$$L_{mse}^*(y_i, \gamma_i) = \begin{cases} (y_i - \gamma_i)^2, & \text{If } \lambda_i^2 < U_{\nu, \alpha} \\ 2\sqrt{U_{\nu, \alpha}}|y_i - \gamma_i| - U_{\nu, \alpha}, & \text{If } \lambda_i^2 \geq U_{\nu, \alpha} \end{cases} \quad (6)$$

where  $U_{\nu, \alpha} \stackrel{def}{=} \min_{i \in [1:N]} (\min(U_{\nu_i}, U_{\alpha_i}))$ . The  $L_{mse}^*$  restricts its gradient magnitude through adjusting its Lipschitz constant,  $U_{\alpha, \nu}$ , when the pseudo observation tend to be decreased (thus, uncertainty is increased) via the NLL loss. Note that the  $L_{mse}^*$  does not propagate gradients for the  $(\nu, \alpha, \beta)$ . We empirically demonstrate that the designed Lipschitz modified loss mitigates the gradient conflict with the NLL by illustrating the high cosine similarity between our Lipschitz loss and NLL loss, as shown in Fig 5.

**The Final Objective of the MT-ENet** The final MTL objective of the MT-ENet,  $\mathcal{L}_{total}$ , is defined as simple combination of the two losses and the regularization:

$$\mathcal{L}_{total} = \frac{1}{N} \sum_{i=1}^N L_{NLL}(y_i, \mathbf{m}_i) + L_{mse}^*(y_i, \gamma_i) + cL_R(y_i, \mathbf{m}_i) \quad (7)$$

where  $L_R(y, \mathbf{m}) = |y - \gamma|(2\nu + \alpha)$  is the evidence regularizer (Amini et al. 2020) and  $c > 0$  is its coefficient.

**Remark 1.** The evidential regularization,  $L_R$ , could also train the model prediction,  $\gamma$ , like the NLL and MSE. However, the main goal of  $L_R$  is not training  $\gamma$  but regulating the NLL loss by increasing the predictive uncertainty for incorrect predictions of the ENet (Sensoy, Kaplan, and Kandemir 2018; Amini et al. 2020). Therefore,  $L_R$  cannot play the role of the MSE based loss function, which improves predictive accuracy. Further discussion is available in Appendix B.

## Related Work

**Related works on multi-task learning** Multi-task learning (MTL) is a paradigm for training several tasks with a single model (Ruder 2017). A naive strategy of MTL is training a model with a linear combination of given loss functions:  $\sum_{i=0}^T c_i L_i$ . However, this linearly combined losses can be ineffective if the loss functions have trade-off relationships between other loss functions (Sener and Koltun 2018; Lin et al. 2019). Several studies have resolved this issue—in the architecture agnostic manner—via dynamically adjusting weights of losses (Sener and Koltun 2018; Guo et al. 2018; Kendall, Gal, and Cipolla 2018; Liu, Liang, and Gitter 2019; Lin et al. 2019) or modifying gradients (Chen et al. 2018, 2020; Yu et al. 2020). Similar to the previous works, we targeted resolving the gradient conflict issue. In particular, we mainly focused on applying the MTL to the ENet while keeping the NLL loss function to maintain the uncertainty estimation capability of the NLL loss function.

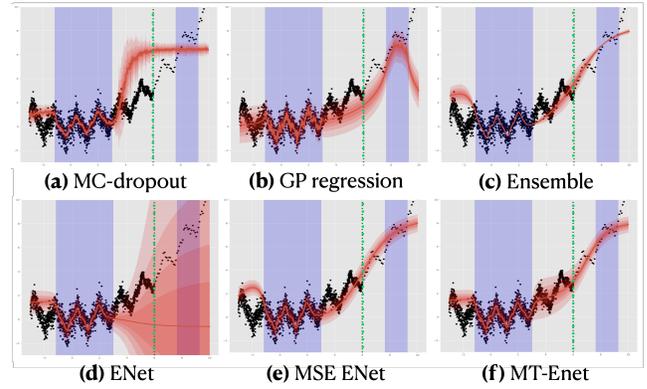


Figure 4: Synthetic dataset results of various models. Red lines represent the predictive mean; Red shades represent the predictive uncertainty; Blue highlighted regions represent our observations. The green lines separate the data-rich and data-sparse regions.

