Top-Ambiguity Samples Matter: Understanding Why Deep Ensemble Works in Selective Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Selective classification allows a machine learning model to reject some hard inputs and thus improve the reliability of its predictions. In this area, the ensemble method is powerful in practice, but there has been no solid analysis on why the ensemble method works. Inspired by an interesting empirical result that the improvement of the ensemble largely comes from *top-ambiguity* samples where its member models diverge, we prove that, based on some assumptions, the ensemble has a lower selective risk than the member model for any coverage within a range. The proof is nontrivial since the selective risk is a non-convex function of the model prediction. The assumptions and the theoretical results are supported by systematic experiments on both computer vision and natural language processing tasks.

A Proofs

A.1 Proof of Lemma 1

The proof of the second inequality in Lemma 1, i.e.,

$$P(\kappa_E(x) \geq t | a = 1) \leq M^{M-1} B (1-t)^M, \tag{A.1}$$

can be reduced to the proof of the first inequality, i.e.,

$$p(\kappa_E(x) = t | a = 1) \leq M^M B (1-t)^{M-1}. \tag{A.2}$$

The reason is as follows. If (A.2) holds, then we have

$$\begin{aligned} P(\kappa_E(x) \geq t | a = 1) &= \int_t^1 p(\kappa_E(x) = t' | a = 1) dt' \\ &\leq \int_t^1 M^M B (1-t')^{M-1} \\ &= M^{M-1} B (1-t)^M, \end{aligned}$$

which directly derives (A.1). Therefore, we only need to show (A.2) at following.

Firstly, we derive the PDF of the average of multiple continuous random variables expressed by the PDFs of these random variables (Lemma A.1), which helps us to analyze the PDF of the ensemble's predictive probabilities.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Lemma A.1. *Let* $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_M$ *be* M *continuous random variables, and their average is* $\mathrm{X}_{\mathrm{avg}} := \frac{1}{M}\sum_{i=1}^{M}\mathrm{X}_i$*. Then the PDF of* $\mathrm{X}_{\mathrm{avg}}$[1] *is*

$$p_{\mathrm{X}_{\mathrm{avg}}}(x_{\mathrm{avg}}) = M\int_{\mathbb{R}^{M-1}} \mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_{M-1}\cdot p_{\vec{\mathrm{X}}}(x_1, x_2, \ldots, x_{M-1}, Mx_{\mathrm{avg}} - \sum_{i=1}^{M-1} x_i), \quad \text{(A.3)}$$

where $p_{\vec{\mathrm{X}}}$ *is* $p_{\mathrm{X}_1,\mathrm{X}_2,\ldots,\mathrm{X}_M}$ *for short.*

Proof. The distribution function of $\mathrm{X}_{\mathrm{avg}}$ is

$$\begin{aligned} F_{\mathrm{X}_{\mathrm{avg}}}(x_{\mathrm{avg}}) &= \int_{\sum_i x_i \le Mx_{\mathrm{avg}}} \mathrm{d}x_1\cdots\mathrm{d}x_{M-1}\mathrm{d}x_M\cdot p_{\vec{\mathrm{X}}}(x_1,\ldots,x_M) \\ &= \int_{\mathbb{R}^{M-1}} \mathrm{d}x_1\cdots\mathrm{d}x_{M-1}\int_{-\infty}^{Mx_{\mathrm{avg}}-\sum_{i=1}^{M-1}x_i} \mathrm{d}x_M\cdot p_{\vec{\mathrm{X}}}(x_1,\cdots,x_M). \end{aligned}$$

Let $x_M = u - \sum_{i=1}^{M-1} x_i$, then the integral above is equal to

$$\begin{aligned} &\int_{\mathbb{R}^{M-1}} \mathrm{d}x_1\cdots\mathrm{d}x_{M-1}\int_{-\infty}^{Mx_{\mathrm{avg}}} \mathrm{d}u\cdot p_{\vec{\mathrm{X}}}(x_1,\ldots,x_{M-1},u-\sum_{i=1}^{M-1}x_i) \\ =&\int_{-\infty}^{Mx_{\mathrm{avg}}}\mathrm{d}u\int_{\mathbb{R}^{M-1}}\mathrm{d}x_1\cdots\mathrm{d}x_{M-1}\cdot p_{\vec{\mathrm{X}}}(x_1,\ldots,x_{M-1},u-\sum_{i=1}^{M-1}x_i). \end{aligned} \quad \text{(A.4)}$$

The PDF of $\mathrm{X}_{\mathrm{avg}}$ is the derivative of $F_{\mathrm{X}_{\mathrm{avg}}}$, which, combined with (,A.4) derives

$$\begin{aligned} p_{\mathrm{X}_{\mathrm{avg}}}(x_{\mathrm{avg}}) =& F'_{\mathrm{X}_{\mathrm{avg}}}(x_{\mathrm{avg}}) \\ =& \frac{\mathrm{d}(Mx_{\mathrm{avg}})}{\mathrm{d}x_{\mathrm{avg}}}\cdot\frac{\mathrm{d}F_{\mathrm{X}_{\mathrm{avg}}}}{\mathrm{d}(Mx_{\mathrm{avg}})} \\ =& M\int_{\mathbb{R}^{M-1}}\mathrm{d}x_1\cdots\mathrm{d}x_{M-1}\cdot p_{\vec{\mathrm{X}}}(x_1,\ldots,x_{M-1},Mx_{\mathrm{avg}}-\sum_{i=1}^{M-1}x_i), \end{aligned}$$

which is exactly (A.3). □

Secondly, we show the relationship between the PDF of confidence score and the PDFs of predictive probabilities. Note that the confidence score of an SR model is the maximum predictive probability, so the following lemma bounds the PDF of confidence by PDFs of predictive probabilities.

Lemma A.2. *Let* Π^k $(1 \le k \le K)$ *be* K *continuous random variables, and* $\mathrm{C} := \max_k \Pi^k$*. Then we have*

$$p_{\mathrm{C}}(\kappa) \le \sum_{k=1}^{K} p_{\Pi^k}(\kappa). \quad \text{(A.5)}$$

Proof. First of all, we prove $\forall\kappa_1, \kappa_2, \kappa_1 < \kappa_2$,

$$F_{\mathrm{C}}(\kappa_2) - F_{\mathrm{C}}(\kappa_1) \le \sum_{k=1}^{K} F_{\Pi^k}(\kappa_2) - F_{\Pi^k}(\kappa_1) \quad \text{(A.6)}$$

It is easy to see that

$$F_{\mathrm{C}}(\kappa) = F_{\Pi^1,\ldots,\Pi^K}(\kappa,\ldots,\kappa) = \int_{(-\infty,\kappa]^K}\mathrm{d}\pi^1\cdots\mathrm{d}\pi^K\ p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K),$$

[1] A PDF has a subscript to denote which random variable this PDF belongs to.

so the left-hand side of (A.6) is

$$\begin{aligned}
&\int_{(-\infty,\kappa_2]^K} \mathrm{d}\pi^1 \cdots \mathrm{d}\pi^K \; p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K) \\
-&\int_{(-\infty,\kappa_1]^K} \mathrm{d}\pi^1 \cdots \mathrm{d}\pi^K \; p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K) \\
=&\int_{(-\infty,\kappa_2]^K \setminus (-\infty,\kappa_1]^K} \mathrm{d}\pi^1 \cdots \mathrm{d}\pi^K \; p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K),
\end{aligned} \tag{A.7}$$

where the last equality is due to $(-\infty,\kappa_1] \subset (-\infty,\kappa_2]$, and the right-hand side of (A.6) is

$$\begin{aligned}
&\sum_{k=1}^{K} \int_{[\kappa_1,\kappa_2]} \mathrm{d}\pi^k \; p_{\Pi^k}(\pi^k) \\
=&\sum_{k=1}^{K} \int_{\mathbb{R}^{k-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k}} \mathrm{d}\pi^1 \cdots \mathrm{d}\pi^K \cdot p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K) \\
\geq&\int_{\bigcup_{k=1}^{K} \mathbb{R}^{k-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k}} \mathrm{d}\pi^1 \cdots \mathrm{d}\pi^K \cdot p_{\Pi^1,\ldots,\Pi^K}(\pi^1,\cdots,\pi^K),
\end{aligned} \tag{A.8}$$

where the last inequality is because $\mathbb{R}^{k-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k}$ for different k, $1 \leq k \leq K$, may have an intersection. To prove (A.6), we only need to prove that the right-hand side of (A.7) is less than or equal to the right-hand side of (A.8), which is equivalent to prove

$$(-\infty,\kappa_2]^K \setminus (-\infty,\kappa_1]^K \subset \bigcup_{k=1}^{K} \mathbb{R}^{k-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k}. \tag{A.9}$$

Now we prove (A.9). $\forall(\pi^1,\ldots,\pi^K) \in (-\infty,\kappa_2]^K \setminus (-\infty,\kappa_1]^K$, we have

$$\forall k, 1 \leq k \leq K, \pi^k \leq \kappa_2, \tag{A.10}$$

$$\exists k_0, 1 \leq k_0 \leq K, \pi^{k_0} > \kappa_1, \tag{A.11}$$

where (A.11) is because if all π^k is less than or equal to κ_1 instead, then $(\pi^1,\ldots,\pi^K) \in (-\infty,\kappa_1]^K$, which contradicts with $(\pi^1,\ldots,\pi^K) \in (-\infty,\kappa_2]^K \setminus (-\infty,\kappa_1]^K$. Thus, $\pi^{k_0} \in [\kappa_1,\kappa_2]$, so

$$(\pi^1,\ldots,\pi^K) \in \mathbb{R}^{k_0-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k_0} \subset \bigcup_{k=1}^{K} \mathbb{R}^{k-1}\times[\kappa_1,\kappa_2]\times\mathbb{R}^{K-k},$$

which is precisely (A.9), and therefore (A.6) is proved.

With (A.6) and the definition of derivatives, it is easy to see that $F'_{\mathrm{C}}(\kappa) \leq \sum_{k=1}^{K} F'_{\Pi^k}(\kappa)$, which is equivalent to $p_{\mathrm{C}}(\kappa) \leq \sum_{k=1}^{K} p_{\Pi^k}(\kappa)$. Thus, Lemma A.2 is proved. □

Based on the lemmas above, we have the following lemma, which is precisely (A.2). When (A.2) is proved, the proof of Lemma 1 completes.

Lemma A.3. *If Assumption 1 holds, then*

$$p(\kappa_E(x) = t | a = 1) \leq M^M B (1-t)^{M-1}. \tag{A.12}$$

Proof. Applying Lemma A.1 to the ensemble, we have

$$p(\pi_E^k = t | a = 1) = M \int_{\mathbb{R}^{M-1}} \mathrm{d}\pi_1^k \cdots \mathrm{d}\pi_{M-1}^k \cdot p(\pi_1^k,\ldots,\pi_{M-1}^k, \pi_M^k = Mt - \sum_{i=1}^{M-1} \pi_i^k | a = 1), \tag{A.13}$$

The integrand in the right-hand side of (A.13) being non-zero requires

$$\begin{cases} 0 \leq \pi_i^k \leq 1, i = 1,2,\ldots,M-1 \\ 0 \leq Mt - \sum_{i=1}^{M-1} \pi_i^k \leq 1 \end{cases}. \tag{A.14}$$

The inequalities above imply $Mt \leq 1 + \sum_{i=1}^{M-1} \pi_i^k \leq M - 1 + \pi_i^k, \forall i \in \{1, 2, ..., M-1\}$, which further derives $Mt - M + 1 \leq \pi_i^k \leq 1, \forall i \in \{1, 2, ..., M\}$. Thus, (A.13) can be rewritten as

$$p(\pi_E^k = t|a=1) = M \int_{[Mt-M+1,1]^{M-1}} \mathrm{d}\pi_1^k \cdots \mathrm{d}\pi_{M-1}^k \cdot p(\pi_1^k, \ldots, \pi_{M-1}^k, \pi_M^k = Mt - \sum_{i=1}^{M-1} \pi_i^k | a=1).$$