To accomplish this, we defined the Lipschitz modified MSE which reconciles with the uncertainty estimation by implicitly alleviating the gradient conflict.

## Experiments

### Synthetic Dataset Evaluation

We first qualitatively evaluate the MT-ENet using a synthetic regression dataset. Fig 4 represents our synthetic data and the model predictions. Experiments were carried out on an imbalanced regression dataset for two reasons: **1)** real-world regression tasks often involve an imbalanced target (Yang et al. 2021); **2)** the ENet could fail on the imbalanced dataset. We observe that the predictions of the ENet for sparse samples (to the right sides of the green dotted lines in Fig 4) do not fit the data despite the well-calibrated uncertainty. This trend of the ENet (Fig 4 (c)) is in line with Theorem 1, which stated that the MSE of the ENet is likely to be insufficient when the epistemic uncertainty is high. Conversely, the results obtained for other models, including the MT-ENet, are acceptable for the sparse sample regions. Training and experimental details are available in Appendix C.

### Performance Evaluation on Real-World Benchmarks

We evaluate the performance of the MT-ENet in comparison with strong baselines through the UCI regression benchmark datasets. The experimental settings and model architecture used in this study are identical to those used by (Hernández-Lobato and Adams 2015). We report the average root mean squared error (RMSE) and the negative log-likelihood (NLL) of the models in Table 1. The MT-ENet generally provides the best or comparable RMSE and NLL performances. This is the evidence that the MT-ENet and the Lipschitz modified MSE loss improve the predictive accuracy of the ENet without disturbing the original NLL loss function. Details of the experiment are given in Appendix C.

Datasets	RMSE			
	MC-Dropout	Ensemble	ENet	MT-ENet
<b>Boston</b>	<b>2.97(0.19)</b>	3.28(1.00)	3.06(0.16)	<b>3.04(0.21)</b>
<b>Concrete</b>	<b>5.23(0.12)</b>	6.03(0.58)	5.85(0.15)	<b>5.60(0.17)</b>
<b>Energy</b>	<b>1.66(0.04)</b>	2.09(0.29)	2.06(0.10)	<b>2.04(0.07)</b>
<b>Kin8nm</b>	0.10(0.00)	<b>0.09(0.00)</b>	<b>0.09(0.00)</b>	<b>0.08(0.00)</b>
<b>Navel</b>	0.01(0.00)	<b>0.00(0.00)</b>	<b>0.00(0.00)</b>	<b>0.00(0.00)</b>
<b>Power</b>	<b>4.02(0.04)</b>	4.11(0.17)	4.23(0.09)	<b>4.03(0.04)</b>
<b>Protein</b>	<b>4.36(0.01)</b>	4.71(0.06)	<b>4.64(0.03)</b>	4.73(0.07)
<b>Wine</b>	<b>0.62(0.01)</b>	0.64(0.04)	<b>0.61(0.02)</b>	0.63(0.01)
<b>Yacht</b>	<b>1.11(0.09)</b>	1.58(0.48)	1.57(0.56)	<b>1.03(0.08)</b>