Considering that $p(\pi_1^k, \ldots, \pi_{M-1}^k, \pi_M^k|a=1)$ is bounded as Assumption 1 claims, let B_k be its least upper bound. Then we have

$$p(\pi_E^k = t|a=1) \leq M \int_{[Mt-M+1,1]^{M-1}} \mathrm{d}\pi_1^k \cdots \mathrm{d}\pi_{M-1}^k B_k = M^M B_k \cdot (1-t)^{M-1}. \quad \text{(A.15)}$$

This inequality combined with Lemma A.2 derives $p(\kappa(x) = t|a=1) \leq \sum_{k=1}^K p(\pi_E^k = t|a=1) \leq M^M(1-t)^{M-1} \cdot \sum_{k=1}^K B_k$, which is equivalent to the conclusion of this lemma since $B = \sum_{k=1}^K B_k$. □

A.2 Proof of Equation (2) of the Main Text (Preparing for Lemma 2)

According to the definition of selective risk, we have

$$\begin{aligned}
&R(f, \kappa; t) \\
=&\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{1}_{y \neq f(x)} | \kappa(x) \geq t] \\
=&P(y \neq f(x) | \kappa(x) \geq t) \\
=&P(y \neq f(x), a=0 | \kappa(x) \geq t) + P(y \neq f(x), a=1 | \kappa(x) \geq t) \\
=&P(y \neq f(x) | a=0, \kappa(x) \geq t) P(a=0|\kappa(x) \geq t) + P(y \neq f(x)|a=1, \kappa(x) \geq t) P(a=1|\kappa(x) \geq t).
\end{aligned}$$

According to the definition of conditional selective risk, the equation above can be rewritten as

$$\begin{aligned}
&R(f, \kappa; t|a=0) P(a=0|\kappa(x) \geq t) + R(f, \kappa; t|a=1) P(a=1|\kappa(x) \geq t) \\
=&R(f, \kappa; t|a=0)[1 - P(a=1|\kappa(x)] \geq t) + R(f, \kappa; t|a=1) P(a=1|\kappa(x) \geq t) \\
=&R(f, \kappa; t|a=0) + P(a=1|\kappa(x) \geq t) \cdot [R(f, \kappa; t|a=1) - R(f, \kappa; t|a=0)] \\
=&R(f, \kappa; t|a=0) + \lambda(f, \kappa; t) \cdot P(a=1|\kappa(x) \geq t), \quad \text{(A.16)}
\end{aligned}$$

where $\lambda(f, \kappa; t) := R(f, \kappa; t|a=1) - R(f, \kappa; t|a=0)$. Applying (A.16) to the ensemble and considering $\lambda(f_E, \kappa_E; t) \leq 1$, we derive an upper bound of the ensemble's selective risk

$$R_E(t) \leq R_E(t|a=0) + P(a=1|\kappa_E(x) \geq t). \quad \text{(A.17)}$$

Thus, Equation (2) of the main text is proved.

A.3 Proof of Theorem 1

Proof. **Step 1.** We prove

$$\lim_{t \to 1^-} R_E(t) < \lim_{t \to 1^-} R_*(t). \quad \text{(A.18)}$$

Firstly, we prove $\lim_{t\to 1^-} R_E(t) \leq \lim_{t \to 1^-} R_*(t|a=0)$. According to Lemma 2, we have

$$\begin{aligned}
\lim_{t \to 1^-} R_E(t) \leq& \lim_{t \to 1^-} \left[R_*(t|a=0) + \frac{\gamma \cdot (1-t)^M}{\gamma \cdot (1-t)^M + P(\kappa_*(x) \geq t, a=0)} \right] \\
=& \lim_{t \to 1^-} R_*(t|a=0) + \lim_{t \to 1^-} \frac{\gamma \cdot (1-t)^M}{\gamma \cdot (1-t)^M + P(\kappa_*(x) \geq t, a=0)} \\
=& \lim_{t \to 1^-} R_*(t|a=0) + \lim_{t \to 1^-} \frac{-M\gamma(1-t)^{M-1}}{-M\gamma(1-t)^{M-1} - p(\kappa_*(x) = t, a=0)} \\
=& \lim_{t \to 1^-} R_*(t|a=0) \quad \text{(A.19)}
\end{aligned}$$

where the second-to-last equality is due to L'Hospital's rule, and the last equality is due to $\lim_{t\to 1^-} M\gamma(1-t)^{M-1} = 0$ and $\lim_{t\to 1^-} p(\kappa_*(x) = t, a=0) = P(a=0) \lim_{t\to 1^-} p(\kappa_*(x) = t|a=0) > 0$ (Assumption 3).

Secondly, we prove $\lim_{t\to 1^-} R_*(t) > \lim_{t\to 1^-} R_*(t|a=0)$. According to (A.16), we have

$$\lim_{t\to 1^-} R_*(t) = \lim_{t\to 1^-} R_*(t|a=0) + \lim_{t\to 1^-} \lambda(f_*, \kappa_*; t) \cdot \lim_{t\to 1^-} P(a=1|\kappa_*(x) \geq t). \tag{A.20}$$

Using Bayes' rule and L'Hospital's rule, we have

$$\begin{aligned}\lim_{t\to 1^-} P(a=1|\kappa_*(x) \geq t) &= \lim_{t\to 1^-} \frac{P(\kappa_*(x) \geq t|a=1)P(a=1)}{P(\kappa_*(x) \geq t|a=1)P(a=1) + P(\kappa_*(x) \geq t|a=0)P(a=0)} \\ &= \lim_{t\to 1^-} \frac{p(\kappa_*(x) = t|a=1)P(a=1)}{p(\kappa_*(x) = t|a=1)P(a=1) + p(\kappa_*(x) = t|a=0)P(a=0)} \\ &> 0,\end{aligned} \tag{A.21}$$

where the last inequality is due to Assumption 3. Applying this inequality along with $\lim_{t\to 1^-} \lambda(f*, \kappa_*; t) = \lim_{t\to 1^-} [R_*(t|a=1) - R_*(t|a=1)] > 0$ (Assumption 2) to (A.20), we have

$$\lim_{t\to 1^-} R_*(t) > \lim_{t\to 1^-} R_*(t|a=0). \tag{A.22}$$

Combining (A.19) and (A.22), we derive (A.18).

Step 2. We prove the ensemble's confidence threshold and the member model's confidence threshold approach 1 when their coverage approach 0. Due to $\phi_*(t) = P(\kappa_*(x) \geq t) = P(\kappa_*(x) \geq t|a=0)P(a=0) + P(\kappa_*(x) \geq t|a=1)P(a=1)$, we derive

$$\begin{aligned}\frac{d\phi_*(t)}{dt} &= -p(\kappa_*(x) = t|a=0)P(a=0) - p(\kappa_*(x) = t|a=1)P(a=1) \\ &\leq -p(\kappa_*(x) = t|a=0)P(a=0).\end{aligned} \tag{A.23}$$

Because $\lim_{t\to 1^-} p(\kappa_*(x) = t|a=0) > 0$ (Assumption 3), there exists $\delta_1 > 0$, such that $p(\kappa_*(x) = t|a=0) > 0, \forall t \in (1-\delta_1, 1)$. Combining this with (A.23), we have $\frac{d\phi_*(t)}{dt} < 0, \forall t \in (1-\delta_1, 1)$ and thus $\phi_*(t)$ is reversible on $(1-\delta_1, 1)$. Considering $p(\kappa_E(x) = t|a=0) = p(\kappa_*(x) = t|a=0)$, we derive $\phi_E(t)$ is reversible on $(1-\delta_1, 1)$ with the same reasoning as above.

Let ϕ_*^{-1} and ϕ_E^{-1} be the reverse functions of ϕ_* and ϕ_E on $(1-\delta_1, 1)$, respectively. It is easy to see that ϕ_*^{-1} is a continuous function, and $\phi_*^{-1}(0) = 1$. Therefore,

$$\lim_{\phi\to 0^+} \phi_*^{-1}(\phi) = 1. \tag{A.24}$$

For the ensemble, we similarly have

$$\lim_{\phi\to 0^+} \phi_E^{-1}(\phi) = 1. \tag{A.25}$$

Step 3. Combining (A.18), (A.24) and (A.25), we have $R_E(\phi_E^{-1}(\phi)) < R_*(\phi_*^{-1}(\phi))$ when $\phi \to 0^+$, which is equivalent to the result of Theorem 1. □

B Details of Experiments

B.1 Datasets

The experiments were conducted on multiple data sets of image classification and text classification. The image classification datasets are CIFAR-10, CIFAR-100, [1] and SVHN [2], whose image sizes are all $32 \times 32 \times 3$ pixels. The datasets of text classification are MRPC [3], MNLI [4] and QNLI [5]. The task of MRPC is to judge whether two paragraphs of text are semantically equivalent. MNLI's task is to judge the inferential relationship between sentences (three categories). The task of QNLI is to determine whether a paragraph has the answer to a given question. The sizes of the training set, development set, and test set of each data set used in experiments are shown in Table B.1. MNLI's development set and test set are divided into *matched* and *mismatched* parts. In the table, (m) represents matched, and (mm) represents mismatched. The matched parts are sampled from the same source as the training set, while the mismatched parts are sampled from different sources. Current selective classification only considers test samples from the same distribution as the training set, so only the matched parts are used in experiments. In addition, test sets of MRPC, QNLI, and MNLI are not accessible, so we use their development sets as test sets. According to [6, 7], since CIFAR-10, CIFAR-100 and SVHN originally had no development set, their development sets were 2000 samples randomly divided from corresponding test sets.

Table B.1: Sizes of training sets, development sets, and test sets for each dataset used in experiments

Datasets	Training Set	Development Set	Test Set	Number of Classes
CIFAR-10	50.0k		10.0k	10
CIFAR-100	50.0k		10.0k	100
SVHN	73.3k		26.0k	10
MRPC	3.7k	0.4k	1.7k	2
QNLI	104.7k	5.5k	5.5k	2
MNLI	392.7k	9.8k (m)/ 9.8k(mm)	9.8k(m)/9.8k(mm)	3

Table B.2: The value of o on each dataset

Dataset	CIFAR-10	SVHN	CIFAR-100	MRPC	QNLI	MNLI-(m)
o	2.20	2.60	4.60	1.80	1.60	2.80

B.2 Model Implementations and Training Procedures

For image classification, the backbone model is VGG-16 [8] with Dropout [9], batch normalization [10]. It is trained in the same way as [7]. The model is optimized using SGD with an initial learning rate of 0.1 (the learning rate decays by half in every 25 epochs), the momentum of 0.9, weight decay of 0.0005, batch size of 128, and a total training epoch of 300. Data preprocessing includes data augmentation (random cropping and flip) and normalization. The implementations of the backbone model and data preprocessing are based on the official open-sourced implementation of SAT to ensure a fair comparison.

For text classification, the backbone model of selective classifiers is BERT-base [11]. Pretrained BERT-base is provided by the Huggingface Transformer Library [12]. It is trained/fine-tuned in the same way as [13], except on dataset MRPC. On QNLI and MNLI, the model is trained/fine-tuned using AdamW [14] for 3 epochs, with a learning rate of 2×10^{-5}, batch size of 32, and the maximum input sequence length of 128. On MRPC, the model is trained/fine-tuned for 10 epoch, with other settings the same as those on QNLI and MNLI. This unique setting of training epoch is due to the small number of samples in MRPC, which makes the training require more epochs to reach convergence on MRPC.

B.3 Hyperparameters of Selective Classifiers

For the hyperparameter o of Gambler, we tune o on validation sets in the same way as [6]. The value of o on each dataset is listed in Table B.2. For the hyperparameter α of SAT, we set $\alpha = 0.99$, the same as [7]. For the hyperparameter λ of Reg-curr, we set $\lambda = 0.05$.