Datasets	NLL			
	MC-Dropout	Ensemble	ENet	MT-ENet
<b>Boston</b>	2.46(0.06)	2.41(0.25)	<b>2.35(0.06)</b>	<b>2.31(0.04)</b>
<b>Concrete</b>	3.04(0.02)	3.06(0.18)	<b>3.01(0.02)</b>	<b>2.97(0.02)</b>
<b>Energy</b>	1.99(0.02)	<b>1.38(0.22)</b>	1.39(0.06)	<b>1.17(0.05)</b>
<b>Kin8nm</b>	-0.95(0.01)	<b>-1.20(0.02)</b>	<b>-1.24(0.01)</b>	-1.19(0.01)
<b>Navel</b>	-3.80(0.01)	-5.63(0.05)	<b>-5.73(0.07)</b>	<b>-5.96(0.03)</b>
<b>Power</b>	2.80(0.01)	<b>2.79(0.04)</b>	2.81(0.07)	<b>2.75(0.01)</b>
<b>Protein</b>	2.89(0.00)	2.83(0.02)	<b>2.63(0.00)</b>	<b>2.64(0.01)</b>
<b>Wine</b>	0.93(0.01)	0.94(0.12)	<b>0.89(0.05)</b>	<b>0.86(0.02)</b>
<b>Yacht</b>	1.55(0.03)	1.18(0.21)	<b>1.03(0.19)</b>	<b>0.78(0.06)</b>

Table 1: RMSE and NLL of UCI regression benchmark datasets. The best score and the second score are highlighted in bolded. Standard errors are reported in the parentheses.

### Drug-target Affinity Regression

Next, we evaluate the MT-ENet on a high dimensional complex drug-target affinity (DTA) regression, which is an essential task for early-stage drug discovery (Shin et al. 2019). Our experiments use two well-known benchmark datasets in the DTA literature: Davis (Davis et al. 2011) and Kiba (Tang et al. 2014). The Davis and Kiba datasets consist of protein sequence data and simplified molecular line-entry system (SMILES) sequences. The target value of the data is the drug-target affinity, which is a single real value.

The model architecture of the baselines and the MT-ENet, referred to as DeepDTA (Öztürk, Özgür, and Ozkirimli 2018), is the same except for the number of outputs: four outputs represent  $(\gamma, \nu, \alpha, \beta)$  for the ENet and MT-ENet; and a single output represents the target value for the MC-Dropout. The inputs of the models are the SMILES and protein sequences. The DeepDTA architecture consists of one-dimensional convolution layers for embedding the input sequences and fully connected layers to predict the target values using the embedded features. The details of training and the model architectures are available in Appendix C.

**Metrics** Our evaluation metrics are the MSE, NLL, ECE (Expected Calibration Error) and CI (Concordance Index). The MSE and NLL are typical losses in the optimizer. The role of the CI is to quantify the quality of rankings and the predictive accuracy of the model (Steck et al. 2008; Yu et al. 2011). The CI is defined as:

$$CI = \frac{1}{N} \sum_{y_i > y_j} h(\hat{y}_i > \hat{y}_j), \quad h(x) = \begin{cases} 1, & x > 0 \\ 0.5, & x = 0 \\ 0, & \text{else} \end{cases}$$

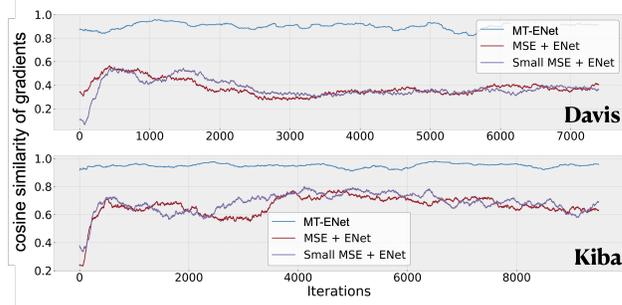


Figure 5: Cosine similarity trends between gradients from the additional loss and NLL. (Blue) the MT-ENet with the modified MSE; (Red) the ENet with the simple MSE; (Purple) the ENet with the weighted (0.1) MSE.

where  $N$  is the total number of data pairs;  $y_i$  is a ground truth, and  $\hat{y}_i$  is a predicted value. The ECE indicates how accurately the estimated uncertainty reflects real accuracy (Guo et al. 2017; Kuleshov, Fenner, and Ermon 2018). The ECE can be calculated as:

$$ECE = \frac{1}{N} \sum_{i=1}^N |acc(P_i) - P_i|$$

Where  $acc(P_i)$  is accuracy using  $P_i$  confidence interval (e.g.  $P_i = 0.95$ ) and  $N$  is the number of intervals. We use  $P_{[0;N]} = \{0.01, 0.02, \dots, 1.00\}$ ,  $N = 100$ .