C Selective Risks of Ensembles

Table C.1 and C.2 shows the selective risks of ensembles under coverage 10%-100% on each dataset. Notably, no ensemble consistently outperforms others under all coverage on all datasets. This phenomenon is because different ensembles have similar overall performance but adopt different trade-offs between coverage and selective risk.

D Further Properties of Selective Classifier Ensemble

D.1 The Effect of Number of Members on Selective Classifier Ensemble

We evaluate AURCs of the SR ensemble (i.e., Deep Ensemble), Gambler ensemble, and SAT ensemble of different numbers of members on CIFAR10, and find that an ensemble with more members has a better performance, but is less efficient. The results are shown in Figure D.1. In most cases, the AURC on the test set of CIFAR-10 decreases as the number of members in the ensemble increases.

Table C.1: The selective risks of ensembles under coverage 10%-100% on image classification datasets. The means and standard deviations are calculated over three trials. The best entries are marked in bold.

Dataset	coverage (%)	Deep Ensemble	Gambler ensemble	SAT ensemble	Reg-curr ensemble
CIFAR-10	100	5.31±0.03	**5.29±0.03**	5.47±0.04	5.74±0.07
	90	**1.68±0.02**	1.99±0.01	2.15±0.06	1.89±0.02
	80	**0.45±0.05**	0.51±0.02	0.63±0.03	0.61±0.09
	70	**0.17±0.01**	0.21±0.01	0.26±0.01	0.18±0.02
	60	0.11±0.01	0.18±0.03	0.17±0.01	**0.08±0.00**
	50	0.11±0.01	0.14±0.02	0.11±0.01	**0.07±0.01**
	40	0.12±0.03	0.15±0.02	**0.06±0.01**	0.07±0.01
	30	0.13±0.05	0.13±0.03	**0.06±0.02**	0.08±0.03
	20	0.12±0.02	0.17±0.05	**0.00±0.00**	0.02±0.02
	10	0.10±0.08	0.14±0.05	**0.00±0.00**	**0.00±0.00**
SVHN	100	2.44±0.01	2.42±0.02	**2.36±0.01**	2.41±0.01
	90	0.59±0.00	0.60±0.03	**0.50±0.01**	0.54±0.02
	80	0.42±0.03	0.38±0.01	**0.34±0.01**	0.39±0.02
	70	0.34±0.02	0.32±0.01	**0.31±0.01**	0.35±0.01
	60	0.32±0.02	0.30±0.01	**0.28±0.01**	0.35±0.01
	50	0.29±0.02	**0.26±0.01**	**0.26±0.00**	0.30±0.01
	40	**0.25±0.02**	0.27±0.02	**0.25±0.01**	0.28±0.01
	30	0.22±0.03	0.26±0.01	**0.20±0.01**	0.25±0.03
	20	0.22±0.01	0.26±0.01	**0.18±0.02**	0.19±0.03
	10	0.21±0.02	0.23±0.00	**0.17±0.02**	0.18±0.03
CIFAR-100	100	**24.66±0.08**	25.50±0.05	25.23±0.13	25.70±0.09
	90	**19.15±0.15**	19.88±0.05	19.77±0.28	20.16±0.14
	80	**14.32±0.22**	15.75±0.09	15.00±0.20	15.22±0.07
	70	**9.78±0.13**	12.11±0.18	10.29±0.24	10.41±0.38
	60	**5.81±0.06**	8.89±0.16	6.43±0.20	6.58±0.27
	50	**2.95±0.04**	6.22±0.10	3.41±0.15	3.45±0.05
	40	**1.40±0.13**	4.37±0.06	1.96±0.13	1.74±0.11
	30	**0.75±0.05**	2.67±0.01	1.13±0.02	0.89±0.06
	20	**0.62±0.06**	1.91±0.04	0.72±0.06	**0.62±0.04**
	10	0.33±0.09	1.42±0.16	0.57±0.09	**0.13±0.05**

In addition, as the number of members in the ensemble grows, the effect of adding one member drops. On the one hand, the result shows that an ensemble with a small number of members has good selective classification performance. On the other hand, it indicates that when the number of member models is large, increasing the number of members to improve the performance of the selective classification ensemble is inefficient.

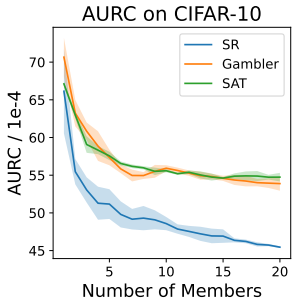

Figure D.1: The AURCs on the test set of CIFAR-10 of the SR ensemble (Deep Ensemble), Gambler ensemble, and SAT ensemble of different numbers of members

Table C.2: The selective risks of ensembles under coverage 10%-100% on text classification datasets. The means and standard deviations are calculated over three trials. The best entries are marked in bold.

Dataset	coverage (%)	Deep Ensemble	Gambler ensemble	SAT ensemble	Reg-curr ensemble
MRPC	100	14.13±0.23	14.62±0.23	**13.64±0.23**	15.28±0.31
	90	11.41±1.02	11.50±0.46	**10.69±0.13**	11.68±0.44
	80	**7.75±0.29**	9.99±0.88	8.46±0.63	8.36±0.29
	70	**6.41±0.33**	8.51±0.72	7.93±0.44	6.64±0.29
	60	**5.71±0.33**	7.35±1.20	6.39±0.19	6.12±0.33
	50	4.08±1.29	4.41±0.40	**3.59±1.01**	4.74±0.23
	40	**3.25±0.29**	3.86±0.76	**3.25±0.76**	3.66±0.50
	30	3.25±0.00	3.52±0.38	3.52±0.38	**2.98±0.77**
	20	**2.44±1.72**	3.25±0.58	3.25±0.58	3.66±0.00
	10	3.25±2.30	4.88±0.00	**1.63±1.15**	5.69±1.15
QNLI	100	8.16±0.04	8.18±0.20	**8.03±0.09**	8.17±0.01
	90	**4.74±0.04**	5.04±0.14	5.03±0.09	**4.74±0.12**
	80	**2.94±0.11**	3.08±0.08	2.97±0.04	2.98±0.01
	70	**1.84±0.10**	1.91±0.04	1.92±0.11	1.88±0.06
	60	**1.20±0.05**	1.27±0.05	1.36±0.01	1.30±0.08
	50	1.04±0.05	1.04±0.08	1.13±0.03	**0.98±0.03**
	40	0.72±0.02	0.73±0.04	1.02±0.11	**0.70±0.06**
	30	0.45±0.03	0.51±0.03	0.83±0.08	**0.43±0.09**
	20	**0.30±0.11**	0.37±0.15	0.67±0.04	**0.30±0.09**
	10	0.30±0.09	**0.12±0.09**	0.61±0.34	**0.12±0.09**
MNLI	100	15.04±0.06	**14.82±0.15**	15.03±0.06	14.89±0.14
	90	**11.01±0.11**	11.78±0.09	11.37±0.06	11.21±0.17
	80	**7.93±0.08**	9.62±0.08	8.26±0.20	8.02±0.07
	70	**5.81±0.04**	8.03±0.23	**5.81±0.12**	5.85±0.22
	60	4.28±0.07	6.41±0.25	4.08±0.19	**4.05±0.10**
	50	3.22±0.04	4.95±0.10	**2.99±0.18**	3.04±0.10
	40	2.69±0.11	3.57±0.21	**2.08±0.03**	2.22±0.06
	30	2.13±0.14	2.35±0.18	**1.57±0.03**	1.75±0.06
	20	**1.34±0.10**	1.53±0.17	1.39±0.13	1.36±0.06
	10	0.98±0.05	1.32±0.14	**0.71±0.22**	**0.71±0.08**

D.2 Good Classification Performance Does Not Imply Good Selective Classification Performance

It is well known that the ensemble has better classification performance than an individual model, but this does not guarantee a better selective classification performance of the ensemble. To demonstrate this, we design an SR model with a big backbone, and show that it has as good classification performance as an Deep Ensemble with a standard backbone but worse selective classification performance than an SR model with a standard backbone. The big backbone is designed to have twice as many filters in every convolutional layer and neurons in every fully connected hidden layer as those of the standard VGG-16, which is therefore called *Big VGG-16*. It is easy to see that its number of parameters is approximately $2^2 = 4$ times as many as that of standard VGG-16. We train an Deep Ensemble of 4 VGG-16s and an SR model with a backbone of Big VGG-16 on CIFAR-10 and show the evaluation results in Figure D.2 and Table D.1. Figure D.2 shows that when coverage is high, the ensemble and the big individual model have similar selective risks, and especially, the classification error rates (i.e., selective risk of 100% coverage) of the ensemble and the big individual model are similar. However, when coverage is low, the big individual model has significantly higher selective risk than the ensemble. Table D.1 shows that the AURC of Big VGG-16 is much higher than the ensemble of 4 VGG-16s and even higher than SR. In summary, we show that a selective classifier with a good classification performance is not guaranteed to have good selective classification performance, so the good selective classification performance of the ensemble is not a trivial result of its good classification performance.

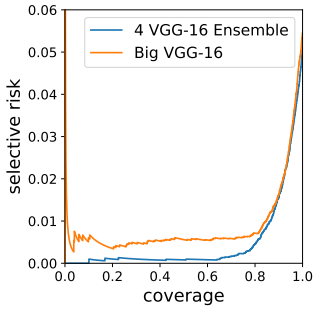

Figure D.2: Risk-coverage curves of the ensemble of 4 VGG-16s and the Big VGG-16 on CIFAR-10

Table D.1: The AURCs($/10^{-4}$) of Big VGG-16, a vanilla VGG-16, and the ensemble of 4 VGG-16s on CIFAR-10. The best entries are marked in bold.

Dataset	Big VGG-16	VGG-16	Ensemble
CIFAR-10	89.2	69.6	**49.3**

D.3 The Effects of Label Noise of SVHN on Selective Classifier Ensembles

In this section, we compare the effect of label noise of SVHN on the Deep Ensemble with that on SAT ensemble, whose result might explain the abnormal experimental results (compared to results on other datasets) on SVHN in Section 6 of the main text. SVHN is not a clean dataset, and much more label noise can be detected in SVHN than in CIFAR-10 and CIFAR-100. Using the soft label of SAT [7], we detect label noise in SVHN, CIFAR-10, and CIFAR-100, and find that SVHN has significantly more label noise than CIFAR-10 and CIFAR-100. The result is presented in the following. In addition, it is known that SAT is robust to label noise [7], while SR is not so, so we conjecture that the label noise of SVHN is why the Deep Ensemble is inferior to SAT on SVHN.

We detect label noise with the help of the soft label of SAT. For a sample x_i, the soft label of SAT [7], t_{i,y_i}, is used to measure x_i's learning difficulty. The soft label of SAT is initialized as 1 and updated at every training epoch as below

$$t_{i,y_i} \leftarrow \alpha \times t_{i,y_i} + (1-\alpha) \times p_\theta(y_i|x_i),$$

where $p_\theta(Y|x)$ is the predictive probability distribution of the classifier, y_i is the label of x_i, α is a hyperparameter. The smaller the t_{i,y_i} is, the lower the true class predictive probability of the classifier on x_i during training time, indicating that x_i is more difficult to learn. By selecting a percentage of samples with the lowest t_{i,y_i}, we get the most difficult samples to learn for the classifier, from which we can easily detect label noise manually.

In training sets of SVHN, CIFAR-10, and CIFAR-100, we detect label noise manually among the top-0.1% difficult (measured by the soft label of SAT) samples. The numbers of mislabeled samples detected in SVHN, CIFAR-10, and CIFAR100 are shown in Table D.2. The result shows that SVHN has significantly more mislabeled samples detected than CIFAR-10 and CIFAR-100, indicating much more label noise in SVHN than in CIFAR-10 and CIFAR-100.