**Alleviated gradient conflict by the Modified MSE** We first demonstrate that the Lipschitz modified MSE can alleviate the gradient conflict by measuring cosine similarities of the gradient vectors. Note that, in this part of study, we use the ENet with the simple MSE as the additional loss (MSE ENet) as the baseline to determine whether our proposed Lipschitz loss can alleviate the conflict.

The moving average of the cosine similarity between gradients is shown in Fig 5, for 500 iterations averaged. The trends of the MT-ENet indicate the high cosine similarities, which is evidence that the Lipschitz modified MSE successfully alleviates the gradient conflict. Also, the results show that simply adjusting the weight of the MSE loss (the purple lines, small MSE) fails to resolve the conflict problem. Since the Lipschitz MSE is designed to avoid the conflict when the NLL loss results in the high predictive uncertainty high ( $B$  in Fig 3), this experimental result is in good agreement with our conjecture regarding the MT-ENet gradient conflict: In the event that the NLL increases the predictive uncertainty of the model, then the gradient conflict is highly likely.

**Performance evaluation** Table 2 represents the performance evaluation results of the Kiba and Davis datasets. The MT-ENet and MSE ENet successfully improve the predictive accuracy metrics (CI, MSE) for both datasets. This result confirms that the additional MSE-based loss functions can improve the point-estimation capability of the ENet as described in Sec . Note also that the MT-ENet enhances accuracy without any significant degradation of the uncertainty estimation capability, which is indicated by the NLL and

Davis				
	ENet	MT-ENet	MSE ENet	Dropout
CI	0.856(0.02)	<b>0.864(0.01)</b>	0.863(0.02)	<b>0.884(0.00)</b>
MSE	0.275(0.00)	0.273(0.01)	<b>0.266(0.01)</b>	<b>0.248(0.01)</b>
NLL	-2.344(0.42)	<b>-2.424(0.07)</b>	<b>-2.430(0.10)</b>	0.633(0.02)
ECE	0.184(0.02)	<b>0.156(0.03)</b>	<b>0.179(0.02)</b>	0.217(0.01)
Kiba				
	ENet	MT-ENet	MSE ENet	Dropout
CI	0.885(0.00)	<b>0.887(0.00)</b>	<b>0.888(0.00)</b>	0.872(0.00)
MSE	0.190(0.00)	0.181(0.00)	<b>0.176(0.00)</b>	<b>0.178(0.00)</b>
NLL	<b>-1.544(0.05)</b>	<b>-1.433(0.07)</b>	-1.276(0.04)	0.465(0.01)
ECE	<b>0.077(0.03)</b>	<b>0.066(0.01)</b>	0.081(0.02)	0.162(0.01)
BindingDB				
	ENet	MT-ENet	MSE ENet	Dropout
CI	0.822(0.00)	0.824(0.00)	<b>0.831(0.00)</b>	<b>0.830(0.00)</b>
MSE	0.704(0.01)	0.694(0.00)	<b>0.637(0.01)</b>	<b>0.641(0.01)</b>
NLL	<b>0.944(0.02)</b>	<b>0.937(0.01)</b>	0.972(0.01)	1.90(0.05)
ECE	<b>0.018(0.01)</b>	<b>0.015(0.00)</b>	0.027(0.01)	0.134(0.01)

Table 2: The performance evaluation results on the DTA benchmark datasets. *MSE ENet* represents the ENet using the simple MSE as the additional loss. *Dropout* represents the MC-Dropout. Standard deviations are reported in the parentheses.

ECE. Notably, in the case of the calibration metric (ECE), the MT-ENet shows the best results among the existing models. Conversely, the NLL and ECE of the MSE-ENet on the Kiba dataset are degraded to some extent. This sacrifice of the uncertainty estimation ability of the MSE-ENet can likely be attributed to the gradient conflict with the NLL loss function.