Table D.2: Numbers of mislabeled samples in the top-0.1% difficult training samples of SVHN, CIFAR-10, and CIFAR-100.

Dataset	#Mislabeled	#Top-0.1%	Proportion
SVHN	73	73	100%
CIFAR-10	1	50	2%
CIFAR-100	1	50	2%

Table D.3: AURC/10^{-4} of Deep Ensemble and SAT ensemble on the clean SVHN

Dataset	#Member	Deep Ensemble	SAT Ensemble
clean SVHN	1	12.3	**7.8**
	2	8.2	**7.1**
	3	7.3	**6.8**
	4	**6.8**	**6.8**
	5	**6.4**	6.8

To verify the effect of label noise, the following experiments are designed. Firstly, we detect label noise manually among the 1% of the hardest-to-learn samples of SVHN training set and test set, using the soft label of SAT. Secondly, we remove the detected mislabeled samples from the original dataset. The remaining SVHN dataset is called the *clean SVHN*. Accordingly, the original dataset is called the *original SVHN*. Finally, we retrain and test the Deep Ensemble and SAT ensemble and compare their test results. In the second step, the reason for removing mislabeled samples rather than modifying them is that some samples cannot be classified even by humans, and some samples are not in the range of categories of SVHN. Thus, the label noise cannot be eliminated by modifying the labels but by removing mislabeled samples.

The test results of the Deep Ensemble and SAT ensemble on clean SVHN are shown in Table D.3. It is not surprising that the AURCs of the Deep Ensemble and SAT ensemble are significantly lower on the clean SVHN than the original SVHN. Furthermore, on the clean SVHN, when the number of members is 5, the AURC of the Deep Ensemble is lower than that of SAT ensemble. Combined with results on the original SVHN, where the AURC of the Deep Ensemble is higher than that of SAT ensemble, we conclude that label noise in SVHN is why the Deep Ensemble has a higher AURC than SAT ensemble. In other words, label noise is why the Deep Ensemble performs worse in selective classification than SAT ensemble on SVHN.

In summary, by experiments, we show that the Deep Ensemble is not as robust to label noise as SAT ensemble, and label noise in SVHN is why the Deep Ensemble is not as good as SAT ensemble on SVHN. We construct the *clean SVHN*, which is SVHN without some mislabeled samples. On the clean SVHN, we compare the Deep Ensemble with SAT ensemble and find that the Deep Ensemble is superior to SAT ensemble in selective classification performance. Combined with former experimental results, we conclude that label noise in SVHN is why the Deep Ensemble is inferior to SAT on SVHN.

Considering the experimental results on the clean SVHN and previous experimental results on CIFAR-10 and CIFAR-100 (see Table D.3 and Table 1 of the main text), the Deep Ensemble is superior to SAT ensemble in selective classification on clean image classification datasets, Thus, Deep Ensemble is the state-of-the-art selective classification method on clean image classification datasets, but is not as robust to label noise as SAT ensemble.

E The Lower Bound of Maximum ϕ_0 in Theorem 1

This section discusses the lower bound of maximum ϕ_0 mentioned in Theorem 1. We aim to calculate the maximum ϕ_0's lower bound without training an ensemble (otherwise, we can measure it directly on the ensemble).

Optimization Problem. To calculate the lower bound of maximum ϕ_0, we need to solve the following optimization problem

$$\min_{t,t_E} \ \phi_*(t) \text{ s.t. } \phi_E(t_E) \geq \phi_*(t) \qquad R_E(t_E) < R_*(t). \tag{E.1}$$

Algorithm 1 A Lower Bound of Maximum ϕ_0.

Input: $\kappa_*(\cdot)$, B, the test set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; the oracle $\Omega : \mathbb{X} \to \{0, 1\}$ that tells whether a sample is definite; the number of member models M.
Output: An lower bound of maximum ϕ_0 mentioned in Theorem 1
$left = 0$
$right = 1$
$\epsilon = 10^{-9}$
while $right - left > \epsilon$ **do**
  $t_* = (left + right)/2$
  $t_E = \text{SEARCHFORTAUENS}(t_*, \kappa_*, \mathcal{D}, \Omega)$
  if t_E is not None and VERIFYSECONDCONSTRAINT(κ_*, t_*, t_E, $\mathcal{D}$, Ω, M, B) is True **then**
    $right = t_*$
  else
    $left = t_*$
  end if
end while
$opt = (left + right)/2$
return $\frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\kappa_*(x_i) \geq opt}$

Using Lemma 2 and $\phi_E(t) \geq P(\kappa_E(x) \geq t, a = 0) = P(\kappa_*(x) \geq t, a = 0)$, we strengthen the constraints of (E.1) to obtain a looser lower bound of maximum ϕ_0:

$$\max_{t_*, t_E} \phi_*(t_*) \text{ s.t. } P(\kappa_*(x) \geq t_E, a = 0) \geq \phi_*(t_*)$$
$$R_*(t_E|a = 0) + \frac{\gamma \cdot (1 - t_E)^M}{\gamma \cdot (1 - t_E)^M + P(\kappa_*(x) \geq t_E, a = 0)} < R_*(t_*).$$

For the convenience of solving this optimization problem, we further strengthen the first constraint to obtain a looser lower bound:

$$\max_{t_*, t_E} \phi_*(t_*) \text{ s.t. } P(\kappa_*(x) \geq t_E, a = 0) = \phi_*(t_*) \tag{E.2}$$
$$R_*(t_E|a = 0) + \frac{\gamma \cdot (1 - t_E)^M}{\gamma \cdot (1 - t_E)^M + P(\kappa_*(x) \geq t_E, a = 0)} < R_*(t_*).$$

Algorithm. We design Algorithm 1 to search for the solution to (E.2), where *oracle* tells whether a sample is definite, and this oracle can be implemented by an ensemble with M' ($M' \ll M$) members. Since t_E is determined by t_* according to the first constraint of (E.2), (E.2) can be reduced to a one-dimensional search problem. Our algorithm adopts a binary search for efficiency, although this method might provide a suboptimal solution. Because $\phi_*(t_*)$ is a non-increasing function of t_*, Algorithm first search for the minimum t_* by binary search. The procedure of Algorithm 1 in each iteration of the binary search is as follows.

1. Given current t_*, Algorithm 1 determines t_E using SEARCHFORTAUENS (see Algorithm 2), a procedure that searches for $t_E \in [0, 1]$ using binary search s.t. $P(\kappa_*(x) \geq t_E, a = 0) = \phi_*(t_*)$. Note that t_E might not exist if t_* is so low that $\phi_*(t_*) > P(a = 0) = \sup_{t_E \in [0,1]} P(\kappa_*(x) \geq t, a = 0)$. This problem will be addressed shortly.
2. Algorithm 1 exams whether t_E exists. If t_E exists, Algorithm 1 then examines whether the second constraint of (E.2) holds for current t_* and t_E, which is implemented by VERIFYSECONDCONSTRAINT (see Algorithm 3).
3. If t_E exists and the second constraint holds, Algorithm 1 searches for a smaller t_* in the left half feasible area; otherwise, Algorithm 1 searches for a greater t_* in the right half feasible area.

Once the binary search completes, Algorithm 1 returns the coverage of minimum t_*.

Experiment. To show that Algorithm 1 works in reality, we run this algorithm in the same setting as Section 6 of the main text. In this experiment, $M = 5$, the oracle is implemented by another ensemble

Algorithm 2 SEARCHFORTAUENS

Input: $\kappa_*(\cdot)$, the confidence threshold t_*; the test set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; the oracle $\Omega : \mathbb{X} \to \{0, 1\}$ that tells whether a sample is definite.
Output: $t_E \in [0, 1]$ that satisfies the first constraint of (E.2) given t_*.
$\phi = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\kappa_*(x_i) \geq t_*}$
if $\phi > \frac{1}{N} \sum_{i=1}^N \Omega(x_i)$ **then**
  return None
end if
$left = 0$
$right = 1$
$\epsilon = 10^{-9}$
while $right - left > \epsilon$ **do**
  $t_E = (left + right)/2$
  if $\frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\kappa_*(x_i) \geq t_E} \cdot \Omega(x_i) < \phi$ **then**
    $right = t_E$
  else
    $left = t_E$
  end if
end while
return $(left + right)/2$

Algorithm 3 VERIFYSECONDCONSTRAINT

Input: $\kappa_*(\cdot)$; the confidence threshold t_*; t_E; the test set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; the oracle $\Omega : \mathbb{X} \to \{0, 1\}$ that tells whether a sample is definite; the number of member models M; and B.
Output: True if and only if t_* and t_E satisfy the second constraint of (E.2).
$\gamma = BM^{M-1} \frac{1}{N} \sum_{i=1}^N [1 - \Omega(x_i)]$
$leftHandSide = \frac{\sum_{i=1}^N \mathbf{1}_{f(x_i;\theta) \neq y_i} \cdot \Omega(x_i) \cdot \mathbf{1}_{\kappa_*(x_i) \geq t_E}}{\sum_{i=1}^N \Omega(x_i) \cdot \mathbf{1}_{\kappa_*(x_i) \geq t_E}} + \frac{\gamma(1-t_E)^M}{\gamma(1-t_E)^M + \frac{1}{N} \sum_{i=1}^N \Omega(x_i) \cdot \mathbf{1}_{\kappa_*(x_i) \geq t_E}}$
$rightHandSide = \frac{\sum_{i=1}^N \mathbf{1}_{f(x_i;\theta) \neq y_i} \cdot \mathbf{1}_{\kappa_*(x_i) \geq t_*}}{\sum_{i=1}^N \mathbf{1}_{\kappa_*(x_i) \geq t_*}}$
return $\mathbf{1}_{leftHandSide < rightHandSide}$

with two member models and outputs `True` if and only if the $S < 10^{-3}$. Note that it is difficult to estimate B, because: 1. we need to train an ensemble with M models to estimate B, which is costly; 2. the domain of $p(\pi_1^k, ..., \pi_M^k | a = 1)$ is of high dimension, so the observed data points are sparse in this domain, which makes the estimation of B more difficult. Thus, we do not estimate B but try several hypothetical values of B to see at what B the lower bound of maximum ϕ_0 is big.

With different Bs, we obtain different lower bounds of maximum ϕ_0 as Table E.1 shows. We can see that when the order of magnitude of B is not too big, the lower bound of maximum ϕ_0 is large and stable, which makes it possible for our algorithm to be used in practical applications. This result also indicates the relationship between the ensemble's diversity and its selective classification performance. Since an ensemble with a smaller B seems to have more diversity over ambiguous samples, the result in Table E.1 suggests that as long as the ensemble has enough diversity over ambiguous samples, the ensemble is guaranteed to have a lower selective risk than the member model under a considerable range of coverage.

References

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

[2] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[3] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing*, 2005.

Table E.1: The lower bound of maximum ϕ_0 (%).

B	1	10	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}	10^{11}	10^{12}	10^{13}	10^{14}
CIFAR-10	73.72	73.72	73.72	73.70	73.62	73.47	72.90	70.50	59.90	0	0	0	0	0	0
SVHN	77.51	77.51	77.51	77.51	77.44	77.23	76.54	68.86	0	0	0	0	0	0	0
CIFAR-100	32.03	32.03	32.00	32.00	31.93	31.79	31.50	30.45	26.91	17.96	10.36	0	0	0	0
MNLI	58.37	57.62	56.48	53.94	49.62	38.73	0	0	0	0	0	0	0	0	0
QNLI	65.31	65.31	65.31	64.34	62.31	59.20	53.07	37.16	0	0	0	0	0	0	0
MRPC	81.37	81.37	81.37	81.37	81.37	81.13	80.64	79.41	78.43	41.67	41.67	0	0	0	0

[4] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, 2018.

[5] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[6] Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. *Advances in Neural Information Processing Systems*, 2019.

[7] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: Beyond empirical risk minimization. *Advances in Neural Information Processing Systems*, 2020.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[13] Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, 2021.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.