### Out-Of-Distribution Testing on the BindingDB Dataset

We examine the out-of-distribution (OOD) detection capability of the MT-ENet on the curated BindingDB dataset (Liu et al. 2007). The BindingDB dataset includes various drug-target (molecule-protein) sequence pairs. Because we excluded kinase protein samples—which are the target of the Kiba dataset—pairs of the kinase proteins from the Kiba dataset and detergent molecules from the PubChem (Kim et al. 2019) are considered as the OOD data. The test dataset of the BindingDB is considered as the in-distribution (ID) data.

For a total of three times, we randomly split the BindingDB dataset into the training (80%), validation (10%), and test (10%) datasets. All examined models are trained three times with the training datasets. The details of the experiments and these datasets are available in Appendix B.

Fig 6 represents the distributions of log epistemic uncertainty and the AUC-ROC score of the examined models. The MT-ENet and ENet show high AUC-ROC scores, which implies excellent detection capability for OOD samples. In addition, Table 2 reports the performance for the test dataset of the BindingDB. Table 2 and Fig 6 show that the MT-ENet improves the ENet in all the metrics on the BindingDB dataset, including the calibration and OOD detection. In contrast, the AUC-ROC and ECE of the MSE-ENet considerably underperform in comparison to other methods,

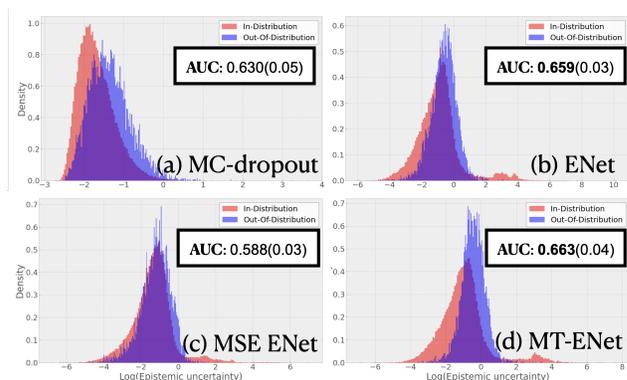


Figure 6: Density plots of log epistemic uncertainty. The AUC-ROC is also reported. The red distributions represent the ID data; The blue distributions represent the OOD data. *MSE ENet* represents the ENet with the simple MSE as the additional loss.

though the MSE-ENet shows the best MSE and CI on the BindingDB dataset. The exception is that the ECE and NLL of the MSE-ENet are better than that of the MC-dropout. This result is evidence that an additional loss function (the simple MSE) for the MSE ENet, which does not avoid the gradient conflict, can result in an overconfidence problem since it could disturb the uncertainty estimation of the NLL loss.

Furthermore, the density plots of the aleatoric uncertainty of the MSE-ENet, ENet, and MT-ENet are provided in Appendix D. The results demonstrate that the aleatoric uncertainties are at a similar level for the OOD and ID data. These results of the experiment are evidence that the MT-ENet is capable of disentangling the aleatoric and the epistemic uncertainties.

## Conclusion

We have theoretically and empirically demonstrated that the MSE-based loss improves the point estimation capability of the ENet by resolving the gradient shrinkage problem. We proposed the MT-ENet which uses the Lipschitz modified MSE as the additional training objectives to ensure that the uncertainty estimation of the NLL is not disturbed. Our experiments show that the MT-ENet improves predictive accuracy not only maintaining but also sometimes enhance the uncertainty estimation of the ENet. Successful real-world deep learning systems must be *accurate*, *reliable*, and *efficient*. We look forward to improving the evidential deep learning—that is, *efficient* and *reliable*—with better *accuracy* through the MTL framework.

## Acknowledgments

This research was results of a study on the *HPC Support Project*, supported by the *Ministry of Science and ICT* and *NIPA (National IT Industry Promotion Agency - Republic of Korea)*.

## References

- Amini, A.; Schwarting, W.; Soleimany, A.; and Rus, D. 2020. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33.
- Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 1613–1622. PMLR.
- Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, 794–803. PMLR.
- Chen, Z.; Ngiam, J.; Huang, Y.; Luong, T.; Kretzschmar, H.; Chai, Y.; and Anguelov, D. 2020. Just Pick a Sign: Optimizing Deep Multitask Models with Gradient Sign Dropout. *arXiv preprint arXiv:2010.06808*.
- Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; and Zarrinkar, P. P. 2011. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11): 1046–1051.
- Ezzat, A.; Wu, M.; Li, X.-L.; and Kwok, C.-K. 2016. Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC bioinformatics*, 17(19): 267–276.
- Gal, Y. 2016. Uncertainty in deep learning. *University of Cambridge*, 1(3): 4.
- Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.
- Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330. PMLR.
- Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 270–287.
- Gurevich, P.; and Stuke, H. 2020. Gradient conjugate priors and multi-layer neural networks. *Artificial Intelligence*, 278: 103184.
- Hernández-Lobato, J. M.; and Adams, R. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, 1861–1869. PMLR.
- Javaloy, A.; and Valera, I. 2021. Rotograd: Dynamic Gradient Homogenization for Multi-Task Learning. *arXiv preprint arXiv:2103.02631*.
- Kendall, A.; and Gal, Y. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems*, 30: 5574–5584.
- Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.
- Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; et al. 2019. PubChem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1): D1102–D1109.
- Kuleshov, V.; Fenner, N.; and Ermon, S. 2018. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2796–2804. PMLR.
- Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.
- Lee, P. M. 1989. *Bayesian statistics*. Oxford University Press London:.
- Lin, X.; Zhen, H.-L.; Li, Z.; Zhang, Q.; and Kwong, S. 2019. Pareto Multi-Task Learning. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Liu, S.; Liang, Y.; and Gitter, A. 2019. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33-01, 9977–9978.
- Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; and Gilson, M. K. 2007. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl\_1): D198–D201.
- Maddox, W. J.; Izmailov, P.; Garipov, T.; Vetrov, D. P.; and Wilson, A. G. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32: 13153–13164.
- Malinin, A.; Chervontsev, S.; Provilkov, I.; and Gales, M. 2020. Regression Prior Networks. *arXiv preprint arXiv:2006.11590*.
- Malinin, A.; and Gales, M. 2018. Predictive Uncertainty Estimation via Prior Networks. In *NIPS’18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, 7047–7058. Curran Associates, Inc.
- Murphy, K. P. 2007. Conjugate Bayesian analysis of the Gaussian distribution. *Technical report*.
- Murphy, K. P. 2012. *Machine learning: a probabilistic perspective*. MIT press.
- Neal, R. M. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.
- Öztürk, H.; Özgür, A.; and Ozkirimli, E. 2018. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17): i821–i829.
- Qi, J.; Du, J.; Siniscalchi, S. M.; Ma, X.; and Lee, C.-H. 2020. On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, 27: 1485–1489.
- Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 525–536.

- Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 3183–3193.
- Shin, B.; Park, S.; Kang, K.; and Ho, J. C. 2019. Self-attention based molecule representation for predicting drug-target interaction. In *Machine Learning for Healthcare Conference*, 230–248. PMLR.
- Steck, H.; Krishnapuram, B.; Dehing-Oberije, C.; Lambin, P.; and Raykar, V. C. 2008. On ranking in survival analysis: Bounds on the concordance index. In *Advances in neural information processing systems*, 1209–1216. Citeseer.
- Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; and Aittokallio, T. 2014. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3): 735–743.
- Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.
- Wilson, A. G.; and Izmailov, P. 2020. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*.
- Yang, Y.; Zha, K.; Chen, Y.-C.; Wang, H.; and Katabi, D. 2021. Delving into Deep Imbalanced Regression. *arXiv preprint arXiv:2102.09554*.
- Yu, C.-N.; Greiner, R.; Lin, H.-C.; and Baracos, V. 2011. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. *Advances in Neural Information Processing Systems*, 24: 1845–1853.
- Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. *Advances in Neural Information Processing Systems*, 33.